ООО "СОТКОМ-ЮГ"

СТАНДАРТ ОРГАНИЗАЦИИ

ПРОЕКТ

Обозначение СТО не присвоено

Открытый протокол систем оповещения. Техническая спецификация

Редакция 0.21 / 22 октября 2026

Для публичного обсуждения

Настоящий проект стандарта не подлежит применению до его утверждения

2026

Предисловие

Разработан ООО "СОТКОМ-ЮГ". Представлен для публичного обсуждения как проект стандарта организации.

Обозначение СТО не присвоено. Стандарт не утвержден; дата введения в действие не установлена.

Правила оформления проекта установлены по выбранным положениям ГОСТ Р 1.5-2012. Основания оформления и сведения о правах использования приведены в приложении Е.

Требования протокола изложены в разделах 1-27 и обязательных частях приложений. Приложение Ж содержит полный профиль групповой передачи речи и обязательно при объявлении его поддержки. Пояснения к ключевым правилам приведены в справочном приложении Д.

Содержание

Предисловие....II
1 Область применения и статус....1
2 Нормативные ссылки....4
3 Термины, определения и сокращения....5
4 Архитектура: АРМ, УУ и ОУ....7
5 Владение и право прерывания....11
6 Идентификаторы и контекст....26
7 Транспорт и ограниченный XML....31
8 Сеанс, физический результат и потеря связи....41
9 Повторы, отмена и неизвестный исход....54
10 Профили ОУ: ВАУ и Сирена....65
11 Датчики и каталог ресурсов....72
12 Опросы, проверки и аварии....75
13 Диагностика, события и снимки....82
14 Защищенный обмен и ключи....95
15 Речь, записи и время....120
16 Конфигурация, контент и обновление....158
17 Сторонние сервисы и расширения....178
18 Проверки и критерии готовности....179
19 Контракт управляющего ядра....183
20 Семейство профилей и совместимость....212
21 Логические цели, реестр и маршрутизация....216
22 Готовность плана и проверка записей....231
23 Каскадная диагностика и групповой отчет....234
24 Регистрация и согласованная конфигурация....236
25 Паспорт масштаба и бюджета времени....248
26 Этапы программы и повторы....250
27 Внешние входы и автономное управление....255

Приложение А (обязательное для ядра) Каталог команд.....259
Приложение Б (обязательное для ядра) Машинные ошибки.....285
Приложение В (справочное) Примеры XML.....287
Приложение Г (справочное) Матрица требований.....367
Приложение Д (справочное) Пояснения к требованиям.....393
Приложение Е (справочное) Сведения об издании и правах использования.....395
Приложение Ж (обязательное при поддержке групповой передачи) Групповая передача речи.....401
Библиография и комплект редакции.....410
Журнал редакции.....413

Открытый протокол систем оповещения. Техническая спецификация

Проект, редакция 0.21

1 Область применения и статус

1.1 Настоящий документ представляет собой техническую спецификацию Открытого протокола систем оповещения. Он предназначен для разработчиков оборудования и программного обеспечения, проектировщиков и испытательных организаций. Документ устанавливает форматы сообщений, правила управления, передачи речи и контроля исполнения, а также критерии совместимости независимых реализаций.

1.2 Открытый протокол систем оповещения задает правила взаимодействия АРМ, устройств управления и оконечного оборудования любых производителей. Протокол устанавливает общие команды, модель состояний, профили возможностей и критерии совместимости. Спецификация не устанавливает торговые наименования, изготовителя или конкретную аппаратную платформу.

Настоящая спецификация устанавливает профиль open-alert-draft/0.21, правила управления конечными исполнениями, телеметрии, речи и сервисных операций. Нормативными являются требования, форматы, диапазоны, состояния, переходы и критерии результата.

Таблица 1 - Область применения и статус

Тип требования	**Применение**
Обязательное	Все реализации выбранного профиля соблюдают его команды, форматы, состояния и запреты.
Условно обязательное	При объявлении дополнительной функции обязательны весь ее контракт и зависимости.
Параметр изделия	Значение указывается в паспорте в установленном диапазоне; отсутствие обязательного значения запрещает использование функции.

1.3 ГОСТ Р 42.3.05-2023 устанавливает информационный обмен технических средств оповещения населения. Его приложения описывают разные интерфейсы; восьмибайтовый основной кадр приложения Б не характеризует весь стандарт. Настоящая спецификация задает самостоятельный формат и не является реализацией приложений А, Б или В этого ГОСТ.

1.4 При заявлении соответствия ГОСТ Р 42.3.05-2023 выполняются требования его пункта 5.2: обязательная поддержка ИП с предусмотренным исключением для производителей оконечных средств и ограничение назначения дополнительных протоколов. Замена ИП командами настоящего протокола на требуемом интерфейсе АРМ–УУ не удовлетворяет этому пункту. Одновременное наличие другого порта с ИП, шлюза или названия «дополняющий протокол» само по себе не устраняет несоответствие назначения дополнительного обмена.

1.5 Пункты 5.1 и 5.3 отделяют информационный протокол от механизмов транспорта, маршрутизации, резервирования и защиты. Эти функции рассматриваются отдельно от альтернативного кодирования

session.start, session.stop и квитанций. Передача файлов до оповещения относится к сервисному обмену лишь тогда, когда она не подменяет формирование и передачу сигналов, команд и квитанций обязательного ИП.

1.6 АРМ–АРМ, АРМ–УУ и сопряжение с вещанием по ГОСТ

Требуемый интерфейс выполняется по соответствующему приложению ГОСТ. Преобразователь проверяет каждое отображаемое действие и не объявляет форматы побайтово совместимыми.

1.7 УУ–ОУ и встроенный контроллер ОУ

Применимость исключения для изготовителя оконечных средств определяется реальной функцией изделия. Переименование устройства управления в оконечное средство не меняет требований к его внешнему интерфейсу.

1.8 Объектовая система и иные области

Область определяется назначением системы и применимыми требованиями, включая пожарные. Отсутствие данного интерфейса в ГОСТ Р 42.3.05-2023 само по себе не освобождает от других обязательных требований.

1.9 Защита, маршрутизация и обслуживание

Оцениваются отдельно в пределах пунктов 5.1–5.3. Их поддержка не доказывает соответствие предметных команд приложениям ГОСТ.

1.10 Проект сопряжения содержит матрицу «требование ГОСТ — интерфейс — команда — подтверждение — испытание». Для приема команды, фактического задействования и выключения фиксируются раздельные доказательства. Автоматическая квитанция не подменяется успешной доставкой TCP, а отсутствующее в нижнем протоколе подтверждение не создается преобразователем. Уровень доказательства сохраняется при преобразовании.

1.11 Статья 26 Федерального закона № 162-ФЗ устанавливает общее правило добровольного применения документов стандартизации

и случаи обязательности, включая публичное заявление о соответствии. Применимость требований к конкретной системе определяется также законодательством, заданием и обязательствами поставки. Несоответствие выбранному ГОСТ не равнозначно общему законодательному запрету самостоятельного протокола; настоящий документ не разрешает отступление от обязательного для проекта интерфейса.

2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие документы. Требования применяются в объеме, прямо установленном соответствующим профилем.

ГОСТ Р 42.3.05-2023. Протоколы информационного обмена. Общие требования.

https://meganorm.ru/mega_doc/norm/gost-r_gosudarstvennyj-standart/8/gost_r_42_3_05-2023_natsionalnyy_standart_rossiyskoy.html

Федеральный закон № 162-ФЗ, статья 26. Правила применения документов стандартизации.

https://base.garant.ru/71108018/3ac805f6d87af32d44de92b042d51285/

RFC 8785: каноническое представление JSON.

https://www.rfc-editor.org/rfc/rfc8785.html

Р 1323565.1.030-2020 и RFC 9367: российские алгоритмы в TLS 1.3.

https://tc26.ru/news/novosti-tk26/opublikovan-rfc-9367-opredelyayushchiy-poryadok-raboty-tls-1-3-s-gost.html

RFC 9367: российские криптонаборы TLS 1.3.

https://www.rfc-editor.org/rfc/rfc9367.html

RFC 7801: Кузнечик.

https://www.rfc-editor.org/rfc/rfc7801.html

RFC 9058: MGM.

https://www.rfc-editor.org/rfc/rfc9058.html

RFC 6986: Стрибог.

https://www.rfc-editor.org/rfc/rfc6986.html

RFC 7836: российские функции выработки ключей.

https://www.rfc-editor.org/rfc/rfc7836.html

RFC 8449: ограничение размера записей TLS.

https://www.rfc-editor.org/rfc/rfc8449.html

RFC 2119 и нормативные уровни требований.

https://www.rfc-editor.org/rfc/rfc2119.html

RFC 8174: применение нормативных ключевых слов.

https://www.rfc-editor.org/rfc/rfc8174.html

RFC 8446: TLS 1.3.

https://www.rfc-editor.org/rfc/rfc8446.html

XML 1.0: синтаксис.

https://www.w3.org/TR/xml/

RFC 9293: TCP.

https://www.rfc-editor.org/rfc/rfc9293.html

RFC 3551: форматы PCMA и GSM для RTP.

https://www.rfc-editor.org/rfc/rfc3551.html

RFC 3550: RTP.

https://www.rfc-editor.org/rfc/rfc3550.html

3 Термины, определения и сокращения

3.1 Обязательные требования выражаются ключевыми словами «ДОЛЖЕН», «ДОЛЖНА», «ДОЛЖНО», «ДОЛЖНЫ», «НЕОБХОДИМО», «ТРЕБУЕТСЯ»; запреты — «НЕ ДОПУСКАЕТСЯ», «ЗАПРЕЩАЕТСЯ», «НЕ ДОЛЖЕН» с соответствующим согласованием. «РЕКОМЕНДУЕТСЯ» означает рекомендацию, отступление от которой требует обоснования; «ДОПУСКАЕТСЯ» и «РАЗРЕШАЕТСЯ» — разрешенную возможность. Ключевые слова в нормативных положениях выделяются прописными буквами; это правило оформления настоящего документа. Сочетание «НЕ ТРЕБУЕТСЯ» означает отсутствие обязанности, а не запрет. Установленные форматы, диапазоны, состояния и правила в повествовательной форме также обязательны, если прямо не обозначены как рекомендации или примеры. Обязательность относится

к указанному профилю и проверяется при оценке соответствия реализации.

3.2 АРМ - автоматизированное рабочее место управления системой оповещения любого производителя, включая его серверные компоненты. Совместимая реализация ДОЛЖНА соблюдать требования заявленных профиля и редакции. nodeId обозначает логический узел; экземпляр серверного процесса, защищенное подключение и оператор имеют отдельные идентичности. Конфигуратор является отдельной ролью, а не обязательной частью конкретного АРМ.

3.3 УУ - устройство управления: участник сетевого обмена, который принимает команды, ведет сеансы и взаимодействует с оконечным устройством (ОУ). ВАУ и Сирена обозначают функциональные типы ОУ, а не модели изделий. УУ может быть отдельным контроллером или входить в состав ОУ; разделение сетевой и исполнительной частей сохраняется в модели протокола.

Таблица 2 - Термины, определения и сокращения

Термин	**Краткое определение**
АРМ / УУ / ОУ	Узел управления / сетевой исполнитель команд / оконечное оборудование.
ВАУ / Сирена	Функциональные типы оконечного оборудования.
targetId / deviceId	Логическая цель в системе / идентичность физического сетевого устройства.
principalId / controlEpoch	Участник управления / непрозрачный идентификатор текущего владения.
capabilityRevision / stateRevision	Ревизия каталога возможностей и точек / ревизия наблюдаемого состояния.
sourceKind / evidence	Вид источника показания / уровень доказательства физического результата.

4 Архитектура: АРМ, УУ и ОУ

4.1 Короткие команды экономят канал связи. XML-сообщение занимает не более 1024 байт и передается своей фактической длиной, без дополнения до предела. Сирены и заранее загруженные сообщения работают без непрерывной передачи звука. Для конечного канала от 64 кбит/с предусмотрены отдельные бюджеты управления и сжатой речи; транспортные расходы учитываются сверх размера команды.

4.2 Один порядок управления подходит для оборудования разных производителей: сирен, речевых усилителей и контроллеров. Общие ко-

манды, состояния и ошибки позволяют развивать систему поэтапно и выбирать оборудование по функциям, надежности и стоимости.

4.3 Протокол учитывает ограниченные ресурсы контроллеров. Размеры сообщений, число ожидающих запросов и объемы очередей ограничены, а дополнительные возможности объявляются отдельно. Изготовитель реализует необходимые функции без обязанности включать все расширения.

4.4 Защищенный режим использует российские криптографические алгоритмы. Отдельный секрет каждой пары «устройство — участник» ограничивает последствия утраты одного ключа. Настройка «Только защищенный канал» запрещает открытое управление; выбранный режим виден оператору.

4.5 Оператор различает прием команды, подтвержденное начало работы и остановку оборудования. Результат сопровождается причиной, доступными физическими подтверждениями и сведениями об авариях. Остановка оператором не отменяет подтверждение уже состоявшегося звучания.

4.6 Потеря ответа не превращает повтор команды в повторный запуск. После обрыва связи можно установить состояние исполнения и адресно остановить его. Поведение при потере управления определяется заранее и не зависит от догадок пульта о состоянии оборудования.

4.7 Одноадресная передача служит основой связи с каждым устройством. Отдельный групповой речевой профиль позволяет передавать один поток многим получателям на общих участках сети, сохраняя индивидуальные права, подготовку и результаты. Это уменьшает нагрузку источника и общих каналов при массовом оповещении.

4.8 Опросы, контроль линий и питания, подготовка сообщений, настройка и обслуживание используют общие правила. Машинные схемы, проверяемые примеры и контрольные векторы помогают находить расхождения до подключения оборудования.

4.9 Назначение команд и подтверждений связано с задачами систем оповещения: проверка готовности, задействование, передача речи, прекращение работы и отчет по получателям. Разбор ГОСТ Р 42.3.05-2023 определяет границы сопряжения и позволяет включать протокол в проект без подмены требований обязательного интерфейса заявлением об общей совместимости.

Таблица 3 - Архитектура: АРМ, УУ и ОУ

Уровень	**Назначение**
АРМ	Сценарий и оператор
Сетевая часть УУ	Протокол, состояние и планировщик
Часть взаимодействия с ОУ	Управление и наблюдение
Взаимодействие с ОУ	Способ взаимодействия выбирает изготовитель; ВАУ, Сирена и другие профили ОУ

Таблица 4 - Архитектура: АРМ, УУ и ОУ

Граница	**Ответственность**
Внешние сервисы / межузловой обмен АРМ	Предупреждения, сценарии, группы, маршруты, происхождение запуска, делегирование, полномочия оператора и итог по получателям.
АРМ / Ethernet-часть УУ	Защищенное соединение, команды, собственные запросы прибора, ответы, события, управление сеансом и наблюдаемость.
Исполнитель УУ / драйвер ОУ	Одинаковые предметные действия для разных производителей, привязка ресурсов и перевод наблюдений в общую модель.
Взаимодействие УУ / ОУ	Управляющие воздействия и наблюдения состояния оконечного оборудования. Способ взаимодействия выбирает изготовитель. Функциональные профили: ВАУ и Сирена.

4.10 Разделение сетевой и исполнительной частей является функциональным и не устанавливает число плат, микроконтроллеров или отдельных изделий. Физический и программный способ взаимодействия УУ с усилителем, сиреной или другим ОУ выбирает изготовитель. ДОПУСКАЕТСЯ последовательный интерфейс, иная цифровая шина, дискретные сигналы или взаимодействие внутри одного микроконтроллера. Наличие COM-порта, отдельной платы управления и отдельного байтового протокола УУ–ОУ НЕ ТРЕБУЕТСЯ. Независимо от выбранного способа сохраняются требования к управлению ресурсами, срокам, сопоставлению действия с результатом и подтверждению начала и вы-

ключения. Для работы АРМ знание внутреннего интерфейса изготовителя НЕ ТРЕБУЕТСЯ. Управление и передача звука описываются отдельно; наличие управляющего интерфейса не означает передачи звука по нему.

4.11 Состояния контроллера, взаимодействия с ОУ и самого ОУ различаются в пределах объявленных возможностей наблюдения. Доступный по сети УУ может иметь недоступный исполнительный тракт. Встроенное исполнение не требует имитации отдельного порта или обмена кадрами; граница наблюдения и достоверность результата указываются в профиле оборудования. Замена драйвера, параметров внутреннего интерфейса или ОУ меняет bindingGeneration и каталог возможностей; во время активного исполнения такая замена запрещена.

4.12 Каталог различает адресуемый выход и общие ресурсы: усилитель, декодер, источник звука, общую шину управления и переключатель входов. Для них объявляются группы конфликтов. Несколько линий не означают независимое воспроизведение на каждой; выбор отдельных линий ДОПУСКАЕТСЯ только при объявленной возможности управления ими.

5 Владение и право прерывания

5.1 Прерывать действующее оповещение может текущий исполняющий узел с соответствующими правами оператора либо вышестоящий узел, который запустил именно это оповещение через нижестоящий. Высота в иерархии, приоритет и знание runId сами по себе такого права не дают.

Таблица 5 - Владение и право прерывания

Ситуация	**Результат**
А запустил через Б; А отменяет свой запуск	Б проверяет сохраненное делегирование и отменяет только ветви этого запуска.
Б запустил самостоятельно; А выше в иерархии	А не получает права остановить этот независимый запуск.
Б как текущий узел останавливает исполняемый у него запуск	Проверяются права оператора Б и точная ссылка на исполнение.
Внешний сервис передает originNodeId или высокий приоритет	Поля сервиса не становятся доверенными полномочиями.
Резервная копия АРМ подключается с общим ключом	Общий ключ не доказывает право активной копии. Нужен механизм исключения двух владельцев.

5.2 АРМ сохраняет originNodeId, runId, branchId и путь делегирования path из node-alert/2; идентичность каждой ветви задается тройкой originNodeId/runId/branchId, отдельного delegationId нет. Прием поручения, изменение ветви, исходящая доставка и аудит связываются локальной транзакцией. Отмена, пришедшая до запуска, оставляет запрет позднего запуска до истечения срока поручения с запасом на часы.

5.3 УУ в базовом профиле доверяет аутентифицированному непосредственному владельцу и проверяет конкретный сеанс. Проверка полной межузловой цепочки на приборе и сквозная подпись поручения являются отдельным расширением. Защита Ethernet-канала не защищает от захваченного доверенного АРМ.

5.4 Допуск АРМ к устройству и владение конкретным исполнением проверяются раздельно. Несколько АРМ могут иметь право управления одним УУ, но это не дает каждому права отменять чужой сеанс. Чтение

состояния, обнаружение прибора и наличие его карточки в АРМ сами по себе не предоставляют владение.

5.5 IP-адрес определяет сетевой маршрут, а не владельца. Смена адреса того же участника требует подтверждения идентичности и control.resume с прежним контекстом; новый адрес не обнуляет историю. Передача управления другому участнику ДОПУСКАЕТСЯ только по правилам профиля после завершения исполнения и подтверждения допустимого состояния ресурсов. Неизвестное выключение не считается свободным устройством. Перезапуск АРМ без сохраненного контекста не дает права присвоить продолжающийся сеанс.

5.6 Передача права говорить, квитирование аварии, чтение состояния и администрирование не передают владение оповещением. Сброс, перезагрузка, отзыв медиапотока и изменение маршрута не могут использоваться для обхода права прерывания. Встроенная аппаратная защита от аварии действует по профилю ОУ и регистрируется с собственной причиной; она не выдается за операторскую отмену.

Состояния владения

5.7 free

control.acquire выдает одну эпоху и прикрепляет канал при допустимом состоянии ресурсов.

5.8 held

Один principalId и instanceId, одна эпоха и один прикрепленный канал. Опрос наблюдателя владение не получает и не продлевает.

5.9 detached

Канал потерян, эпоха и история сохранены. Тот же экземпляр восстанавливается через resume. До истечения lease другой участник получает busy при наличии исполнения; свободное владение уступает real по правилу обслуживания ниже.

5.10 draining

Lease истек во время исполнения: применяется stop/finish. Новый session.start запрещен, эпоха сохраняется для адресной отмены и чтения результата.

5.11 blocked

Выключение неизвестно. Диагностика и адресная остановка своего известного исполнения разрешены; новый владелец и session.start запрещены.

5.12 released

Явное освобождение, истечение lease/idleDeadlineMs, конец технического теста либо передача свободного владения real при подтвержденном off. Старый acquire не выдает управление повторно.

Выдача управления и повторы

5.13 acquireSeq - uint64 > 0, монотонный номер попытки в области principalId/deviceId/bootId. АРМ ДОЛЖЕН сохранить его и instanceId до отправки. УУ хранит для каждого допущенного principalId старший acquireSeq, нормализованный запрос и его последнее решение до смены bootId. Повтор того же номера и содержимого возвращает ту же controlEpoch; изменение instanceId или purpose дает conflict; меньший номер дает resultExpired. Новый больший номер расходуется и при busy. Отказ аутентификации, старый bootId и неверная оболочка номер не расходуют. Повтор потерянного busy остается busy; новая попытка требует нового номера.

5.14 Ответ acquire содержит current: существует ли выданное владение сейчас. Повтор после освобождения возвращает прежнее решение с current=false и не выдает эпоху заново. Повтор acquire с другого соединения только читает решение: для прикрепления нужен resume. Тем самым потеря ответа acquire не требует угадывать эпоху или останавливать ОУ.

5.15 В повторе acquire неизменны решение о выдаче, acquireSeq, instanceId, purpose и назначенная эпоха. current, highWater, attachSeq и leaseRemainingMs отражают состояние на момент ответа. Восстановление связи продлевает lease только после успешного resume; оно не отменяет уже принятую остановку и не возвращает stopping в running.

Переподключение и дублер

5.16 instanceId идентифицирует единственный активный экземпляр АРМ и сохраняется с контекстом. resume несет возрастающий attachSeq. После отсоединения прежнего канала УУ атомарно сохраняет новый attachSeq и прикрепляет новый канал. При живой прежней привязке другой канал получает busy/detail=ownershipHeld. Точный повтор на том же канале идемпотентен; равный номер с другого канала получает staleAttachment, меньший номер также отклоняется. Потеряв ответ, экземпляр использует следующий сохраненный attachSeq. Изменение IP не меняет instanceId, эпоху или историю.

5.17 control.release обязателен в ядре. После него повтор той же команды на исходном либо новом допущенном канале той же идентичности по правилам раздела 9 получает сохраненный ответ, пока запись входит в кеш завершенной эпохи; он не восстанавливает владение. Проверка точного повтора release выполняется до отказа по released, но после проверки principalId и bootId. При вытеснении записи возвращается resultExpired. Новая выдача управления не принимает изменяющие команды прежней эпохи.

5.18 Базовый профиль использует одного активного писателя для instanceId. Для резервирования ТРЕБУЕТСЯ внешнее ограждение прежней копии: ее питание либо все пути управления к УУ блокируются независимым от нее средством и подтверждаются до передачи контекста резерву. Ограждение сохраняется до возврата узла под контроль координатора. Отсутствие доказанного ограждения запрещает автоматический

resume резервом. Один лишь общий ключ, attachSeq или решение самой копии о своей активности не являются доказательством. Такая схема сохраняет происхождение запуска и не передает чужое оповещение другому principalId.

Потеря связи и таймеры

5.19 Потеря транспорта и истечение lease - разные события. Закрытие прикрепленного канала само по себе не сокращает и не продлевает уже выданный lease. Политика onControlLoss применяется при его истечении; немедленное аннулирование действующих полномочий по разделу 14 является самостоятельным событием. Отказ ОУ и собственные сроки исполнения обрабатываются независимо. Состояния владения ниже не заменяют состояния сеанса starting/running/stopping/finished.

5.20 free / released

Событие: Успешный новый acquire

Действие: Выдать эпоху и прикрепить канал.

Новое владение: held

Судьба таймеров: Lease начинается при выдаче управления; таймер сеанса еще не запущен.

5.21 held

Событие: Закрытие прикрепленного соединения, включая таймаут сборки или отправки

Действие: Отсоединить канал, сохранить эпоху и принятое исполнение.

Новое владение: detached

Судьба таймеров: Lease идет от последнего продления; все сроки сеанса продолжаются.

5.22 held

Событие: Разрешенный запрос прикрепленного владельца до истечения lease

Действие: Обработать запрос по его контракту; lease продлевается только для команд, которым это разрешено. Отклоненный запрос, TCP ACK и запрос наблюдателя не продлевают lease.

Новое владение: held

Судьба таймеров: Предусмотренное контрактом продление начинает lease заново. Сроки сеанса неизменны; idleDeadlineMs меняется только при явно описанных переходах, включая проверку полного файла по разделу 16.

5.23 detached

Событие: Успешный resume до истечения lease

Действие: Атомарно прикрепить новый канал. Уже начатая остановка продолжается.

Новое владение: held

Судьба таймеров: Lease начинается заново; prepareTimeoutMs, durationMs, maxDurationMs и сроки остановки не сбрасываются.

5.24 held / detached

Событие: Lease истек; есть конечное исполнение

Действие: Применить сохраненную stop/finish. При stop начать stopping; при finish продолжить принятое исполнение. Если stopping уже начат, продолжить его.

Новое владение: draining

Судьба таймеров: Lease больше не разрешает session.start. Сроки исполнения не продлеваются; сроки остановки начинаются только при первом переходе в stopping.

5.25 held / detached

Событие: Lease истек; исполнения нет и off подтвержден

Действие: Освободить владение; сохранить доступный результат и историю по их срокам.

Новое владение: released

Судьба таймеров: Lease прекращается; срок хранения результата не меняется.

5.26 held / detached

Событие: Применен отзыв полномочий, на которых основано текущее владение

Действие: Немедленно аннулировать право управления и выполнить stop независимо от настройки onControlLoss=finish. При активном исполнении сохранить его происхождение; без исполнения освободить владение только при подтвержденном off.

Новое владение: draining при исполнении; released при подтвержденном off; иначе blocked

Судьба таймеров: Lease больше не разрешает команды независимо от остатка. Уже идущие сроки исполнения/остановки не сбрасываются. Отозванные полномочия не допускают resume.

5.27 draining

Событие: Транспорт восстановлен; успешный resume прежнего владельца

Действие: Прикрепить канал для адресной отмены и чтения. Запрет нового session.start сохраняется, finish не становится новым запуском.

Новое владение: draining

Судьба таймеров: Resume не отменяет факт потери управления и не сбрасывает сроки исполнения/остановки.

5.28 blocked

Событие: Транспорт восстановлен; успешный resume прежнего владельца с достоверным контекстом

Действие: Разрешить диагностику и адресную остановку своего исполнения. Само подключение не подтверждает off.

Новое владение: blocked

Судьба таймеров: Сроки не перезапускаются; снятие блокировки требует процедуры восстановления.

5.29 draining / blocked

Событие: Исполнение завершено, off подтвержден, результат надежно сохранен и восстановление завершено

Действие: Освободить эпоху. Следующее управление требует нового acquire.

Новое владение: released

Судьба таймеров: Сохранение результата продолжается независимо от освобождения эпохи.

5.30 held

Событие: Обычный сеанс завершен при действующем lease и подтвержденном off

Действие: Сохранить результат; право владельца остается до lease или idleDeadlineMs.

Новое владение: held

Судьба таймеров: idleDeadlineMs=finishedAtMs+10000; lease не продлевается. Технический сеанс после сохранения результата освобождает эпоху сразу.

5.31 Любое

Событие: Выключение неизвестно либо не удалось надежно закрыть запись воздействия

Действие: Запретить новый session.start и выдачу управления; сохранить происхождение и выполнять восстановление.

Новое владение: blocked

Судьба таймеров: Таймаут и повторное подключение не снимают блокировку.

5.32 При now >= срока lease сначала применяется его истечение, затем обрабатывается поступивший запрос: запоздалый ping/resume не возвращает draining в held. В draining/blocked успешный resume может обновить привязку и срок связи, но этот срок не дает права нового session.start. Закрытие уже отозванного канала или канала наблюдателя не меняет владение. Само открытие TCP/TLS без успешного resume не прикрепляет владельца. После released прежний resume получает staleEpoch; повтор acquire только читает прежнее решение с current=false.

5.33 service-handover

Входные условия: Свободная эпоха maintenance, идет загрузка; разрешенный acquire real получен при t=0.

Обязательный исход: Шаг обслуживания завершен до 100 мс, решение и ответ новой выдачи до 500 мс; старая эпоха не фиксирует очереди; подтвержденные байты сохранены.

5.34 idle-heartbeat

Входные условия: Свободная эпоха real выдана при t=0; heartbeat каждую 1000 мс, START отсутствует.

Обязательный исход: Освобождение не позднее t=10000 мс; heartbeat не продлевает idleDeadlineMs.

5.35 active-owner

Входные условия: Действующее или неопределенное чужое исполнение и acquire real другого участника.

Обязательный исход: busy; действующее чужое исполнение не прерывается.

Свободное владение, обслуживание и реальная тревога

5.36 control.acquire содержит обязательный control.purpose=real|exercise|technical|maintenance. Назначение сохраняется в эпохе и ответах acquire/resume; session.start.purpose ДОЛЖНО совпадать с ним.

maintenance не разрешает session.start. Назначение real требует отдельного права реального оповещения, а не выводится из административной роли. До первого session.start свободная эпоха живет не более 10000 мс от acquire; после завершения обычного сеанса - не более 10000 мс от finished. Проверка связи, чтение и повторы этот idleDeadlineMs не продлевают. Для maintenance действуют только два исключения раздела 16: новый устойчивый префикс bulk/1 освежает idleDeadlineMs, а выполняющаяся validating/verifying использует фиксированный срок проверки до 60000 мс. Оба исключения сохраняют независимый lease и отзыв в пользу real. Истечение освобождает только эпоху без активного/неопределенного исполнения при подтвержденном off. Сеансы technical и exercise освобождают эпоху сразу после off и фиксации результата.

5.37 При control.acquire purpose=real другого допущенного участника и отсутствии действующего/неопределенного сеанса УУ отзывает свободную эпоху exercise/technical/maintenance на ближайшей границе сервисного шага. Шаг не превышает 100 мс; решение о выдаче новой эпохи и ответ укладываются в ordinaryResponseMs=500 мс от полного запроса. До ответа новый владелец не действует. Сначала аннулируются полномочия старого соединения, затем публикуется новый controlEpoch; старые очереди повторно проверяют эпоху перед фиксацией любого изменения. Подтвержденный блок загрузки сохраняется, неподтвержденный не считается принятым. Отзыв не запускает rollback действующей конфигурации и не стирает подготовленный файл.

5.38 Действующее оповещение, в том числе технический сеанс, этим правилом не прерывается. При busy для разрешенного real УУ фиксирует очередь намерений по principalId до 35000 мс от первой попытки. Повторы и новые acquireSeq того же участника не сдвигают срок и место; чужой instanceId не наследует место. Порядок FIFO, при равном времени - лексикографический principalId. Новая выдача свободного

владения ДОПУСКАЕТСЯ только первому живому намерению; maintenance/exercise/technical не опережают его. Клиент real повторяет acquire с новым acquireSeq через 500..2000 мс после получения busy, пока его срок актуален; следующий запрос не отправляется до ответа на предыдущий. В alarmTransitionMaxMs входят ожидание этого интервала, завершение уже отправленного запроса, доставка и обработка следующей попытки; для допуска теста используется их общая верхняя оценка по разделу 12. Возвращенный busy не превращается в выдачу самопроизвольно. При отсутствии нового полного допустимого acquire в течение 10000 мс намерение удаляется; исходный срок очереди 35000 мс не продлевается. Потеря соединения и смена bootId удаляют намерение. Число записей ограничено числом заранее допущенных участников. Уже выданная real-эпоха сохраняется до собственного idleDeadlineMs или завершения сеанса.

5.39 Граница доступности: отзыв свободного сервисного владения занимает не более 500 мс локально; доставка запроса/ответа и следующего session.start добавляет бюджет сети. При занятой свободной real-эпохе добавляется не более оставшихся 10000 мс idleDeadlineMs и период повторной попытки. Неисправность хранилища, неизвестное выключение и активное чужое исполнение не обходятся приоритетом. Операции, выводящие сам контроллер из готовности (firmware.commit и network.stage), разрешены только при местном разрешении обслуживания и готовом независимом тракте тревоги, указанном в конфигурации развертывания. Разрешение и готовность подтверждаются независимыми входами, не устанавливаются сетевой сервисной командой; без них unsafeState. На время перезагрузки/смены сети тревога обслуживается резервным трактом.

Допуск учений и доступность реального оповещения

5.40 Техническая проверка mode=test либо purpose=technical имеет maxDurationMs не более 30000. Полноценные учения purpose=exercise допускают сирену, запись, программу или поток общей длительностью до 1080000 мс, включая подготовку и паузы. Учения всегда применяют onControlLoss=stop, stopBeginMs не более 100 и stopConfirmMs не более 3000; отсутствие таких физических возможностей дает unsafeState до воздействия. После off и сохранения результата эпоха автоматически освобождается. Следующий учебный запуск требует нового acquire и не обходит ожидающую тревогу.

5.41 Сохранение результата и освобождение эпохи после off занимают не более 500 мс. Для технической проверки и учения до 30000 мс применяется единый бюджет раздела 12: 30000+3000+2000=35000 мс до принятия нового session.start первого ожидающего real. Последние 2000 мс включают сохранение результата, освобождение, ожидание повтора, сетевой обмен и обработку acquire и session.start. Предел сетевой доставки 1500 мс сам по себе не обеспечивает этот срок. Если сумма выбранных пределов больше, допустимая длительность теста или короткого учения уменьшается. Длительное учение допускается только при независимом готовом тракте реального оповещения, не занятом учением, с alarmTransitionMaxMs не более 35000 мс. При отсутствии такого тракта длительность exercise ограничивается 30000 мс с соблюдением общего бюджета. Самостоятельная тревога не отбирает чужую эпоху; независимость ресурсов и путей управления проверяется до допуска длительного учения. Если проект требует меньшего срока, применяется именно он. Неподтвержденный off и отказ Flash обрабатываются предусмотренным резервом.

5.42 Живая привязка означает одновременно открытое допущенное соединение выбранного режима (аутентифицированное при tls) и now < leaseDeadlineMs. Lease длится 30000 мс от последнего допу-

стимого продления; heartbeat или другой разрешенный запрос прикрепленного владельца продлевает его по общим правилам. TCP ACK и запрос наблюдателя не продлевают. Обнаруженное закрытие TLS/TCP либо истечение таймера сборки/отправки немедленно отсоединяет канал: resume разрешен без ожидания остатка lease, но только прежней идентичности и после требуемого ограждения резервной копии. Необнаруженный обрыв остается живой привязкой до указанного предела; УУ не угадывает доступность удаленной копии.

5.43 Плановая смена активной копии при подтвержденном off: прежняя копия выполняет control.release и прекращает управление, затем новая выполняет control.acquire с собственным сохраненным контекстом. После release resume завершенной эпохи запрещен. Передача активного сеанса без новой эпохи: координатор независимо ограждает прежнюю копию, УУ отсоединяет ее канал, резерв с сохраненным разрешенным контекстом выполняет resume; уже начатая остановка не отменяется. Если ограждение не доказано, такой переход запрещен. Для кластера с несколькими УУ отдельные привязки не выбирают единую активную копию: разные копии могут занять разные УУ. Координатор ДОЛЖЕН обеспечить общее независимое ограждение для всего набора УУ до передачи полномочий.

5.44 Прикрепление является ограждением на стороне УУ: каждая изменяющая команда, heartbeat и отложенная сервисная операция повторно сверяют внутренний идентификатор соединения с текущей привязкой непосредственно перед изменением состояния. Старый канал не получает полномочий от знания controlEpoch или большого cmdSeq. Resume с другого канала ДОПУСКАЕТСЯ только после отсоединения прежнего. Пока прежний канал прикреплен и его lease не истек, возвращается busy/detail=ownershipHeld без изменения attachSeq, highWater, привязки и lease. Точный повтор resume на уже прикрепленном канале сохраняет идемпотентность. Истечение lease сначала при-

меняет политику потери управления и отсоединяет старый канал; это не разрешает новый session.start в draining.

5.45 Привязка не выбирает лидера среди копий с одинаковыми секретами. Автоматическая передача сохраненного instanceId резерву ДОПУСКАЕТСЯ только после независимого ограждения прежней копии, установленного разделом 5. До этого резерв может наблюдать, но НЕ ДОЛЖЕН выполнять acquire, resume, session.start или продлевать lease от имени активной копии. Увеличиваемое самой копией число, в том числе attachSeq, не является разрешением арбитра. Для реализации с программным арбитром его проверяемое разрешение и отдельная идентичность каждой копии требуют отдельного согласованного профиля; арбитраж между несколькими УУ не входит в базовый профиль.

6 Идентификаторы и контекст

Таблица 6 - Идентификаторы и контекст

Поле	Тип и смысл
deviceId / bootId	128 бит, 32 строчных hex-символа. Идентичность устройства / текущего запуска прошивки. bootId меняется при перезагрузке. IP не является deviceId.
channelId / controlEpoch	Отдельные 128-битные ID защищенного подключения и периода владения. Переподключение не сбрасывает историю команд текущего владения.
capabilityRevision	Ревизия возможностей и состава каталога точек. Каталог и значения связываются через нее; изменение состава или смысла точки меняет ревизию.
id / to	128-битный ID запроса и ссылка ответа на запрос. Направление обмена входит в область корреляции.

Окончание таблицы 6

Поле	**Тип и смысл**
cmdSeq / startSeq	uint64 в десятичной записи. cmdSeq растет в bootId/controlEpoch; startSeq равен cmdSeq исходного session.start.
sessionId / operationId	128-битные ID принятого сеанса и длительной операции. Идентификатор сеанса назначает УУ.
ouId / bindingGeneration	Локальная привязка ОУ и ее поколение. Это контекст драйвера и наблюдений, а не замена криптографической идентичности.
stateRevision / eventSeq	Версия согласованного состояния и курсор событий в пределах bootId. Значения не сравниваются между загрузками.

6.1 Счетчики uint64 передаются строкой десятичных цифр, без знака, экспоненты и дробной части. В JavaScript используется BigInt или строка, а не Number. При исчерпании счетчика новые действия в его области запрещены; перенос через ноль недопустим. Завершение уже принятого исполнения и чтение результата сохраняются. Новая область создается только штатной сменой контекста после подтвержденного выключения; скрытая перезагрузка ради сброса счетчика запрещена.

6.2 controlEpoch является непрозрачным идентификатором, а не упорядоченным fencing-счетчиком. УУ Обязательно принимать изменяющую команду только при точном совпадении bootId, текущего controlEpoch, владельца и действующего канала. Несовпавшая эпоха отклоняется независимо от лексикографического значения. При control.resume привязка нового канала и отзыв прежнего выполняются

атомарно до разрешения следующего изменения; уже переданные физические действия учитываются при согласовании состояния. Право резервной копии выполнить resume проверяется по правилу ограждения раздела 5, отдельно от знания эпохи.

6.3 Группа фиксируется в плане АРМ до запуска. Добавление прибора в группу не включает его в уже начатую ветвь. Два маршрута к одному УУ требуют дедупликации на уровне координатора.

6.4 Назначение запуска (реальное оповещение, учения или техническая проверка) задается отдельно от физического режима siren/recording/program/stream/test. Учебное оповещение может звучать. Предварительная оценка readiness не имеет воздействия. Назначение передается полем purpose=real|exercise|technical; право каждого назначения хранится отдельно; подмена звучащих учений тихой проверкой или обратная подмена запрещены.

Резерв номеров для остановки и освобождения

6.5 Для cmdSeq установлены три области: обычные номера 1..18446744073709551613, номер завершения 18446744073709551614 и номер освобождения 18446744073709551615. Обычный новый номер превышает highWater не более чем на 4096; пропуски в этом окне допустимы. Последние два номера НЕ ДОЛЖНЫ использоваться для session.start, изменения конфигурации, неизвестного действия или другого обычного изменения. Проверка области номера относится к общей оболочке и предшествует расходованию highWater.

6.6 1..18446744073709551613

Допустимое действие: Любое разрешенное изменяющее действие

Условие расходования: Новый номер в highWater+1..min(highWater+4096,18446744073709551613); предметные отказы сохраняются по общему правилу.

6.7 18446744073709551614

Допустимое действие: Только session.stop или session.cancel

Условие расходования: Все поля и полномочия проверены. При незавершенном или невыключенном собственном исполнении ссылка ДОЛЖНА совпадать с ним. При отсутствии такого исполнения session.cancel ставит запрет позднего session.start, а session.stop ДОПУСКАЕТСЯ только для известного завершенного сеанса.

6.8 18446744073709551615

Допустимое действие: Только control.release

Условие расходования: Нет действующего или неопределенного исполнения, off подтвержден, результат и закрытие журнала сохранены. Расходование номера и освобождение эпохи атомарны.

6.9 Любой отказ проверки зарезервированного номера, включая неверное тело, чужой startSeq/sessionId, неизвестный сеанс и преждевременный control.release, НЕ ДОЛЖЕН изменять highWater, занимать запись дедупликации или продлевать lease. Возвращается invalidArgument для области/формы, forbidden для полномочий, notFound для неизвестного session.stop, conflict с detail=operationStateMismatch для чужого исполнения либо unsafeState для преждевременного control.release. Это специальное правило имеет приоритет перед обычным расходованием номера при предметном отказе.

6.10 После расходования номера завершения новые обычные действия запрещены. Точный повтор session.stop/session.cancel с тем же номером и нормализованным телом сохраняет смысл и доступен до освобождения эпохи; его решение не вытесняется. Повтор не начинает воздействие заново и не продлевает сроки остановки. Исполнитель продолжает ранее принятую остановку и предусмотренное восстановление. При подтвержденном off применяется номер освобождения; следующее владение получается новым acquireSeq. Нумерация никогда не переносится через ноль.

6.11 АРМ ДОЛЖЕН резервировать номер завершения независимо от результата обычной команды. При unknown исходного session.start он использует session.cancel с исходным startSeq; при известном sessionId - session.stop. Одновременные обычные изменения после выдачи отмены запрещены до решения этой гонки. УУ ДОЛЖЕН сохранять возможность принять адресную отмену своего исполнения при highWater в обычной области; значение highWater=18446744073709551615 допустимо только у освобожденной эпохи.

6.12 cmdSeqWindow - обязательное имя limit в device.capabilities, unit=count, value=4096. Клиент хранит номера до передачи и сверяет highWater при восстановлении. Протокол не требует скачков выше окна: операции выдаются последовательно с отдельным резервом отмены. Смена процесса, восстановление резервной копии и откат локального журнала не разрешают произвольный скачок вверх. При утраченном контексте старой эпохи АРМ выясняет исход через чтение, при необходимости прекращает собственное исполнение и получает новую эпоху после безопасного освобождения. Номер ниже highWater не становится новым действием; доступен только сохраненный точный повтор либо resultExpired.

6.13 Окно обычного cmdSeq равно 4096. Для нового обычного номера обязательны cmdSeq > highWater и cmdSeq - highWater <= 4096; вычитание выполняется только после сравнения, без переполнения. При исходном highWater=0 первый номер находится в 1..4096. Известный точный повтор проверяется до окна. Превышение окна дает invalidArgument до расходования номера, изменения состояния, дедупликации и продления lease. Зарезервированные номера адресной остановки и освобождения исключены из окна; их условия раздела 6 сохраняются. После исчерпания обычной области АРМ останавливает свое исполнение, освобождает эпоху и получает новую; счетчик не переносится через ноль.

7 Транспорт и ограниченный XML

7.1 Прикладная модель отделена от транспорта: команды описывают владение, конечное исполнение, отмену и физическое наблюдение. Ограниченный XML выбран для небольших сообщений контроллера, строгого разбора и чтения трасс при наладке. Лимиты являются условиями профиля; их реализуемость на каждом изделии подтверждается отдельно. NETCONF и RESTCONF задают прежде всего управление данными; MQTT задает доставку сообщений; их применение само по себе не определяет эти правила оповещения. Адаптеры таких систем располагаются на границе АРМ и сохраняют предметную семантику.

7.2 Используется постоянное TCP-соединение выбранного режима tls либо plain для управления и отдельный канал медиаданных. Прикладное обрамление после проверки защиты: xmlLength:uint32 BE, затем ровно xmlLength байтов UTF-8. TCP-сегмент не является сообщением. Часть префикса, часть тела и несколько сообщений в одном чтении обрабатываются корректно.

Таблица 7 - Транспорт и ограниченный XML

Ограничение профиля	**Значение**
Размер XML / глубина / число элементов	1024 байта / 4 с учетом корня / 32
Атрибутов на элемент / длина имени	12 / 32 ASCII-байта
Декодированное строковое значение	До 128 байт UTF-8
Сборка неполного сообщения	10000 мс от первого байта кадра, включая префикс длины, без продления каждым байтом
Запросы в полете в проектном профиле	1 обычный запрос, 1 слот session.stop/ session.cancel и 1 слот channel.ping; по одному запросу без ответа на слот
Большие ответы	Страницы с ограничением по байтам; недостаточно ограничивать только число строк

7.3 Базовые корни сообщений: request и response. При явно включенной возможности eventsPush=1 также используются notification и notificationAck. event является записью журнала внутри events в ответе events.read либо в notification. v="1" задает версию оболочки, profile="open-alert-draft/0.21" - точную редакцию проектной схемы. Пространства имен в базовом профиле отсутствуют.

7.4 DTD, пользовательские сущности, внешние ресурсы, CDATA, комментарии, processing instructions, XML-декларация и BOM запрещены.

7.5 Пять стандартных XML-сущностей и числовые ссылки допускаются после проверки символов и размера декодированного значения.

7.6 Повтор атрибута, неверный тип, неизвестное управляющее поле или лишний элемент-одиночка отклоняют весь запрос. Порядок атрибутов и вид кавычек не меняют смысл.

7.7 Неверная длина или синтаксис XML закрывают соединение. Корректный XML с ошибкой команды получает структурированный отказ. Парсер не включает выходы по мере чтения.

XML в примерах показывает прикладное тело. При tls оно передается внутри защищенного потока, при явно выбранном plain — непосредственно по TCP. Настройка securityPolicy УУ определяет допустимость открытого режима; порт и направление задаются при вводе.

Обрамление и приоритет отмены

7.8 Пределы этой таблицы относятся к сообщениям АРМ-УУ после расшифрования. АРМ-АРМ, пакет конфигурации и внешнее предупреждение имеют собственные профили и бюджеты; целый CAP-документ не помещается в команду УУ автоматически. Страница является самостоятельным ограниченным ответом; следующая страница запрашивается отдельно.

7.9 УУ и АРМ ДОЛЖНЫ обслуживать дополнительный слот session.stop/session.cancel независимо от ожидания обычного ответа. Ответы могут идти в другом порядке и связываются через to. Отмена имеет приоритет перед следующей страницей, сервисным блоком и новым запуском. Уже передаваемый кадр TCP не прерывается: его максимальный размер и время передачи входят в бюджет остановки. УУ ДОЛЖЕН ограничивать исходящую очередь и неблокирующе принимать входящие команды; медленное чтение клиентом не блокирует исполнитель. Истечение таймера передачи закрывает канал и вызывает переход по разделу 5: прикрепленный владелец переходит в detached, оставшийся lease продолжает действовать. Сам таймаут не сокращает lease и не применяет onControlLoss раньше его истечения; отдельное событие от-

зыва полномочий определено в разделе 14. Закрытие канала не подтверждает исполнение session.stop.

7.10 После готовности управляющего транспорта protocol.hello ДОЛЖЕН быть принят за 10000 мс; в plain срок начинается с принятия TCP, в tls - с завершения TLS. После hello отсутствие полного допустимого прикладного кадра в течение 30000 мс закрывает только это соединение. Частичные байты, TCP ACK и пакеты речи не продлевают срок. Постоянный клиент при отсутствии иных запросов посылает channel.ping не реже одного раза в 10000 мс. Такой ping без controlEpoch подтверждает живость канала и не продлевает владение. Прикрепленный владелец сохраняет отдельный heartbeat с эпохой каждые 5000 мс, в том числе во время чтения и передачи речи.

Таймеры и бюджеты

7.11 heartbeatIntervalMs / controlLeaseMs

5000 / 30000 мс. Lease продлевает только принятый запрос прикрепленного владельца; TCP ACK и запрос наблюдателя его не продлевают.

7.12 ordinaryResponseMs

До 500 мс от полного проверенного кадра до постановки ответа на отправку. Длительная операция возвращает accepted и идентификатор; ожидание физического результата не удерживает обычный слот.

7.13 cancelAdmissionMs

До 100 мс от полного проверенного session.stop/session.cancel до решения исполнителя. Это локальная задержка, без времени сетевой доставки.

7.14 frameAssemblyMs / writeProgressMs

10000 / 10000 мс. Первый таймер начинается с первого байта кадра. Второй начинается при готовности ответа, включает ожидание в очереди и передачу целого кадра защищенному транспорту; буфер TLS и

фактическая отправка также ДОЛЖНЫ укладываться в этот бюджет. Продвижение отдельных байтов срок не продлевает.

7.15 prepareTimeoutMs

От acceptedAtMs принятого session.start до выполнения критерия running для всех выбранных ресурсов. Включает подготовку, ожидание драйвера и подтверждений ОУ; доставка ответа accepted АРМ не является началом. Истечение в starting инициирует остановку с причиной prepareTimeout.

7.16 durationMs (siren)

От перехода сеанса в running; включает все паузы intermittent. Время starting не входит в durationMs, но входит в maxDurationMs. Истечение инициирует stopping без ожидания конца цикла сигнала.

7.17 maxDurationMs

От acceptedAtMs, включая starting и паузы. При истечении во время starting/running инициируется stopping с причиной maxDurationExceeded. Это предел до начала остановки; подтверждение off имеет отдельный бюджет.

7.18 stopBeginMs / stopConfirmMs

Обязательные значения паспорта ОУ: 1..1000 и 1..10000 мс, stopBeginMs <= stopConfirmMs; для technical не выше 100/3000 мс. Сроки идут от решения stopping до выдачи физической команды и до подтверждения выключения. Истечение второго дает stopUnconfirmed.

7.19 recoveryTimeoutMs

От входа в reconciling после загрузки или обнаруженного отказа. Возврат сети, опрос и повтор команды не начинают срок заново; истечение требует localInterventionRequired.

7.20 resultRetentionMs

От перехода в finished в текущем bootId; новый снимок или reconnect срок не перезапускает. Граница гарантии после reboot определена в разделе 8.

7.21 Все сроки считаются по монотонным часам УУ. acceptedAtMs фиксируется однократно при принятии session.start в сохраняемом намерении; затраты его фиксации не дают права перенести срок вперед. УУ проверяет истекшие сроки перед новым физическим действием. При now >= срока таймер истек: позднее подтверждение не дает заново включить ОУ. Если одновременно истекли prepareTimeoutMs и maxDurationMs, применяется maxDurationExceeded. Повтор session.start, session.stop/session.cancel и resume не перезапускает подготовку, длительность или уже начатую остановку. Для остановки по внутреннему сроку или потере управления stopBeginMs/stopConfirmMs отсчитываются от решения перейти в stopping. Физический сигнал может начаться до подтверждения running; задержка подтверждения и прекращения сигнала входит в паспорт ОУ, durationMs не обещает точную акустическую длительность.

7.22 Кроме обычного слота и слота session.stop/session.cancel существует один отдельный слот channel.ping. Каждый слот допускает не более одного запроса без ответа; повтор использует тот же id. При одновременной готовности порядок обработки: отмена, проверка связи, обычная работа. Проверка связи не обходит байты, уже поставленные в TCP; очередь отправки ограничена одним текущим кадром и по одному ожидающему кадру на слот. Запрос страницы и сервисная команда не могут бесконечно удерживать обычный слот. Байты файлов передаются по отдельному каналу bulk/1 и в управляющую очередь не помещаются.

7.23 Полная задержка отмены считается как доставка полного кадра + cancelAdmissionMs + stopBeginMs/stopConfirmMs. TCP сохраняет порядок байтов, поэтому потерянные предыдущие байты задерживают session.stop. Паспорт развертывания ДОЛЖЕН задавать максимальные RTT, потери, время восстановления и допустимый сетевой перерыв, в которых проверяется предел. При полном отказе сети удаленная остановка не гарантируется: действует сохраненная stop/finish и

maxDurationMs. Требование остановки независимо от сети обеспечивается местной защитой или независимым каналом с собственными полномочиями и подтверждением.

Кодирование в пределах размера кадра

7.24 maxXmlBytes=1024 ограничивает байты всего XML в UTF-8 после расшифрования, включая оболочку, имена, кавычки и XML-сущности; четырехбайтовый префикс длины в этот предел не входит. Лимиты 12 атрибутов, 32 элементов и глубины 4 сохраняются. Передатчик ДОЛЖЕН измерять фактическую сериализацию до отправки. Отступы в примерах не являются обязательными байтами. Не допускаются обрезание чисел, идентификаторов, текста или частичная передача команды ради размера.

7.25 Если явно не задан более узкий тип, локальные идентификаторы resourceId, pointId, sourceId, conditionId, criterionId, testId, contentId, programId, коды и значения token содержат 1..32 символа [A-Za-z0-9_.:-]. Идентификаторы экземпляров, запросов, сеансов и снимков содержат ровно 32 строчных hex. cursor содержит 1..32 символа [A-Za-z0-9_.:-]; это непрозрачное значение, клиент его не вычисляет. Числа кодируются каноническими десятичными строками своего диапазона. Ограничение произвольного строкового значения 128 декодированных байт не разрешает идентификатор длиной 128 байт.

7.26 Служебные идентификаторы не требуют XML-экранирования. Произвольный человекочитаемый текст остается UTF-8: при двойных кавычках передатчик заменяет только &, < и двойную кавычку стандартными сущностями; апостроф передается буквально, > ДОПУСКАЕТСЯ кодировать как >. config.get передает параметр записями entry: id (положительный порядковый номер в канонической десятичной записи), facet=path|value, part=1..3, parts=1..3, type=bool|enum|int|counter|string, text до 64 декодированных байт UTF-8. Обе facet обязательны; каждая со-

держит все части 1..parts без повторов, type одинаков. Соединение text по part восстанавливает путь и значение до 128 байт каждое, без обрезания. Разделение проходит только между кодовыми точками UTF-8; короткая строка имеет одну часть, пустое значение — одну часть text=""; путь не пустой. Уникальный id связывает параметр во всем снимке. Это формат частей значения, а не фрагментация XML. profiles.get parameter использует name/type/part/parts/value с тем же разбиением значения до 128 байт на части по 64 байта; остальные поля привязки профиля сохраняются. Сумма частей каждой facet или parameter не превышает 128 байт; третья часть нужна только при границе многобайтового символа. Размеры строк проверяются в байтах, а не в символах. device.describe использует version, manufacturer и softwareDeveloper как token до 32 ASCII-символов; последние два являются идентификаторами фактических изготовителей, их полные наименования приведены в паспорте. Обязательный элемент info protocolName="Открытый протокол систем оповещения" обозначает только название протокола; обязательная маркировка компанией отсутствует.

7.27 Команда с допустимыми максимальными значениями всех полей и четырьмя ресурсами ДОЛЖНА помещаться целиком. Ответы делятся только между целыми записями; сводка первой страницы, execution, completion, fault и timeline передаются отдельными страницами при необходимости. Запись с оболочкой и самым длинным cursor ДОЛЖНА помещаться в 1024 байта. Проверка максимальных размеров всех команд и примеров входит в комплект wire-budget текущей редакции. Она не заменяет проверку конкретного кодировщика изделия.

7.28 После неудачного TCP/TLS-соединения или неуспешного resume клиент применяет отдельный для deviceId счетчик n, начиная с 0: перед следующей попыткой ожидает случайное равномерное время от B/2 до B включительно, B=min(30000, 500*2^min(n,6)) мс, затем увеличивает n. Одновременно ДОПУСКАЕТСЯ одна попытка управления этим

УУ от экземпляра. Счетчик сбрасывается после 60000 мс устойчивого аутентифицированного обмена. Новая команда оператора и смена IP не сбрасывают задержку. Отказ forbidden прекращает автоматические попытки до исправления полномочий; несовместимый профиль требует изменения настройки. Уже прикрепленный канал продолжает heartbeat и приоритетный session.stop без этого ожидания. Эти интервалы учитываются в бюджете восстановления; они не являются гарантией доставки отмены при отказе сети.

7.29 Предел команды 1024 байта — 1 КиБ, а не 1 МБ. Он относится только к UTF-8 XML. Четырехбайтовая рамка увеличивает прикладную передачу до 1028 байт. record_size_limit=512 включает внутренний тип TLS, поэтому одна запись без padding содержит не более 511 прикладных байт. Максимальный кадр требует как минимум трех записей, но разбиение может дать больше; приемник не предполагает совпадения кадра и TLS-записи. На каждую защищенную запись добавляются 5 байт TLS-заголовка, 1 байт внутреннего типа и 16 байт MGM-метки. При трех записях и отсутствии padding получается 1094 байта TCP payload; TCP/IP, ACK, повторы и Ethernet учитываются дополнительно. Шифротекст и открытый кадр не требуют одного общего буфера: разбор поточный, пиковая память подтверждается для конкретной реализации.

7.30 В этой редакции сетевые адреса профиля — IPv4. Это явное ограничение совместимости, не свойство криптографии и не запрет разработать отдельный IPv6-профиль. Преобразование IPv6 в строку IPv4, усечение адреса и неявный выбор маршрута запрещены; неподдержанный адрес дает invalidArgument до воздействия. Расширение адресации требует схемы, области link-local, правил маршрута и проверки размеров, а не только разрешения двоеточия в поле.

7.31 Структурная XML-схема wire-0.21.xsd, реестр wire-contract-0.21.json и проверка open_protocol_xml.py входят в нормативный комплект. XSD проверяет допустимые имена, лексические типы

и словари; отдельная проверка применяет пределы UTF-8 в байтах, сочетания полей, формы каждой команды и правила контекста. Совместимость не устанавливается одной успешной проверкой XSD. Примеры имеют scenarioId и начальное состояние; одинаковые идентификаторы в разных самостоятельных сценариях не объединяются в одну историю. Сквозные трассы control-traces-0.21.json отдельно задают переходы, повтор и потерю ответа.

Бюджеты канала от 64 кбит/с

7.32 Расчетный нижний ориентир — доступные приложению 64 кбит/с суммарно по узкому участку в двух направлениях; для полнодуплексного канала дополнительно проверяется каждое направление. Паспорт фиксирует фактическую скорость, RTT, потери, туннели и длительности перерывов. 1024 байта XML занимают 128 мс при 64 кбит/с без обрамления, TCP/IP и защиты. При максимальном заполнении записей TLS и отсутствии padding кадр с 1024 байтами XML и четырехбайтовой длиной занимает три записи и 1094 байта полезной нагрузки TCP: 1028 байт кадра и 3*22 байта расходов TLS. Три записи — минимум для такого кадра; меньшие записи увеличивают их число и суммарные расходы. Например, разбиение прикладных байтов на 1+511+511+5 дает четыре записи и 1116 байт в TCP. record_size_limit ограничивает размер записи сверху, а не их количество. Суммарные расходы и повторы добавляются к времени доставки; 1024 не является размером пакета Ethernet.

7.33 Базовые сетевые сроки: heartbeatIntervalMs=5000, controlLeaseMs=30000, frameAssemblyMs=10000, writeProgressMs=10000. ordinaryResponseMs=500 и cancelAdmissionMs=100 отсчитываются после получения полного допустимого кадра; физические stopBeginMs/ stopConfirmMs не увеличиваются из-за медленного канала. Перед первым байтом session.stop могут находиться уже отправляемые данные

TCP. Очередь управления ограничена прежними слотами, файловые и речевые потоки обслуживаются отдельно с приоритетом управления. Сквозной срок session.stop складывается из ожидания и передачи кадра, допуска, выдачи физической команды и подтверждения off.

7.34 Отсоединение канала само по себе не заканчивает действующий lease. При восстановлении до его истечения продолжается прежнее конечное исполнение, если оно еще допустимо по всем остальным срокам. По истечении 30000 мс применяется сохраненная stop/finish; уже начатая остановка не отменяется возвращением сети. Локальная запись и программа с finish обеспечивают продолжение без канала; живая речь не получает обещания продолжения при исчезновении источника. Отзыв прав действует немедленно и не ожидает lease.

7.35 deadlineUtcMs межузлового поручения ограничивает его доставку через АРМ и не обновляется посредниками. Конечный АРМ вычитает погрешность часов и бюджет последнего обмена, затем выбирает acceptBeforeMs в монотонной шкале УУ не позднее остатка срока и не далее 30000 мс от текущего uptimeMs. Если гарантированного остатка нет, session.start не отправляется. maxDurationMs начинается от принятия session.start и ограничивает подготовку и исполнение, а не срок маршрутизации. hopLimit ограничивает переходы независимо от этих трех бюджетов.

8 Сеанс, физический результат и потеря связи

Предлагаемый автомат сеанса

starting -> running -> stopping -> finished

starting ----------> stopping -> finished

starting / running ----------> finished (только с установленным результатом)

Таблица 8 - Сеанс, физический результат и потеря связи

Ответ или состояние сеанса	**Что подтверждает**
accepted	Команда принята исполнителем. Усилитель еще может не быть включен.
running	Выполнен критерий начала из профиля ОУ. Критерий ДОЛЖЕН быть указан, а не выведен из TCP ACK.
finished	Логическое исполнение завершено с result и reason. Состояние выходов передается отдельно.

Таблица 9 - Сеанс, физический результат и потеря связи

Состояние ресурса	**Наблюдение**
off / commandedOff / onObserved / unknown / fault	Отдельная ось состояния вместе с источником и уровнем доказательства. Отправленная команда выключения сама по себе не доказывает off.

Границы исполнения

8.1 На одном УУ в базовом профиле ДОПУСКАЕТСЯ один сеанс. Параметры запуска фиксируют ресурсы, версию контента и снимок конфигурации. maxDurationMs ограничивает все исполнение от принятия, включая подготовку и паузы. Истечение этого срока инициирует остановку, но не доказывает физического выключения неисправного ОУ. Срок приема session.start и таймаут подготовки задаются отдельно.

8.2 Для начала и окончания передаются отдельные критерии и доказательства: принятие УУ, ответ ОУ, наблюдаемый режим, электрическое или акустическое наблюдение при наличии. Если профиль не да-

ет обратной связи, выход остается commandedOff, а недоказанный физический итог — unknown. В ядре следующий запуск требует подтвержденного off; commandedOff и unknown сохраняют блокировку до восстановления по разделу 8. Успешный тест нельзя обосновать одним отправленным session.stop.

8.3 Естественное окончание конечной программы отличается от исчерпания общего предела и от аварийного завершения. Причина maxDurationExceeded сама по себе не является успехом, даже если выход уже выключен. Перед дополнительным session.stop после автономного завершения проверяются контекст исполнения и правила драйвера: поздняя команда НЕ ДОЛЖНА прерывать другую работу ОУ.

Ошибки и потеря связи

8.4 Потеря АРМ при сирене или встроенной записи

Для purpose=real применяется сохраненная на конкретном УУ настройка onControlLoss: stop или finish; для exercise/technical применяется stop. Она фиксируется при запуске вместе с configRevision.

8.5 stop

Переход к остановке и проверке выходов. Возврат связи не запускает программу заново.

8.6 finish

Доигрывание конечной программы в принятом пределе. Принятая подготовка в starting продолжается в пределах prepareTimeoutMs и maxDurationMs.

8.7 Наблюдатель остается на связи

Его запросы не продлевают жизнь канала владельца.

8.8 Потеря взаимодействия с ОУ при доступном УУ

Отдельная авария источника ОУ. Политика реакции задается профилем; onControlLoss относится к АРМ.

8.9 Перезагрузка УУ

Новый bootId, старые команды не действуют. Автозапуск прошлого сеанса запрещен. ОУ могло остаться включенным: сначала проверка и согласование состояния.

Для живой речи отдельно задаются потеря аудио, потеря управления, завершение потока, сброс буфера и возможная резервная запись. Значение stop/finish нельзя автоматически переносить на живую речь.

Сохранение намерения

8.10 До передачи физического запуска УУ ДОЛЖЕН надежно записать намерение воздействия: версию формата, номер записи, deviceId, bootId, principalId, controlEpoch, cmdSeq, sessionId, нормализованное поручение, точные ресурсы с bindingGeneration, ревизии, onControlLoss и пределы исполнения. После завершения записи ее целостность проверяется чтением. Только затем разрешены accepted и передача задания ОУ. Отказ записи или проверки дает storageFault без воздействия.

8.11 Постоянный журнал использует обнаружение оборванной записи и атомарный признак фиксации: контрольная сумма/длина/поколение, затем commit-маркер по гарантиям носителя. Единственная действующая запись не стирается до надежной фиксации замены. Незавершенная или поврежденная запись при загрузке означает unknown; ее нельзя интерпретировать как отсутствие запуска. Успешная фиксация намерения без физической передачи также не доказывает, что ОУ звучало. Новый bootId не позволяет повторить session.start автоматически.

8.12 Закрытие записи воздействия ДОПУСКАЕТСЯ только после надежного сохранения результата и критерия off из профиля ОУ. Сбой Flash после воздействия сохраняет блокировку следующего session.start; адресная остановка текущего исполнения не ждет записи Flash и ДОПУСКАЕТСЯ по сохраненному в RAM контексту владельца. Если после reboot нельзя доказать происхождение воздействия или привязку ре-

сурса, автоматическая остановка по поврежденному журналу запрещена; используется местная процедура восстановления. Паспорт ДОЛЖЕН задавать емкость, ресурс перезаписи, максимальное время фиксации и поведение при полном журнале.

8.13 Приемочные отключения питания выполняются до и после записи намерения, commit-маркера, ответа accepted, передачи session.start ОУ, приема session.stop, подтверждения off, записи результата и стирания прежнего блока. Для каждой границы проверяются отсутствие повторного session.start, сохранение происхождения воздействия и достижимость предусмотренного восстановления.

Восстановление после питания

8.14 Восстановление имеет состояния reconciling, ready и localInterventionRequired, отдельные от сеанса. После reboot УУ меняет bootId, запрещает session.start и читает журнал незавершенного воздействия и наблюдения ОУ. Если сохраненная привязка совпадает и журнал указывает собственное незавершенное исполнение, заранее настроенная процедура recoveryStop может выдать ограниченную остановку его ресурсов. Эта процедура является восстановлением после сбоя; она не принимает произвольный сетевой session.stop без sessionId.

8.15 Достоверный off

Профиль задает источник, свежесть и необходимое число независимых наблюдений. После выполнения критерия УУ переходит в ready.

8.16 Собственное незавершенное воздействие

Применяется описанная в паспорте recoveryStop: точные ресурсы, допустимые команды, предел попыток и срок подтверждения. Отправленная команда дает commandedOff, а не off.

8.17 Потеря взаимодействия с ОУ без перезагрузки

Сохраняются владелец, сеанс и неизвестный результат. После восстановления взаимодействия с ОУ выполняется та же проверка выключения; новое включение само не возникает.

8.18 Неизвестное происхождение включения

Чужое или местное воздействие не присваивается старому сеансу. Автоматическая остановка без разрешенного правила профиля запрещена.

8.19 ОУ без достаточной обратной связи

Используется независимая обратная связь либо местное восстановление с подтвержденным физическим отключением. Нажатие сетевой кнопки квитирования или истечение таймера не снимает блокировку.

8.20 Истек recoveryTimeoutMs

localInterventionRequired с причиной, ресурсами и необходимым действием. УУ продолжает диагностику; новый достоверный критерий off позволяет завершить восстановление без нового включения.

8.21 Паспорт ОУ ДОЛЖЕН содержать recoveryTimeoutMs, recoveryStop либо запрет автоматического воздействия, критерий off, источник местного подтверждения и порядок снятия localInterventionRequired. Местное подтверждение привязано к текущим bootId и bindingGeneration и поступает через объявленный физический сервисный вход или доверенную сервисную процедуру; удаленный флаг оператора не заменяет это доказательство. Профиль без автоматической обратной связи обязан явно требовать местного вмешательства, а не обещать автоматическое восстановление.

8.22 finish относится только к уже принятому конечному исполнению, включая его starting. Потеря управления не отменяет принятую подготовку, но не продлевает prepareTimeoutMs или maxDurationMs. Следующий самостоятельный сеанс, для которого session.start еще не

принят, не исполняется. Все этапы уже принятого mode=program входят в единое поручение и продолжаются по правилам раздела 26. Перезапуск УУ всегда переводит ресурсы в восстановление, независимо от finish.

Хранение результатов

8.23 Результат через session.get

Не менее resultRetentionMs от перехода в finished в пределах одной непрерывной загрузки УУ. Отключение АРМ, release и новая эпоха в той же загрузке не сокращают срок; права чтения проверяются отдельно.

8.24 Намерение и итог в постоянном журнале

Для восстановления сохраняются происхождение, ресурсы, решение о завершении и свидетельство критерия off. Незакрытое или сомнительное воздействие не удаляется по истечении resultRetentionMs. Полный набор страниц, checks и observations не обязан храниться в энергонезависимой памяти ядра.

8.25 Перезагрузка УУ

Старый bootId в session.get получает staleBoot. Запрос с новым bootId не является способом адресовать прошлую загрузку; неизвестный прежний sessionId дает notFound либо resultExpired по доступной истории. Ни один из этих отказов не доказывает отсутствия воздействия.

8.26 Сведения восстановления

Журнал прежней загрузки используется локальной процедурой recovery. Через device.state нового bootId доступны текущее recovery, состояние ресурсов и аварий. Это не выдача прежнего результата и не перенос его времени в новую шкалу uptimeMs.

8.27 Архив между загрузками

Чтение прежних результатов требует отдельного версионированного профиля постоянного журнала с адресацией исходного bootId,

сроком хранения и правами. log.read профиля 0.21 адресует только журнал текущего bootId. APM сохраняет полученные результаты в своем журнале.

Временная шкала и допустимые результаты

8.28 Временная шкала исполнения является частью результата. Все отметки относятся к uptimeMs одного bootId, имеют тип uint64 в миллисекундах и не зависят от UTC. Отметка отсутствовавшего перехода опускается; число 0 является допустимым временем и не обозначает отсутствие перехода. Переподключение и повтор сообщения не изменяют уже зафиксированные отметки.

8.29 Кардинальность и страницы

Ровно одна запись timeline во всем полном снимке результата. Она может находиться на любой странице; повтор или отсутствие записи нарушает полноту. sessionId ДОЛЖЕН совпадать с session.id первой страницы. Запись не включается в resourceCount/checkCount/ observationCount/executionCount/faultCount. Все страницы result содержат одинаковый evaluatedAtMs - время создания снимка; времена наблюдений и уже состоявшихся переходов не превышают его.

8.30 prepareDeadlineMs

Обязательно. min(acceptedAtMs + prepareTimeoutMs, acceptedAtMs + maxDurationMs). Сложения выполняются без переполнения; непредставимый срок отклоняет session.start до воздействия.

8.31 runningAtMs

Присутствует тогда и только тогда, когда впервые достигнут критерий running. acceptedAtMs <= runningAtMs < prepareDeadlineMs. В program означает начало исполнения плана после общей подготовки; первая pause также начинает план, хотя выходы остаются off.

8.32 runDeadlineMs

Присутствует вместе с runningAtMs. Для сирены stopMode=duration min(runningAtMs + durationMs, acceptedAtMs + maxDurationMs); для stopMode=command acceptedAtMs + maxDurationMs, для записи/потока/теста/программы - acceptedAtMs + maxDurationMs. Естественный конец может наступить раньше.

8.33 stopRequestedAtMs

Время приема первой адресной отмены этого исполнения. Если отмены не было, атрибут отсутствует. Повторная отмена не сдвигает время.

8.34 stoppingAtMs

Время первой причины перехода к stopping: отмена, конец, предельный срок или отказ. Если первой причиной является отмена, равно stopRequestedAtMs; если истечение срока - этому сроку. Более поздняя отмена и повторная причина не сдвигают уже начатую остановку.

8.35 stopIssuedAtMs

Время фактической выдачи первой физической команды выключения после stoppingAtMs. Отсутствует, если команда не требовалась при уже достоверном off или не была выдана вследствие отказа. Не подменяется временем постановки в очередь.

8.36 stopBeginDeadlineMs / stopConfirmDeadlineMs

Оба обязательны при наличии stoppingAtMs. Равны stoppingAtMs + stopBeginMs и stoppingAtMs + stopConfirmMs; сроки фиксируются при переходе.

8.37 finishedAtMs

Обязательно только для state=finished. Не раньше acceptedAtMs и имеющихся runningAtMs/stoppingAtMs/stopIssuedAtMs. Итоговое доказательство off относится к промежутку от stoppingAtMs, либо acceptedAtMs при отсутствии физического запуска, до finishedAtMs включительно.

8.38 При now >= любого действующего предельного срока сначала обрабатывается его истечение. При одновременном естественном конце и максимальном сроке выбирается maxDurationExceeded. Успех запрещен, если running достигнут после срока подготовки, физическая команда остановки выдана после stopBeginDeadlineMs либо off получен после stopConfirmDeadlineMs. Нарушенный бюджет отражается failed с причиной prepareTimeout, stopDispatchTimeout или stopConfirmTimeout; неизвестное выключение дает unknown/stopUnconfirmed. Позднее подтверждение выключения снимает блокировку ресурса через его текущее состояние, но не переписывает завершенный результат в успех.

8.39 Наблюдение теста с validity=valid ДОЛЖНО принадлежать своей фазе: baseline - от acceptedAtMs до runningAtMs включительно; running - от runningAtMs включительно до stoppingAtMs исключительно; stopped - от stoppingAtMs до finishedAtMs включительно. Если верхняя граница фазы еще не наступила, используется время создания снимка. Наблюдение фазы, в которую исполнение не входило, передается с validity=unavailable, freshness=never, observedAtMs=0 и reason=phaseNotEntered; оно не удовлетворяет критерию успеха. Сохраненное старое показание не становится доказательством новой фазы.

8.40 succeeded

completed при полном выполнении либо operatorStop для siren/recording/stream с confirmed всех ресурсов, без нарушения исполнения и с off в срок. Полнота передается отдельно в completion.

8.41 cancelled

operatorStop, когда условия succeeded/operatorStop не выполнены, либо controlLostStop: известное прекращение и off в срок. Недостаток обязательного доказательства дает unknown.

8.42 failed

maxDurationExceeded, prepareTimeout, stopDispatchTimeout, stopConfirmTimeout, sourceUnavailable, driverFault, storageFault,

contentMismatch, mediaIncomplete, mediaClockFault, hardwareProtection или testFailed, equipmentFault: известное нарушение требования, выходы подтверждены off. Конкретная неисправность сохраняется в fault.

8.43 unknown

stopUnconfirmed, observationLost или recoveryUncertain: обязательное доказательство физического исхода отсутствует. Отправка команды и истечение таймера не разрешают другой итог.

8.44 Отсутствие наблюдений аварии требует freshness=never, validity=unavailable, state=unknown, observedAtMs=0 и changedAtMs=0. При наличии наблюдений changedAtMs <= observedAtMs; в снимке устройства observedAtMs <= uptimeMs снимка. Ответ channel.ping без controlEpoch ДОЛЖЕН содержать leaseRemainingMs=0, даже если на УУ существует другой владелец. Наблюдатель не продлевает его lease.

Принятие, подтверждение исполнения и завершение

8.45 Принятие команды (АП), подтверждение исполнения и завершение являются разными фактами. accepted подтверждает проверку и сохранение поручения, но не работу ОУ. Подтверждение исполнения означает, что выбранный ресурс фактически выполнил критерий начала заданного воздействия. Оно передается сразу после получения достаточного доказательства, без ожидания остановки. Обозначение «РП» не является термином или полем этого протокола; АРМ может использовать его для отображения подтверждения исполнения.

8.46 Для каждого выбранного ресурса УУ ведет execution. В siren подтверждается начало заданного сигнала, в recording/stream — начало вывода отсчетов именно принятой записи/потока в исполнительный тракт. Прием файла, пакета RTP, включение усилителя без подачи принятого звука и TCP ACK недостаточны. Допустимые источники и граница физического наблюдения определены executionCriterion аппаратного профиля. reportedMode не объявляется акустическим измерением; от-

сутствие акустического датчика не запрещает явно объявленный цифровой или электрический критерий. УУ без достаточного доказательства сообщает unknown, а не confirmed.

8.47 pending

Критерий еще ожидается в пределах подготовки. Отправленная команда не меняет pending на confirmed. В finished этот статус запрещен.

8.48 confirmed

Критерий фактически выполнен; criterionId, sourceId, evidence и observedAtMs обязательны. Факт, время и доказательство сохраняются неизменными до конца хранения результата, в том числе после остановки или аварии.

8.49 notExecuted

Достоверно установлено, что воздействие данного прохода не начиналось. ДОПУСКАЕТСЯ при отмене до передачи физического запуска либо другом доказанном отсутствии воздействия; один таймаут или отсутствие ответа этого не доказывает.

8.50 unknown

Доказательства начала недостаточно. До завершения последующее достоверное наблюдение может уточнить статус; после finished исторический результат не переписывается.

8.51 У сирены, записи, потока и теста имеется по одной execution на каждый ресурс. У program имеется запись для каждой пары ресурс/проход звучащего этапа; stageIndex и passIndex обязательны. Пауза execution не создает: запуск таймера паузы не подтверждает звучание. Для будущего прохода статус pending, после отмены не начатого прохода — notExecuted. session.started отражает переход running, а session.execution сообщает подтверждение отдельного ресурса. Эти события не взаимозаменяемы. Потеря события восстанавливается через session.get; снимок не вызывает повторного воздействия.

8.52 Нормальный конец

succeeded/completed: полный план завершен, все необходимые execution подтверждены, обязательные проверки пройдены, все выходы off в срок.

8.53 Оператор остановил сирену, запись или поток после подтвержденной работы

succeeded/operatorStop при confirmed всех выбранных ресурсов, отсутствии нарушений исполнения и подтвержденном off в срок. Неполная запись отмечается completion=interrupted; это не ошибка начала воспроизведения.

8.54 Оператор отменил до начала воздействия

cancelled/operatorStop при notExecuted всех ресурсов и подтвержденном off. Положительное подтверждение исполнения отсутствует.

8.55 Сработала лишь часть ресурсов либо остановлена неполная программа/проверка

cancelled/operatorStop при известном исходе и off. Подтверждения сработавших ресурсов сохраняются, неполнота явно видна; общий успех всей программы/проверки не выдается. Неизвестное начало хотя бы одного ожидаемого воздействия дает unknown/observationLost.

8.56 После подтвержденного начала возник отказ

execution=confirmed сохраняется. Итог failed с причиной отказа при известном off; при неизвестном off — unknown/stopUnconfirmed. Отказ сохраняется в fault независимо от последующего session.stop и снятия аварии.

8.57 Потеря управления с политикой stop

cancelled/controlLostStop при известном исходе и off, либо unknown при недостатке доказательств. Это не штатная команда оператора. Политика finish продолжает только принятый конечный план.

8.58 Приоритет терминального итога: неизвестное выключение; установленный отказ исполнения; недостаток обязательного

доказательства; разрешенная отмена; успех по установленным выше условиям. Нормальная остановка не заменяет уже установленный отказ. reason выбирается детерминированно: stopUnconfirmed при неизвестном off; иначе самая ранняя зафиксированная причина отказа, при равном времени — лексикографически по коду; иначе observationLost для недоказанного начала; иначе operatorStop/controlLostStop/completed по первой причине stopping. Все дополнительные отказы остаются в fault. Коды fault, совпадающие с причиной failed из словаря resultReason, сохраняют эту причину; остальные аппаратные коды с effect=failed дают equipmentFault. recoveryUncertain применяется при недоказанном происхождении в процедуре восстановления. Временные нарушения и maxDurationExceeded являются отказами, а не штатным концом.

8.59 АРМ показывает раздельно принятие, подтверждение исполнения каждого ресурса, текущую аварию и завершение с причиной. Подтвержденный запуск не стирается поздней аварией, но текущая строка и общий итог не могут показывать безусловный успех при fault.effect=failed или unknown. Событие effect=none не ухудшает результат; degraded сохраняет подтверждение и видимый признак ухудшения. Конкретные названия колонок и цвета протокол не устанавливает.

9 Повторы, отмена и неизвестный исход

9.1 В профиле обработка изменяющего запроса имеет две границы. Сначала проверяются целое обрамление, XML и его лимиты, общая оболочка, тип cmdSeq, полномочия, bootId, controlEpoch и канал. Отказ на этой границе НЕ ДОЛЖЕН менять highWater. После нее обычный новый cmdSeq выше highWater расходуется до проверки предметных аргументов, поддержки действия, срока и состояния ресурса. Любое решение второй границы, включая invalidArgument, unsupported, unsafeState, busy и startWindowExpired, ДОЛЖНО сохраняться в кеше. Неизвестный action с обычным номером и корректным контекстом изме-

няющего запроса также расходует номер и получает unsupported. Емкость кеша - не менее 8 обычных решений; решение зарезервированного session.stop/session.cancel сохраняется дополнительно до освобождения эпохи.

9.2 Точный повтор в кеше возвращает прежнее решение. Другое нормализованное содержимое запроса при том же номере дает conflict. Старый номер без сохраненного решения и без адресной отмены получает resultExpired; highWater сохраняется. Для повтора сравнивается нормализованное содержимое полей, а не порядок XML-атрибутов. Транспортный id служит корреляции и не превращает тот же cmdSeq в новое действие. Ошибка парсинга до достоверного извлечения контекста не расходует номер.

9.3 Точный повтор сохраненной команды возвращает ее решение без нового действия. Совпадение номера при другом содержимом является конфликтом.

9.4 Старый номер без решения и адресной отмены в кеше получает resultExpired. История не восстанавливается предположением по текущему idle.

9.5 Прежний accepted из кеша не означает, что сеанс сейчас работает. Актуальное состояние читается отдельно.

9.6 При reconnect сохраняются highWater, кеш и активный startSeq. После control.resume прежний канал теряет право изменяющих команд.

9.7 Новый cmdSeq для повторной доставки session.start создает новое действие и поэтому не применяется как повтор.

9.8 Гарантия отсутствия повторного исполнения ограничена контекстом bootId/controlEpoch. Exactly-once через отключение питания не заявляется.

Отмена без полученного sessionId

session.start cmdSeq=1

CANCEL cmdSeq=2, startSeq=1

Доставка 1,2: остановить исполнение 1.

Доставка 2,1: запретить поздний запуск 1.

START cmdSeq=3, затем старый session.cancel target=1: сеанс 3 не затрагивается.

Отмена конкретного запуска

9.9 У cancel собственный новый номер; startSeq строго меньше cmdSeq. Если активного совпадения нет, ответ означает отсутствие активного исполнения с этой ссылкой и запрет позднего session.start. Это не утверждение, что звук никогда не включался. Один дополнительный слот разделяют session.stop и session.cancel; обычные изменения ждут разрешения этой гонки.

9.10 Сохраненное решение исходного session.start имеет приоритет при его повторе. Если такого решения нет, но в ограниченном кеше есть подтвержденная отмена этого startSeq, запоздавший session.start получает startBlockedByCancel со ссылкой cancelSeq. Код означает запрет позднего исполнения и не доказывает историческое отсутствие запуска. После вытеснения этой записи возвращается resultExpired; highWater продолжает блокировать session.start. Для долговременного аудита APM сохраняет решения и происхождение отмены в своем журнале.

Сравнение повторных запросов

9.11 Нормализация повтора выполняется после проверки обрамления, безопасного XML и общей оболочки, до проверки предметного формата. Граница оболочки проверяет v/profile/id/action и типы bootId/controlEpoch/cmdSeq; изменяющий запрос требует полный контекст. Проверка прав, актуальности контекста и highWater выполняется УУ отдельно. Нормализация не выдает разрешения на исполнение. Неизвестный action с достоверным контекстом и неверные предметные поля сохраняются в представлении сравнения: unsupported и invalidArgument после этой границы расходуют номер по общему правилу.

9.12 Из сравнения исключается только транспортный id. Сравниваются имена элементов, все остальные атрибуты и дочерние записи. Атрибуты упорядочиваются по имени; разрешенные XML-ссылки декодируются. Строки сравниваются по Unicode-кодовым точкам без изменения регистра, обрезки пробелов или Unicode-нормализации. Межэлементные пробелы не значимы. Значения по умолчанию не подставляются. Канонические числа и bool проверяются на предметной границе; ошибочные строки 01, -1 или banana до нее остаются различными строками и позволяют воспроизвести прежний отказ.

9.13 resources сравнивается как множество только если содержит непустой набор resource с заданными уникальными id: записи сортируются по id. При дубликатах, отсутствии id или постороннем элементе сохраняется исходный порядок, без потери записей. Остальные последовательности сохраняют порядок. Неверное содержимое не становится допустимым от нормализации. Общая оболочка с неверным cmdSeq, поврежденный XML или превышенный лимит не дают представления принятого повтора и не расходуют номер.

Права, занятость и структурированный отказ

9.14 TLS-аутентификация подтверждает principalId, но сама по себе не дает права управлять ОУ. Права задаются для действия, назначения real/exercise/technical/maintenance и области ресурсов. IP-адрес, знание sessionId или чужого controlEpoch не предоставляют полномочий. Непрошедший TLS участник не получает прикладной XML-ответ. Участник без требуемого права получает status=error и error.code=forbidden с обязательным detail до проверки занятости; чужие владелец, сеанс, ресурс и состояние в отказ не включаются. В явно выбранном plain применяется та же проверка локальных прав; логический principalId не является доказательством подлинности участника.

Таблица 10 - Права, занятость и структурированный отказ

error.code / detail	**Условие и последствие**
forbidden / roleDenied	Нет права действия или назначения. Новая попытка требует изменения разрешений, а не ожидания освобождения.
forbidden / scopeDenied	Нет права на запрошенную область ресурсов. Внешний ответ не перечисляет существующие закрытые ресурсы.
forbidden / ownerMismatch	Попытка изменить, остановить или восстановить чужое исполнение. Знание его идентификатора не дает права; воздействие и таймеры неизменны.
busy / ownershipHeld	Разрешенный control.acquire конфликтует с действующим несовместимым владением. Сведения о другом участнике не раскрываются.
busy / executionActive	Разрешенный новый session.start конфликтует с действующим сеансом, в том числе собственным. Точный повтор исходного session.start возвращает сохраненное решение до этой проверки.

Окончание таблицы 10

error.code / detail	**Условие и последствие**
busy / localControl	Ресурс занят разрешенным местным управлением. Сетевой запуск не становится отложенным заданием.
busy / serviceActive	Несовместимая сервисная операция находится в неделимой стадии; действуют сроки и правила уступки обслуживанием.
busy / capacityExhausted	Исчерпана разрешенная квота снимков, опросов, медиаподготовок либо резерв результатов при превышении заявленной нагрузки.
unsafeState	Недоказанный off, восстановление, аппаратная блокировка или другое небезопасное состояние. Это не обычная занятость; ожидание busy не заменяет восстановление.

9.15 error содержит code; detail обязателен только для forbidden, busy и conflict и берется из соответствующего закрытого словаря. Для startBlockedByCancel обязателен cancelSeq. Для остальных кодов detail отсутствует. Человекочитаемый текст не является машинным кодом. resource может указывать только доступный отправителю ресурс или класс session, ownership, snapshot, poll, media, service, history. При forbidden он отсутствует. Все общие отказы оболочки, прав и контекста применяются ко всем соответствующим командам, даже если карточка перечисляет только предметные ошибки.

9.16 Порядок проверок: корректность обрамления/оболочки; права действия, назначения и области; bootId; принадлежность известной эпо-

хи/исполнения участнику; актуальность эпохи и прикрепления; точный повтор; предметные ограничения; занятость/безопасность. Чужая известная эпоха дает forbidden/ownerMismatch, устаревшая собственная или неизвестная — staleEpoch; проверка прав всегда предшествует раскрытию контекста. При нескольких отказах одной группы: roleDenied, scopeDenied, ownerMismatch; для busy: ownershipHeld, localControl, executionActive, serviceActive, capacityExhausted. Обязательные сроки и сохраненная отмена проверяются согласно общему порядку session.start.

9.17 forbidden и отказы контекста не расходуют acquireSeq/cmdSeq/highWater, не продлевают lease и не создают сеанс. Предметный busy обычной команды расходует ее номер и сохраняется как решение; повтор возвращает тот же отказ даже после освобождения. Новая попытка — новый номер и повторная проверка состояния. Исключения зарезервированных номеров session.stop/session.cancel/control.release сохраняются. busy никогда не ставит session.start в очередь. Очередь намерений real для acquire является только правилом арбитража: отдельные успешные acquire и session.start по-прежнему обязательны.

9.18 Разрешенные session.stop/session.cancel своего исполнения не получают busy из-за того же исполнения, обычного запроса, заполненной очереди телеметрии или занятого сервиса. Приоритетный слот и резерв состояния отмены обеспечиваются до допуска session.start. Отказ остановки не подменяется положительным подтверждением выключения. Чужой session.stop не становится допустимым после таймаута или при purpose=real.

9.19 При остановке АРМ различает намерение оператора, отправку команды, принятие УУ и подтвержденное выключение каждого выбранного ресурса. Нажатие кнопки, завершение местного таймера и успешная запись в ТСР не являются подтверждением off. При приемке отдельно измеряют задержку от действия оператора до off, включая очередь

клиента, восстановление TCP/TLS и чтение результата. stopBeginMs и stopConfirmMs аппаратного профиля отсчитываются по установленной шкале УУ и не заменяют эту полную задержку. Пока выключение не доказано, АРМ сохраняет видимое ожидание либо неизвестный исход.

9.20 Если при восстановлении управления прежний controlEpoch уже истек, staleEpoch запрещает команды в этой эпохе, но не доказывает ни выключение, ни успех прежнего сеанса. При разрешенном чтении АРМ получает полный сохраненный session.get для прежних bootId/sessionId и актуальное device.state. Исторический результат, причина controlLostStop и доказательства off сохраняются как получены. Текущее off без полного результата не превращает неизвестную историю в succeeded. Повтор START, новая выдача владения и запуск другого сеанса не используются для восстановления результата остановки.

9.21 stop-after-lease-expiry

Входные условия: Сирена работает; TCP разорван до нажатия STOP, восстановление TLS длится дольше оставшегося lease. control.resume возвращает staleEpoch. Сохраненный результат доступен; отдельно его утрата при известном текущем off.

Обязательный исход: Первый вариант возвращает полный итог исходного сеанса и подтверждение off, сохраняя controlLostStop и остальные факты. Второй сохраняет неизвестную историю. Ни новый START, ни новый захват владения для чтения не выполняются. Полная задержка STOP включает восстановление связи.

9.22 microphone-pending-preparation

Входные условия: Одно нажатие тангенты во время обновления подготовки UDP. Повторы с отменой до готовности, окончанием сеанса, потерей источника и отсутствием подтверждения речевого режима.

Обязательный исход: Действующее намерение выполняется после prepared и готовности медиаканала ровно одним START. При startup=outputReady захват и звук начинаются после подтверждения ОУ.

Отмененное намерение не запускается позднее; отсутствие подтверждения не дает успешного включения микрофона.

9.23 unsafeState

Проверяемое условие: Недоказанный off, блокировка восстановления или отсутствие обязательного независимого тракта до нового воздействия/обслуживания.

Команды: control.acquire, control.release, session.start, audio.prepare, audio.group.prepare, audio.discard, content.begin, content.commit, config.stage, config.commit, network.stage, firmware.begin, firmware.verify, firmware.commit, firmware.cancel

9.24 invalidSignature

Проверяемое условие: Ключ издателя отсутствует, отозван или не разрешает данный тип и аппаратную ревизию; при проверке образа подпись неверна; при активации ключ не находится в состоянии active.

Команды: firmware.begin, firmware.verify, firmware.commit

9.25 rollbackDenied

Проверяемое условие: Версия образа ниже действующего минимального разрешенного номера версии, повторно проверяемого перед активацией.

Команды: firmware.begin, firmware.verify, firmware.commit

9.26 notFound

Проверяемое условие: Адресованная существующая сущность не найдена после проверки прав; неизвестный session.cancel по startSeq создает запрет и этим правилом не заменяется.

Команды: session.start, session.stop, audio.prepare, audio.group.prepare, audio.state, audio.finish, audio.discard, transfer.hello, firmware.verify, firmware.commit, firmware.cancel, firmware.status, content.commit, content.cancel, content.status, config.commit, config.cancel, network.confirm, network.cancel

9.27 storageFault

Проверяемое условие: Не удалось прочитать или атомарно сохранить необходимое постоянное состояние, журнал намерений, пакет либо образ. Отказ не является успешным исполнением и не подтверждает off.

Команды: control.acquire, control.resume, control.release, session.start, session.stop, session.cancel, config.stage, config.commit, config.cancel, network.stage, network.confirm, network.cancel, firmware.begin, firmware.verify, firmware.commit, firmware.cancel, firmware.status, content.begin, content.commit, content.cancel, content.status, config.get, content.describe

9.28 staleEpoch

Проверяемое условие: Указанная эпоха не является допустимым контекстом команды; освобожденная эпоха проверяется по отдельному правилу повтора control.release.

Команды: channel.ping, transfer.hello

9.29 staleAttachment

Проверяемое условие: Действующее управляющее соединение не совпадает с прикрепленным каналом либо привязкой операции.

Команды: channel.ping, transfer.hello

9.30 Одинаковые полномочия не требуют одинакового набора предметных отказов: чтение, подготовка и фиксация проверяют разные состояния. Таблица общих отказов и предметная карточка действуют совместно; итоговый список каждой команды и машинный wire-contract формируются из одних записей. unsafeState не обозначает обычную занятость, notFound не раскрывает чужую сущность, storageFault не означает noSpace. Сбой сохранения при остановке не отменяет обязательную попытку физического session.stop и не создает доказательства его результата.

Номера, слоты и точка принятия решения

9.31 cmdSeq оболочки имеется только у изменяющих предметных команд. protocol.hello, control.acquire и control.resume используют собственные правила; channel.ping, чтения и transfer.hello не расходуют cmdSeq и не вытесняют решения из его кеша. cmdSeq внутри selection команды command.get является адресом прежнего решения. Наличие controlEpoch в чтении не превращает чтение в изменение.

9.32 На управляющем соединении одновременно допускаются один обычный запрос, один запрос адресной session.stop/session.cancel и один heartbeat. Чтение занимает обычный слот, но не кеш решений. Новый запрос в занятом слоте получает busy/detail=capacityExhausted; точный повтор текущего id присоединяется к ожидающему решению. При одновременной готовности полных кадров порядок обработки: session.stop/session.cancel, heartbeat, обычный запрос. Это порядок допуска, а не право отменить уже состоявшийся физический результат.

9.33 Решение по изменяющему запросу линеаризуется при атомарной фиксации результата допуска вместе с highWater и записью повтора; session.start фиксирует также намерение до воздействия. Проверки прав, контекста, окна и наличия слота предшествуют этой точке. Для отмены одновременно сохраняется адрес startSeq и запрет позднего запуска. Непосредственно перед физическим session.start исполнитель повторно проверяет сохраненную отмену, актуальность поколения прикрепления и срок подготовки. Если session.cancel принят первым, воздействия нет; если session.start принят первым, отмена останавливает именно его, а успешное принятие session.start не становится доказательством звучания.

9.34 Резервные номера session.stop/session.cancel и control.release не проходят окно cmdSeqWindow, не меняют обычный highWater и не занимают его восемь записей. Отказ вне окна, forbidden и busy из-за занятого слота не расходуют номер, не освежают lease и не вытесняют

решения. Отказ по предметным условиям после допуска сохраняется по общему правилу изменяющих команд. Повтор с измененным содержимым получает conflict, без нового воздействия.

9.35 После потери ответа control.release клиент устанавливает новое соединение того же режима, выполняет protocol.hello и повторяет исходный запрос с прежними id, bootId, controlEpoch и cmdSeq. До проверки требования действующего прикрепления УУ проверяет прежнюю идентичность участника и кеш control.release завершенной эпохи. Точный повтор возвращает исходный released; измененный запрос — conflict. Известная завершенная эпоха без подробного решения дает resultExpired, неизвестная или чужая эпоха — staleEpoch либо forbidden с соблюдением неразглашения. Смена загрузки дает staleBoot. Восстановление результата не прикрепляет канал, не выдает владение и не продлевает таймеры. Оно не исполняет новую команду control.release: если исходное освобождение не было принято, ТРЕБУЕТСЯ обычное восстановление своей еще существующей эпохи через resume.

9.36 При tls прежняя идентичность доказывается действующим ключом того же principalId; отозванный участник получает forbidden. При plain сопоставляются только прежнее открытое пространство clientId и местные права: криптографического доказательства личности нет. Открытый повтор никогда не получает результат защищенной эпохи. При утрате кеша отсутствие подробности не считается доказательством неисполнения; перед новым acquire читается актуальное состояние.

10 Профили ОУ: ВАУ и Сирена

10.1 Профиль перечисляет реальный состав оборудования, допустимые режимы, ресурсы, интерфейс к ОУ, критерии начала и выключения, таймауты и источники наблюдений. ВАУ обозначает оборудование для подачи сигналов и речевых сообщений; Сирена - оборудование для

подачи звукового сигнала. Наличие каждой функции и точки диагностики объявляется отдельно и подтверждается испытаниями реализации.

Таблица 11 - Профили ОУ: ВАУ и Сирена

Свойство	**ВАУ**	**Сирена**
Основное питание	Наличие питания и доступные измерения; напряжение и число фаз задаются профилем	Наличие питания и доступные измерения; напряжение и число фаз задаются профилем
Резервное питание	Наличие источника, состояние и доступные измерения объявляются отдельно	Наличие источника, состояние и доступные измерения объявляются отдельно
Общие отказы	Неисправность ОУ и усилителя при наличии раздельных источников	Неисправность ОУ и исполнительного механизма при наличии раздельных источников
Специфические наблюдения	Состояния установленных выходных линий и признаки работы усилителя	Признаки работы сирены, состояния фаз и токов при наличии соответствующих датчиков
Аудио	Поддержанные источники речи, записи, состояние тракта и буфера	Записи и внешние аудиовходы являются отдельными возможностями; тип Сирена сам по себе их не обещает

10.2 Признак отсутствия питания - дискретная авария, а не численное измерение напряжения. Низкое напряжение АКБ не дает процента

заряда. Число позиций в ответе фидеров не равно числу установленных линий: состав берется из проверенной конфигурации. Состояния off/open/overload/on не дают отдельного надежного energized при неисправности.

Драйвер и нижний интерфейс

10.3 Контракт драйвера определяется функционально: описание возможностей, привязка ресурсов, подготовка, запуск, остановка, получение наблюдений, соблюдение сроков и выдача результата. Наличие выделенного программного драйвера и обработчика входящих байтов НЕ ТРЕБУЕТСЯ; имена функций и структуру программного обеспечения выбирает изготовитель. При обмене с ОУ без идентификаторов корреляции операции сериализуются. Позднее подтверждение после таймаута не подтверждает новую команду. Смена bindingGeneration исключает применение ответов и наблюдений прежней привязки к новому действию.

10.4 Дедупликация сетевых команд сама по себе не дает гарантии однократного физического действия. Если управляющее воздействие запуска передано исполнительному тракту, а результат неизвестен, УУ не повторяет воздействие автоматически: сначала выполняются доступная проверка состояния и оценка достаточности доказательства. Общий режим "запись" не подтверждает конкретное содержание. После перезапуска или неоднозначного подтверждения исход остается unknown. Профиль задает правила повторов и проверки выключения; для интерфейса с обменом сообщениями также задаются правила удаления устаревших данных и выдержки перед следующим обменом.

10.5 При наличии двух путей к одному ОУ, например Ethernet и радиоканала через внешний шлюз, АРМ хранит одну логическую цель и связь маршрутов с общим физическим ресурсом. Таймаут первого пути не доказывает отсутствие запуска и не разрешает повторный

session.start по резерву. Профиль резервирования ДОЛЖЕН обеспечить исключение двух исполнителей, сохранение происхождения и последующее согласование. Радиоканал не объявляется встроенной возможностью УУ.

Профиль оборудования определяет состав ОУ, способ управления, источники подтверждения, границу наблюдения и пределы реакции. Параметры линии, адресация и формат кадров задаются только для использующего их интерфейса. Подтверждение по внутреннему интерфейсу само по себе не является доказательством слышимости оповещения населением.

Машинный паспорт физического исполнения

10.6 profiles.get обязательно для каждого объявленного ресурса. Запрос selection resourceId либо resourceId/snapshotId/cursor. Ответ profiles: snapshotId, resourceId, bindingGeneration, capabilityRevision, more/cursor. Первая запись profile содержит id/revision/kind=vau|siren; остальные записи имеют форму parameter name/type/part/parts/value, type=bool|enum|int|counter|string, name до 32 ASCII-символов. Полное значение до 128 байт передается частями по 64 байта по разделу 7; все части каждого имени обязательны. Записи executionCriterion задают подтверждение начала по разделу 19. Пара id/revision однозначно обозначает полный набор параметров. АРМ сохраняет его до session.start; смена любого параметра меняет ревизию и bindingGeneration.

10.7 prepareTimeoutMs / stopBeginMs / stopConfirmMs

Положительные целые миллисекунды; stopBeginMs <= 1000, stopBeginMs <= stopConfirmMs <= 10000. Для технического теста пределы не выше 100 и 3000 мс.

10.8 runningEvidence / offEvidence

reportedMode|electrical|acoustic|localConfirmation и соответствующие runningPointId/offPointId из каталога; observedAtMs и качество обязательны. Команда, отправленная без наблюдения, не является ни одним из этих доказательств.

10.9 evidenceMaxAgeMs / stableSamples

1..1000 мс и 1..8 последовательных независимых наблюдений. Повтор чтения одного sampleSeq не увеличивает stableSamples. Для теста независимые off/control-каналы задаются independentStop=true и independentOffPointId.

10.10 continuousSupported / intermittentSupported

bool. При continuous сигнал не имеет программных пауз. Для intermittent обязательны onMs/offMs:100..60000, firstPhase=on, finalPartial=truncate, toleranceMs=0..100. Частотное/акустическое описание включается параметром signalSpecId и его signalSpecRevision; оно является частью неизменного паспорта, а не выводится из имени pattern.

10.11 durationMinMs / durationMaxMs / durationStepMs

100..3600000, min<=max, step>=1. Запрошенная длительность попадает в диапазон и кратна step от min; иначе unsupported, без округления.

10.12 recoveryStopAttempts / recoveryTimeoutMs

0..3 попыток и 1..10000 мс. При ненулевом числе попыток профиль задает точные собственные ресурсы и независимый критерий off. Нельзя останавливать чужой обнаруженный запуск по одной догадке о происхождении.

10.13 Соответствие сетевому протоколу АРМ–УУ не требует определенного внутреннего интерфейса УУ–ОУ и не означает совместимость с произвольным усилителем или сиреной. Паспорт изделия ДОЛЖЕН описывать состав исполнительного тракта, способ взаимодействия, границу наблюдения, критерии подтверждения начала и состояния off, обработку отказов и конечные сроки. Для встроенного исполнительного тракта профиль указывает integrated и границу наблюдения. Публикация внутреннего программного интерфейса, электрической схемы или соб-

ственного протокола изготовителя условием соответствия сетевому протоколу не является.

10.14 Если изготовитель дополнительно объявляет совместимость сменных УУ и ОУ разных производителей по определенному интерфейсу, он ДОЛЖЕН указать открытый версионированный профиль этого интерфейса. Профиль определяет физическое взаимодействие, управляющие воздействия, наблюдения, ошибки, повторы, отмену, критерии начала и выключения и конечные сроки. Параметры линии, адресация, границы и контроль целостности кадров задаются при использовании обмена сообщениями; для дискретных сигналов задаются уровни, назначение и временные соотношения. Эти требования относятся к дополнительно заявленной совместимости интерфейса УУ–ОУ.

10.15 При объявлении такой совместимости profiles.get передает lowerProtocolId/lowerProtocolRevision и lowerSpecUri/lowerSpecHash обычными parameter. lowerSpecHash содержит хеш Стрибог-256 от точной последовательности байтов описания: 256 бит, 32 байта, 64 строчных hex. Идентификатор и ревизия имеют тип string/counter, URI и hash — string, предел значения 128 байт. URI имеет схему https; описание доступно независимо от изготовителя АРМ. При отсутствии отдельной декларации совместимости интерфейса УУ–ОУ передавать эти параметры НЕ ТРЕБУЕТСЯ; это не ограничивает соответствие сетевому протоколу АРМ–УУ.

Реестр описаний сигналов

10.16 signalProfileId/signalProfileRevision этапа программы ссылаются на id/revision существующего аппаратного профиля в profiles того же пакета конфигурации. kind=vau|siren соответствует виду оконечного устройства; этап сирены ДОПУСКАЕТСЯ для обоих видов при объявленной поддержке выбранного pattern и длительности. Его targetIds ДОЛЖНЫ покрывать все цели данного этапа. Для такого этапа в профи-

ле любого вида обязательны signalSpecId/signalSpecRevision и параметры выбранного вида сигнала. Они ссылаются на id/revision записи signalSpecs пакета open-alert-config/5 и согласованы с ней. Отсутствие ссылки, другая ревизия или несогласованность параметров отклоняют весь пакет с invalidArgument.

10.17 Вид builtIn обозначает штатный сигнал оконечного устройства: команда запуска выбирает его целиком, без задания акустической частоты или формы через протокол. Запись signalSpec для builtIn содержит только id, revision и kind. Изделие назначает id и revision автоматически; ручной ввод этих значений и акустических параметров оператором НЕ ТРЕБУЕТСЯ. При изменении штатного сигнала изменяется его идентификатор либо редакция, а привязанный аппаратный профиль получает новую идентичность или редакцию. builtIn НЕ ДОЛЖЕН интерпретироваться как tone с условной частотой или использоваться для обхода проверки явно заданного частотного сигнала. Допустимые pattern, длительность и критерии исполнения объявляются обычным порядком.

10.18 id / revision

id: token; revision: положительный uint64. Пара обозначает неизменное описание. Коллекция signalSpecs сортируется по id и не содержит повторов; пустая допустима, если нет профилей сирены.

10.19 kind / waveform

kind=builtIn|tone|linearSweep|mechanical. Для builtIn waveform и числовые параметры сигнала отсутствуют. Для остальных видов waveform=sine|square|mechanical; mechanical требует waveform=mechanical, синтезируемые сигналы требуют sine либо square. Изделие объявляет только реализованные сочетания.

10.20 minHz / maxHz / toleranceHz

Канонические целые: 1 <= minHz <= maxHz <= 20000; 0 <= toleranceHz < minHz. Для tone minHz=maxHz. Для mechanical границы

обозначают допустимую основную частоту в установившемся режиме; форма и гармоники не нормируются этим видом.

10.21 periodMs

Для linearSweep: 100..60000 мс, minHz < maxHz. Первая половина периода линейно повышает частоту от minHz до maxHz, вторая понижает обратно; первая фаза начинается с minHz. Для tone/mechanical равно 0.

10.22 profiles.get передает описание выбранного сигнала обычными parameter. Для всех видов обязательны signalSpecId типа string, signalSpecRevision типа counter и signalKind типа enum. Для builtIn параметров signalWaveform, signalMinHz, signalMaxHz, signalToleranceHz и signalPeriodMs НЕТ; добавление этих параметров в связанный профиль отклоняется с invalidArgument. Для tone, linearSweep и mechanical они обязательны: signalWaveform имеет тип enum, остальные counter. Значения совпадают с записью signalSpecs. Для intermittent onMs/offMs задают внешнюю огибающую; каждый on запускает штатный сигнал либо первую фазу заданного частотного сигнала. Эти поля описывают задание ОУ, но не заявляют измерение звукового давления. Критерии физического подтверждения остаются в executionCriterion. Иная параметризованная форма сигнала требует согласованного расширения; имя изготовителя или pattern не подменяет описание.

11 Датчики и каталог ресурсов

11.1 Каталог имеет уровни controller, ethernet, ouInterface и ou/{ouId}, а также линии, входы, выходы и точки наблюдения. Возможности контроллера и подключенного ОУ разделены. Новый производитель добавляется драйвером и профилем без переопределения смысла базовых команд.

11.2 Таблица ниже описывает логическую модель, а не один XML-элемент. inventory.get передает постоянное описание точки, sensors.read

- изменяемое показание. Связь: pointId, capabilityRevision и bindingGeneration в контексте снимка. Источник sourceKind отвечает на вопрос, откуда получено значение; evidence описывает уровень доказательства физического результата. Эти поля не взаимозаменяемы.

Таблица 12 - Датчики и каталог ресурсов

Группа полей точки	**Содержание**
Идентичность	resourceId, parentId, pointId, sourceId; capabilityRevision и bindingGeneration
Смысл	kind, тип bool/enum/int/counter, единица и масштаб; список enum; пороги и применимость
Источник	sourceKind: localInput, ouReply, controllerMetric или derived
Наличие	presence: present, absent, expectedMissing или unknown; поддержка отделена от физического наличия
Способ чтения	readMethod: cached, refreshable или checkOnly; допустимый интервал и время устаревания
Значение	value плюс validity: valid/invalid/unavailable; freshness: fresh/stale/never и причина
Время и последовательность	sampleSeq, observedAtMs, changedAtMs, evaluatedAtMs, maxAgeMs и bootId; качество абсолютных часов отдельно

Каталог и достоверность показаний

11.3 В каталоге описание точки разделяется на самостоятельные записи point с id и facet=identity|meaning|source|acquisition. identity содержит resourceId/parentId/sourceId; meaning - kind, тип, единицу и

масштаб; source - sourceKind/presence; acquisition - readMethod/ maxAgeMs. Дополнительные enum и threshold передаются отдельными записями с pointId в том же снимке; точные поля и проверка полного каталога заданы в разделе 19. Каждая запись ДОЛЖНА укладываться в 12 атрибутов, а целая страница с оболочкой - в 1024 байта. Продолжение переносит целые записи, а не обрывки XML; конец каталога устанавливается только последней страницей снимка. Значения не дублируют весь каталог.

11.4 Чтение кеша не обновляет observedAtMs. Если ОУ не передает время измерения, observedAtMs обозначает время получения ответа УУ. После потери источника наблюдений ОУ старое аварийное значение сохраняется с признаком stale/sourceLost; отсутствие ответа не очищает аварию. Собственные метрики контроллера при этом могут оставаться свежими.

11.5 Температура, ток, напряжение, сопротивление линии, акустический контроль и внешние пожарные входы могут описываться общей моделью. Они объявляются в capabilities только при подтвержденном аппаратном источнике. Каждая объявленная точка ДОЛЖНА иметь определенный источник и правила качества данных.

11.6 Линия, фаза питания и вход ЧС имеют разные kind и собственные pointId. Несколько входов не сворачиваются в один bool. Профиль сохраняет исходный код и версию декодирования наряду с нормализованным значением. Обрыв, перегрузка/КЗ, выключение, неизвестность и отсутствие точки различаются; общий бит перегрузки/КЗ не позволяет выдумать отдельный диагноз. Зарезервированные биты не создают установленные линии. Варианты ОУ с одним общим ответом и с детализацией по фазам имеют разные каталоги.

12 Опросы, проверки и аварии

12.1 Предусмотрены два вида опроса: без запуска ОУ и с запуском ОУ. Опрос с запуском - самостоятельный сценарий проверки работоспособности под воздействием, с включением оборудования и сбором результатов в рабочем состоянии. Чтение кеша датчиков не заменяет ни один из этих сценариев.

12.2 sensors.read

Чтение доступного снимка без аппаратного воздействия. Значение, возраст и качество передаются вместе.

12.3 Опрос без запуска: poll.start / get / cancel

Свежий опрос источников без включения ОУ. operationId связывает результаты; состав ожидаемых точек фиксируется на старте. Отмена относится только к своему заданию опроса.

12.4 diagnostics.read

Инженерные показатели программного и аппаратного состояния. Не выполняет скрытую перезагрузку или тест выходов.

12.5 Опрос с запуском: session.start mode=test

Профиль testId с kind=activationPoll. Запускает ОУ, собирает предусмотренные наблюдения, останавливает и проверяет выключение. Использует общий исполнитель сеансов, права владельца, sessionId и startSeq.

Таблица 13 - Опросы, проверки и аварии

Этап опроса с запуском	**Что фиксируется**
Подготовка / starting	Точные ОУ и выходы, testId/testRevision, допустимые эффекты, ожидаемые точки, исходные показания и состояние ресурсов. Проверяются права и отсутствие конфликта.
Запуск / starting	Команда включения ОУ и подтверждение по профилю. accepted означает прием задания, а не состоявшееся включение.
Наблюдение / running	Показания при запущенном оборудовании и результаты проверок: например, доступные признаки работы сирены, усилителя, линий или питания. Только реально поддержанные источники.
Остановка / stopping	Штатное завершение конечной программы либо отмена/таймаут. Выполняется остановка именно ресурсов этого сеанса.
Проверка выключения / stopping -> finished	Отдельный результат возврата в допустимое состояние. Неизвестное выключение не считается успехом и сохраняет блокировку ресурса.

12.6 tests.list задает неизменный каталог планов тестов с типизированными критериями и наблюдениями фаз. Поддержка mode=test требует выполнения формата каталога настоящего раздела и фиксации testId/testRevision при session.start.

Запуск и оценка проверки

12.7 Для запуска нужны controlEpoch, cmdSeq и права управления выбранными ресурсами. Повтор session.start не включает ОУ повторно. Если sessionId известен, отмена выполняется session.stop; если ответ запуска потерян - session.cancel по startSeq. Команда poll.cancel относится только к опросу без запуска и не является неявной остановкой тестового сеанса.

12.8 Результат опроса с запуском читается через session.get и связан с session.finished: исходные и рабочие наблюдения, checks[] по каждому ресурсу с passed/failed/inconclusive/skipped, полнота набора и отдельное подтверждение выключения. У каждого наблюдения сохраняются фаза, источник, время и качество. Полный сбор данных может выявить неисправность; успешная остановка не превращает провалленную проверку в passed.

12.9 Функциональная часть получает passed только при выполнении всех обязательных проверок; отсутствие нужной обратной связи дает inconclusive. Общий успех требует также подтвержденного завершения воздействия. Истечение таймера в АРМ не доказывает прекращения теста: УУ ограничивает сеанс своим maxDurationMs и отдельно сообщает физический результат остановки.

12.10 Опрос с запуском не перехватывает действующее оповещение. При занятости возвращается busy; отложенный неожиданный запуск после освобождения не создается. При purpose=technical потеря управления всегда применяет stop. Для программы purpose=real сохраняются настроенная stop/finish и конечный предел; exercise всегда применяет stop и предел 1080000 мс с условиями допуска учений. Восстановление сети или взаимодействия с ОУ не запускает проверку заново. Другие воздействия требуют явной политики профиля.

12.11 Состояние опроса queued/running/finished/cancelled отделено от complete - покрытия ожидаемых точек. Каждая точка использует

validity/freshness/reason каталога; недоступная точка не исчезает из результата. При смене каталога результаты не смешиваются с новым набором точек.

12.12 Описание теста содержит эффекты none/readBus/ switchOutput/audible/pauseAudio/resetInterface/writeStorage. Слово "диагностика" не делает звучащую проверку безопасным чтением. Фоновый опрос без запуска не превращается автоматически в опрос с запуском. Плановый опрос с запуском задается отдельно: полномочный инициатор, профиль, ресурсы, разрешенное окно и срок приема. Пропущенное окно не создает позднего автоматического включения. Периоды опроса ограничиваются возможностями взаимодействия с ОУ и бюджетом нагрузки; session.stop имеет приоритет перед новой сервисной работой.

Аварии и переход к оповещению

12.13 conditionId обозначает условие аварии, alarmId - конкретный эпизод. Подтверждение оператором в АРМ не устраняет неисправность и не сбрасывает аппаратную защелку. Ее сброс, если доступен, требует отдельного разрешенного действия и подтвержденного результата.

12.14 Паспорт развертывания ДОЛЖЕН задавать alarmTransitionMaxMs: максимальное время от возникновения санкционированной реальной тревоги на АРМ до принятия УУ нового session.start после технической проверки. Это предел принятия команды, а не начала звучания. В него входят оставшееся исполнение теста, выключение, сохранение результата, освобождение эпохи, ожидание повторной попытки acquire, доставка и обработка acquire и session.start. Учитываются также незавершенный запрос acquire и получение его busy, если они задерживают следующую попытку. Допуск проверки требует соблюдения этого бюджета и при предусмотренной потере ее АРМ. При отсутствии исправного независимого пути выключения и

наблюдения off техническая проверка запрещена. Адресная отмена допускается только уполномоченным владельцем; самостоятельная тревога не получает права прерывать чужое исполнение.

12.15 Переход от технической проверки к реальной тревоге выполняет текущий владелец: адресный session.stop/session.cancel проверки, подтвержденный off и автоматическое освобождение технической эпохи, затем новый acquire purpose=real и session.start с purpose=real. Каждый шаг имеет самостоятельный результат; при unknown второй session.start запрещен. Другой АРМ получает busy и обращается к текущему исполняющему узлу по разрешенной ветви, а не повышает приоритет в обход владения. Назначение real само по себе не дает прав прерывания.

12.16 Для mode=test назначение ДОЛЖНО быть technical, maxDurationMs ДОЛЖНО быть не более 30000. Паспорт теста задает также stopBeginMs/stopConfirmMs. Планировщик допускает проверку только в разрешенное окно и при допустимом бюджете перехода к тревоге. После off и сохранения результата УУ автоматически освобождает техническую эпоху. Продолжение любого плана требует нового acquire; heartbeat не сохраняет прежнее владение.

Доступность тревоги после технического теста

12.17 Технический сеанс выполняется с onControlLoss=stop независимо от настройки обычного оповещения. Его maxDurationMs не превышает 30000 мс, stopBeginMs не превышает 100 мс, stopConfirmMs не превышает 3000 мс. После его завершения, подтверждения off и сохранения результата УУ автоматически освобождает эпоху; heartbeat не удерживает завершенный технический сеанс. Допуск теста требует исправного независимого пути подтверждения/обеспечения выключения и доступного места постоянного журнала.

12.18 Для каждого ресурса, допускающего звучащую техническую проверку, аппаратный профиль ДОЛЖЕН задавать независимый от те-

стируемого канала способ выключения и наблюдения off: управляемое отключение выхода с обратной связью либо отдельный канал управления и подтверждения. Потеря подтверждения по основному интерфейсу ОУ НЕ ДОЛЖНА оставлять исправный резервный путь неиспользованным. Процедура восстановления ограничена stopConfirmMs. При неисправности независимого пути звучащая проверка запрещена, а диагностика без воздействия остается доступна.

12.19 При принятом тесте и отсутствии более раннего действующего или ожидающего real-запуска alarmTransitionMaxMs первого живого участника очереди не превышает 35000 мс. Сумма включает до 30000 мс оставшегося исполнения, до 3000 мс остановки и до 2000 мс последующего перехода: сохранение результата, освобождение эпохи, ожидание следующей попытки acquire, обмен acquire и принятие session.start. На сохранение результата и освобождение отводится не более 500 мс; оставшиеся 1500 мс являются общим бюджетом ожидания, сети и обработки обеих команд, а не только сетевой доставки. Если в момент освобождения еще выполняется прежний acquire, в этот бюджет входит получение busy и интервал до повторной попытки 500..2000 мс. АРМ тревоги ДОЛЖЕН заранее установить управляющий канал выбранного режима tls либо plain. Выбранные пределы всех составляющих ДОЛЖНЫ укладываться в alarmTransitionMaxMs; разрешенный интервал повтора 2000 мс сам по себе не дает гарантии перехода за 35000 мс при максимальной длительности теста. Если сумма превышает предел, maxDurationMs теста уменьшается до допуска либо тест не выполняется. Потеря владельца применяет stop по истечении lease и не отменяет общего предела. Установление нового соединения после возникновения тревоги в эту гарантию не входит.

12.20 Отказ хранилища после воздействия не превращает неизвестный результат в свободный ресурс. Для системы с требованием оповещения при единичном отказе проект развертывания ДОЛЖЕН

предусматривать независимый резервный исполнитель и тракт доставки сообщения без общего отказавшего хранилища, контроллера и ключа активации. Переключение маршрута относится к тому же runId и не дает права прерывать чужое исполнение. Требуемая отказоустойчивость задается составом независимых трактов и предельным временем перехода; таймаут ожидания не считается восстановлением.

Каталог тестов и задания опроса

12.21 tests.list обязательно при mode=test. Запрос содержит selection с необязательным testId либо snapshotId/cursor. Ответ tests содержит snapshotId, capabilityRevision, more, cursor при more=true. Запись test: id, revision, resourceId, kind=activationPoll, maxDurationMs, stopBeginMs, stopConfirmMs, effects — непустой список имен через запятую, в лексикографическом порядке, без пробелов и повторов, из testEffect. Отдельные записи check: testId, id, resourceId, phase=baseline|running|stopped, pointId, operator=eq|ne|lt|le|gt|ge, expected (значение типа точки). Запись observation: testId, id, resourceId, pointId, phase, type, sourceId и sourceKind. Все ID локального типа имеют 1..32 ASCII-символа; revision и сроки - uint64. Для каждого ресурса обязательны наблюдения трех фаз и минимум один критерий. none ДОПУСКАЕТСЯ только единственным эффектом; audible требует switchOutput, resetInterface требует readBus. Указываются все эффекты теста. pauseAudio/resetInterface/writeStorage требуют maintenance и не допускаются в session.start mode=test; для них нужен отдельный явно объявленный сервисный профиль. Полный каталог фиксируется до session.start; смена ревизии дает conflict до воздействия.

12.22 Проверка comparison выполняется над типизированным значением: eq/ne для bool/enum/int/counter, остальные операторы только для int/counter; единица и scale берутся из каталога точки. Наблюдение удовлетворяет критерию только при validity=valid, freshness=fresh и при-

надлежности нужной фазе по timeline. Отсутствие фазы или показания дает skipped с причиной; ложное сравнение - failed/testFailed. Совокупный succeeded требует passed всех обязательных критериев и off всех ресурсов.

12.23 poll.start не включает выходы и не использует владение оповещением. Запрос poll содержит clientOperationId (32 hex), deadlineMs (uptime УУ, не дальше 30000 мс), refresh=cache|readBus и 1..8 point id. clientOperationId дедуплицируется отдельно в principalId/bootId до deadlineMs+resultRetentionMs; другой набор полей дает conflict. Ответ accepted с operation id/clientOperationId/state=queued. ДОПУСКАЕТСЯ одно задание на участника и не более двух на УУ; превышение дает busy. Получение наблюдений от ОУ выполняется с более низким приоритетом, чем управление, и не продлевает deadlineMs. Неизвестная точка дает invalidArgument до создания задания.

12.24 poll.get: selection operationId либо operationId/snapshotId/cursor. Ответ poll с snapshotId, operationId, state=queued|running|finished|cancelled, expectedCount, complete, evaluatedAtMs, more/cursor; записи point используют формат sensors.read. При finished или cancelled полный снимок содержит каждую ожидаемую точку; неполученная точка имеет validity=unavailable, freshness=never, observedAtMs=0 и reason=deadlineExceeded|cancelled|sourceUnavailable. complete=true означает покрытие выбранного множества, а не исправность. poll.cancel: poll operationId/clientOperationId, аутентифицированный создатель задания; ответ cancelled operationId. Отмена идемпотентна, не отменяет session и не удаляет уже полученные наблюдения. Запросы опроса не продлевают lease.

13 Диагностика, события и снимки

13.1 Базовая доставка событий в проектном профиле - pull: APM вызывает events.read с bootId и afterSeq последнего обработанного со-

бытия. Ответ содержит ограниченную страницу записей event и nextSeq; more=true означает, что на момент снимка страницы оставались записи. more=false завершает текущую выборку, но не подписывает клиента на будущие события. Без явно включенной подписки инициативных сообщений нет. Возможность eventsPush=1 добавляет notification и notificationAck по правилам настоящего раздела. events.read занимает обычный слот, session.stop/session.cancel - отдельный; подтверждения уведомлений не занимают эти слоты.

13.2 УУ хранит ограниченный кольцевой журнал, общий для читателей, и не накапливает отдельную неограниченную очередь для медленного АРМ. Если afterSeq предшествует сохраненной истории, ответ сообщает gap=true и oldestSeq. АРМ ДОЛЖЕН получить новый согласованный снимок с границей eventSeq, затем читать журнал после этой границы. Повторный gap обрабатывается по ограниченному алгоритму восстановления ниже. Курсор АРМ сдвигается только после обработки страницы; повторная доставка не создает повторное событие. Переполнение транспортной очереди обрабатывается по разделу 7.

Таблица 14 - Диагностика, события и снимки

Слой	**Минимальное содержание**
Контроллер	bootId, причина запуска, uptime, длительность цикла, запас стека, таймеры, Flash, очереди и потери журнала. Причина сбоя неизвестна, если нет достоверного сохраненного следа.
Ethernet	Link, восстановления, назначение и состояние каждого сокета, таймауты. Закрытый резервный сокет не является неисправностью.
Управление	Владелец, состояние канала, возраст последнего подтвержденного обмена, highWater, сеанс и причины завершения.
Взаимодействие с ОУ	Профиль и поколение, отказы, сроки и возраст последних наблюдений; при обмене сообщениями также ошибки кадров и поздние ответы.
Медиа / выходы	Поток, буфер, потери, underrun, фактические наблюдения ресурсов и неизвестные исходы.
Счетчики	Область boot/stream/session/persistent, поколение, правила насыщения и сброса. Значения разных поколений не вычитаются.

13.3 События session.started/finished/execution/fault, sensor.qualityChanged, resource.changed, inventory.changed и alarm.raised/cleared/qualityChanged получают eventSeq и stateRevision. poll.finished относится к отдельному расширению. В примерах атрибут seq записи event кодирует eventSeq. Потеря части кольцевого журнала явно обозначается gap. Клиент получает актуальный снимок и продолжает с новым курсором. Снимок восстанавливает текущее состояние, но не утраченную историю.

Полнота снимков

13.4 Многостраничное чтение снимков использует snapshotId, bootId, ревизии соответствующего контейнера и ограниченный cursor. Истечение снимка возвращает snapshotExpired; клиент начинает заново. Страницы разных снимков не объединяются. events.read читает изменяющийся журнал через afterSeq/nextSeq без snapshotId/cursor и без snapshotExpired; повтор может вернуть другой доступный диапазон или gap. В базовом варианте поток событий не фильтруется, чтобы не скрывать разрывы последовательности.

13.5 Набор ожидаемых точек фиксируется ревизией каталога и областью запроса. Каждая страница связана с запросом и снимком, имеет явное продолжение или конец. Повторы не увеличивают полноту, перестановка не меняет нумерацию точек, конец по таймауту не заменяет отсутствующие страницы. Снимок согласован относительно модели, но observedAtMs отдельных измерений могут различаться.

13.6 При подключении АРМ получает снимок с границей eventSeq, затем запрашивает события после этой границы через events.read. Изменения сохраняются журналом в пределах его емкости; утрата обнаруживается через gap. Уже активные аварии входят в снимок независимо от появления нового фронта. Для состояния учитываются bootId и ревизия: запоздавшее событие не перезаписывает более новое значение. Восстановление истории и начальный снимок не являются новым внешним запуском.

13.7 Для каждого полного каталога и снимка паспорт задает максимальное число страниц P и условия канала. ДОЛЖНО выполняться P * (RTTmax + responseMaxMs + transferMaxMs) + retryBudgetMs <= snapshotReadMaxMs <= 27000 мс. Слагаемые относятся к одному последовательному обмену; retryBudgetMs включает ограниченные повторы для заявленных потерь, transferMaxMs - сериализацию обоих направлений. Оставшиеся 3000 мс являются запасом до истечения снимка. При

P=100 и RTT=500 мс одни сетевые обмены занимают около 50000 мс: такой каталог на таком канале не соответствует ядру. Изготовитель обязан ограничить объявляемый объем или предложить другой профиль; автоматическое перечитывание не устраняет невозможность уложиться в срок.

13.8 Поддержка многотысячного парка подтверждается паспортом системы: число УУ и наблюдателей, частота events.read, число удерживаемых владений, пиковые события, общий перерыв сети и скорость восстановления. При N владельческих соединений ping с периодом heartbeatIntervalMs=5000 дает N/5 запросов в секунду в среднем, а опрос событий раз в T секунд - N/T; ответы и прочие чтения учитываются отдельно. АРМ ДОЛЖЕН ограничивать пулы соединений, применять квоты и разброс reconnect. Приемка ДОЛЖНА включать одновременное восстановление при session.start/session.stop на заявленной нагрузке.

13.9 Базовое право events.read, также необходимое для events.subscribe, дает доступ ко всему журналу одного УУ; частичное право чтения датчиков не дает этого права. Таким образом базовый курсор не скрывает запрещенные события под видом gap. Изменение прав прерывает чтение и требует новой проверки доступа. Фильтрованный журнал является отдельным профилем с собственным пространством курсоров.

13.10 Паспорт телеметрии содержит eventCapacity, eventRateMax, eventBurstMax, snapshotReadMaxMs и eventDrainMinPerSec. Для заявленной нагрузки ДОЛЖНО выполняться eventCapacity > eventBurstMax + eventRateMax * snapshotReadMaxMs / 1000 и eventDrainMinPerSec > eventRateMax, с запасом на плановые перерывы чтения. Частота учитывается после разрешенной обработки дребезга; число наблюдателей входит в расчет общей пропускной способности. Пределы являются условиями испытаний, а не следствием размера счетчика.

13.11 После трех последовательных gap или snapshotExpired при восстановлении АРМ прекращает автоматический полный перечитывающий цикл, помечает историю incomplete и переходит к периодическому device.state с сохранением управления и текущих аварий. Повтор восстановления выполняется с ограниченной задержкой 5, 10, 20, затем 60 секунд, со случайным разбросом до 20 процентов; переход обратно в синхронное состояние требует успешного снимка и чтения его продолжения без gap. Потерянные исторические переходы остаются потерянными, даже если текущий снимок получен.

13.12 Дребезг подавляется у физического источника по явно настроенному правилу до создания событий. Утвержденные переходы alarm.raised/cleared, изменение качества и результаты сеансов НЕ ДОЛЖНЫ объединяться с потерей перехода. Частые численные измерения могут агрегироваться только в объявленный интервал с min/max/last и счетчиком; переполнение журнала всегда видно как gap. nextSeq равен последнему переданному seq, а для пустого ответа без gap - исходному afterSeq.

Инициативные уведомления УУ и подтверждение доставки

13.13 Возможность eventsPush=1 добавляет инициативные уведомления УУ по действующему управляющему TCP-соединению. В режиме tls они передаются внутри уже установленного TLS, в явно выбранном plain - в открытом соединении с прежними ограничениями доверия. Отдельный порт, дополнительный TCP-сокет и новое рукопожатие для события НЕ ТРЕБУЮТСЯ. УУ самостоятельно обнаруживает изменение, формирует запись журнала и передает notification без запроса очередного события со стороны АРМ. Это уведомление о наблюдении, а не ответ на команду и не разрешение включения ОУ.

13.14 УУ объявляет limit name=eventsPush value=1 и обе команды events.subscribe/events.unsubscribe. АРМ ДОЛЖЕН явно включить до-

ставку через events.subscribe после protocol.hello и проверки возможностей. До успешной подписки УУ НЕ ДОЛЖНО передавать notification. При отсутствии возможности остается events.read. Подписка не захватывает владение, не изменяет состояние ОУ и не продлевает lease. Участник ДОЛЖЕН иметь право чтения полного журнала; права проверяются при подписке и перед каждой новой или повторной отправкой. Отзыв прав закрывает подписку и прекращает выдачу ее данных.

13.15 На одном соединении разрешена одна подписка. Запрос events.subscribe содержит bootId в оболочке и subscription subscriptionSeq/afterSeq. subscriptionSeq - возрастающий номер намерения подписки на данном соединении, 1..18446744073709551615; afterSeq - номер последнего обработанного события того же bootId. Ответ ok содержит subscription id/subscriptionSeq/afterSeq. УУ создает новый 32-значный строчный hex id и закрепляет подписку за данным соединением, режимом защиты и участником. Ответ полностью передается раньше первого notification. Новый успешный events.subscribe с большим subscriptionSeq атомарно заменяет прежнюю подписку. Повтор последнего принятого subscriptionSeq с теми же id запроса и afterSeq возвращает прежний ответ без замены, пока подписка действует. Меньший subscriptionSeq либо повтор прекращенной подписки дает notFound и не возобновляет доставку; тот же номер с измененными id/afterSeq дает invalidArgument. УУ хранит только наибольший принятый номер и контекст последней подписки на соединении. Новое намерение использует больший subscriptionSeq и новый id запроса; после переподключения нумерация начинается с 1. Отказ подписки не меняет прежнюю подписку и наибольший принятый номер.

13.16 notification, УУ -> АРМ

Обязательные поля: v=1, profile, bootId, subscriptionId

Содержимое и назначение: Ровно один events с nextSeq, more, gap, oldestSeq. Записи event совпадают с журналом events.read. Полей id, to, status, action и controlEpoch в оболочке нет.

13.17 notificationAck, АРМ -> УУ

Обязательные поля: v=1, profile, bootId, subscriptionId, nextSeq

Содержимое и назначение: Пустой элемент. Подтверждает проверку и обработку обычной страницы до nextSeq включительно. Ответ на него не передается. Это отдельный тип кадра, без id, cmdSeq и controlEpoch.

13.18 events.unsubscribe, АРМ -> УУ

Обязательные поля: Обычная request с bootId; subscription id

Содержимое и назначение: Ответ ok с subscription id. Неизвестный или уже удаленный id также дает ok и не затрагивает иную подписку. После ответа уведомлений этой подписки нет.

13.19 notification и notificationAck используют прежнюю рамку uint32be длина + XML UTF-8 и прежние пределы: 1024 байта XML, глубина 4, до 32 элементов и 12 атрибутов на элемент. profile равен open-alert-draft/0.21. notification не сопоставляется с ожидающим request по to и не освобождает его слот. notificationAck также не занимает обычный, heartbeat или STOP-слот команд. Начатый кадр передается целиком: смешивать его байты с ответом или другим уведомлением ЗАПРЕЩАЕТСЯ.

13.20 Обычная страница содержит хотя бы одно событие, gap=false и непрерывные seq от afterSeq+1 до nextSeq включительно. afterSeq на стороне УУ означает подтвержденный курсор подписки. more описывает наличие следующей записи на момент формирования страницы; more=false НЕ прекращает подписку. При отсутствии новых событий пустые уведомления не отправляются. События не объединяются с потерей утвержденных переходов; многозаписная страница допускается только в пределах кадра. Порядок и полезная нагрузка совпадают с жур-

налом, включая сработку, возврат, качество, изменение ресурса и результаты сеансов. При недостаточной детализации события АРМ читает необходимые актуальные точки отдельно.

13.21 УУ хранит не более одного неподтвержденного кадра на подписку. Пока он ожидает notificationAck, новые записи остаются в общем ограниченном журнале; отдельная растущая очередь копий ЗАПРЕЩАЕТСЯ. АРМ проверяет bootId, subscriptionId, непрерывность и содержимое, обрабатывает страницу и только затем передает notificationAck. Для новой страницы nextSeq ДОЛЖЕН совпадать с последней записью. УУ продвигает курсор только при точном совпадении nextSeq с ожидающим обычным кадром. Дубликат уже обработанного уведомления не повторяет изменение индикации, журналирование или автоматический сценарий; АРМ повторяет подтверждение. TCP ACK сам по себе не является подтверждением прикладной обработки.

13.22 Если подтверждение не получено за 5000 мс после полной отправки кадра, УУ повторяет тот же XML без изменения nextSeq, oldestSeq, more и полезной нагрузки. До подтверждения сохраняется один и тот же кадр; повтор выполняется не чаще одного раза в 5000 мс и уступает управлению. Дубликаты notificationAck с nextSeq не выше уже подтвержденного курсора игнорируются. Подтверждение неизвестной или прежней подписки, другого bootId либо номера впереди отправленной страницы не продвигает курсор и не освобождает буфер. Переподключение уничтожает подписку; новый идентификатор создается только новым запросом.

13.23 Если очередной afterSeq вне доступной истории, УУ передает notification с events gap=true, more=false, nextSeq=afterSeq, oldestSeq и без записей. Подписка приостанавливает выдачу обычных страниц; этот ограниченный кадр повторяется с тем же интервалом до замены или отмены подписки. notificationAck не снимает gap. АРМ получает новый согласованный снимок с eventSeq и заменяет подписку через

events.subscribe с этой границей. Потерянные переходы отмечаются как неизвестная история. Снимок, повторная доставка и восстановление после разрыва не создают вымышленного нового физического фронта для автоматического запуска.

13.24 events.unsubscribe прекращает подписку только на соединении, где она создана. Уже начатое уведомление завершается до ответа на отмену; еще не начатое удаляется. После успешной замены подписки старые подтверждения не действуют на новую. Обычный events.read остается независимым чтением журнала: он не подтверждает notification, не меняет курсор подписки и не отменяет ее. При необходимости восстановления АРМ явно выбирает курсор нового events.subscribe. Смена bootId всегда требует согласованной начальной синхронизации.

13.25 Выбор следующего исходящего кадра ДОЛЖЕН отдавать приоритет ответам session.stop/session.cancel, затем heartbeat и другим ответам команд, затем notification. Приоритет применяется между кадрами; уже переданные байты не отзываются. Обслуживание входящих STOP, аппаратной защиты, речевого потока и сроков исполнения не зависит от получения notificationAck. Неисправный или медленный наблюдатель не расходует квоты снимков и слоты команд других клиентов. Действуют прежние пределы записи и сборки кадра; бесконечное накопление данных в TCP/TLS-буфере запрещено.

13.26 При успешной подписке АРМ НЕ ДОЛЖЕН продолжать частый пустой events.read только ради обнаружения новых событий. Сохраняются редкий контроль живости соединения, начальный снимок, явный опрос и восстановление после gap. Уведомления и их подтверждения не заменяют владельческий heartbeat и не продлевают владение. Паспорт масштаба ДОЛЖЕН указывать число подписчиков, емкость общего журнала, eventRateMax/eventBurstMax/eventDrainMinPerSec и измеренные задержки от изменения входа до приема в АРМ. Проверка

включает шторм событий и одновременные START/STOP; объявление eventsPush=1 без выполнения всего контракта ЗАПРЕЩАЕТСЯ.

Форматы диагностики и журнала

13.27 diagnostics.read: selection scope=controller|ethernet|control|ouInterface|media либо snapshotId/cursor. Ответ diagnostics с snapshotId, uptimeMs, more/cursor; записи metric name/type/value/unit, где type=bool|int|counter|enum, name и unit до 32 ASCII-символов. Обязательны uptimeMs, bootReason, activeConnections, commandQueueDepth, rejectedFrames, lostEvents и значения используемых очередей; счетчик сбрасывается только при новом bootId. Чтение не сбрасывает состояния и счетчики. log.read: selection afterSeq/limit, limit=1..8; ответ log oldestSeq/nextSeq/gap/more и записи entry seq/atMs/level/code/sessionId (последний необязателен). Правила непрерывности и gap совпадают с events.read; level=info|warning|error. Длинный свободный текст, ключи и пользовательское аудио в журнал не включаются.

13.28 command.get: selection controlEpoch/cmdSeq с bootId в оболочке; только участник исходной команды. Ответ command cmdSeq/state=accepted|rejected|expired; action обязательно при сохраненном решении и отсутствует при expired; при доступном решении дочернее тело исходного ответа совпадает с сохраненным первоначальным решением. Допустимые тела: session, cancellation, released, prepared, discarded, eof, upload, content, transaction, committed, cancelled, network, staged, clock, operation либо error. Дополнительная оболочка response внутри command запрещена; транспортные to/id исходного ответа не переносятся. При вытеснении возвращается expired, которое не означает отсутствия воздействия и не разрешает повторный session.start новым номером. Максимальный сохраненный ответ ДОЛЖЕН помещаться в общий XML-лимит вместе с оболочкой command.

Ошибка исполнения и диагностическое событие

13.29 Ошибки до accepted передаются только отказом команды: они не создают положительного подтверждения исполнения или фиктивного session.finished. После accepted неисправность передается как состояние/событие оборудования и, при влиянии на сеанс, как fault в его результате. Обычная остановка оператором, снятие аварии и штатный переход off не являются неисправностями.

13.30 powerLost / phaseLost / batteryLow

Потеря питания / фазы / низкий резерв питания. Влияние определяется реально поддержанным режимом и резервированием.

13.31 lineOpen / lineShort / amplifierFault

Обрыв линии / короткое замыкание / отказ усилителя. Указывается точный resourceId или pointId линии.

13.32 ouLinkLost / outputMismatch / sourceLost

Потеря интерфейса ОУ / несоответствие наблюдаемого выхода заданию / потеря источника звука.

13.33 overTemperature / hardwareProtection

Перегрев / срабатывание защиты. Сам факт защиты не подтверждает off без установленного критерия.

13.34 storageFault / mediaIncomplete / driverFault

Отказ хранения / нарушение условий конечной границы или доставки пригодного звука / отказ исполнительного драйвера. Отдельная потеря UDP с маскированием учитывается как mediaPacketLoss.

13.35 mediaClockFault

Недопустимая связь времени и отсчетов, потеря действующей оценки часов, превышение установленной емкости или установленный отказ тактирования. Код сам по себе не доказывает потерю UDP. Смещение уровня очереди и временный underrun не запускают таймер этого отказа.

13.36 mediaPacketLoss

Утрата позиций UDP-звука, замененных тишиной. effect=degraded, severity=warning. Поток продолжается; предупреждение и счетчики сохраняются в результате, включая succeeded/completed.

13.37 vendorFault

Условие изготовителя; conditionId указывает опубликованное описание. Новые произвольные значения закрытого code запрещены.

13.38 inventory.get передает condition id/pointId/resourceId/mode/code/severity/effect для каждого объявленного аварийного условия; mode — конкретный режим либо all. conditionId эпизода alarm ссылается на эту запись. severity=info|warning|critical задает важность, effect=none|degraded|failed|unknown — влияние на выполнение выбранного режима. Это разные оси: авария незадействованной линии не превращается автоматически в отказ другого ресурса. Аппаратный профиль определяет обязательные условия, источник, правила возникновения/снятия и зависимости ресурсов; состав и смысл меняют capabilityRevision. Неизвестное качество условия не означает его снятия.

13.39 Состояния active/cleared/unknown, alarmId, времена и правила восстановления эпизода сохраняются. Новый fault фиксируется при возникновении влияющего условия, а также для уже действующего условия при accepted. fault.effect=failed инициирует остановку; unknown означает потерю обязательного доказательства и также инициирует остановку. degraded допускает продолжение только по явно объявленному профилю. fault.alarmId связывает нарушение с точным эпизодом оборудования. Снятие обновляет clearedAtMs, но не удаляет исторический fault и не превращает failed в succeeded. Переход из degraded в failed сохраняется новой записью. Поздний session.stop не очищает отказ.

13.40 session.fault передает один fault с sessionId; alarm.raised/cleared/qualityChanged по-прежнему передают ровно один alarm. Со-

бытия исполнения и fault публикуются атомарно с stateRevision соответствующего изменения. Переполнение кольца дает gap; текущие аварии восстанавливаются через device.state, исторические нарушения принятого сеанса — через session.get. УУ резервирует емкость обязательного результата до accepted; потеря обязательной истории дает storageFault, а не пустой успешный результат.

13.41 Паспорт развертывания использует зарегистрированный параметр alarmTransitionMaxMs: целое 100..35000 мс для допуска технической проверки, не меньше суммы выбранного maxDurationMs теста, его stopConfirmMs и бюджета закрытия/нового управления. Это параметр развертывания, не обязательный limit device.capabilities. При невозможности соблюсти его тест запрещен. Для обычных исполнений 1 <= stopBeginMs <= 1000 и stopBeginMs <= stopConfirmMs <= 10000; для technical прежние более строгие пределы 100/3000 мс сохраняются. Верхние границы ограничивают соответствующий профиль, а не объявляют неисправное оборудование физически выключенным: пропуск срока остается отказом либо unknown по правилам результата.

13.42 Нижняя допустимая граница объявленного таймера — 1 мс. Объявленный срок включает дискретность таймера, худшее планирование, обмен с ОУ, фильтрацию обратной связи и погрешность измерения. Изделие ДОЛЖНО подтвердить верхнюю границу испытанием при максимальной совместной нагрузке; округление вниз и использование средней задержки вместо гарантированного предела запрещены. Если фактический предел превышает максимум профиля, соответствующая функция не объявляется.

14 Защищенный обмен и ключи

14.1 Защищенный TCP-обмен использует TLS 1.3 с российским криптонабором Кузнечик-MGM и Стрибог-256 по Р 1323565.1.030-2020 и RFC 9367. Требование российских алгоритмов распространяется на

шифрование, имитозащиту, хеширование, выработку ключей, аутентификацию и защиту хранилищ ключевой информации. В режиме tls участники ДОЛЖНЫ завершить взаимную аутентификацию до прикладных действий. Связь principalId с проверенным ключом устанавливает конфигурация доверия, а не поле из XML.

14.2 Защита ДОЛЖНА связывать сообщение с устройством, участником, направлением, подключением и поколением ключей. Аутентификация и целостность проверяются до исполнения. Прикладные полномочия и защита от повторного session.start сохраняются поверх криптографического канала. Открытое управление ДОПУСКАЕТСЯ только явно выбранным plain при securityPolicy=optional; автоматическое отключение защиты после ошибки запрещено.

14.3 Аутентификация и шифрование

Общие идентификаторы алгоритмов и версии профиля, согласование канала, целостность и аутентификация сообщений, тестовые векторы независимых реализаций.

14.4 TLS и прикладное обрамление

TLS обеспечивает защиту записей; внутри его потока находятся uint32 BE длины и XML. Лимит XML не является размером TLS record, сертификата или сообщения handshake.

14.5 Ключи и параметры TLS

Защищенный режим использует только tls13-gost-kuz-mgm/1, параметры которого зафиксированы ниже. Поколения ключей, доверие и роли задает конфигурация; nonce, ключи записей и record sequence обслуживает российский TLS-профиль.

14.6 Полные идентификаторы участников

ID аутентифицируются либо однозначно связываются при согласовании канала. Молчаливое усечение запрещено.

14.7 Модель доставки

Транспорт, порядок, потери, повторы и восстановление проверяются для каждого объявленного профиля; свойства одного транспорта не переносятся на другой автоматически.

14.8 В режиме tls ядро использует индивидуальные PSK и временный обмен на группе GC256A. Сборка прикладного сообщения ведется по длине в расшифрованном потоке; границы TLS records и TCP-сегментов с границами XML не отождествляются. Криптографические счетчики независимы от cmdSeq: повтор прикладного поручения сохраняет cmdSeq, но отправляется через TLS как новые прикладные данные с корректным состоянием защиты. Повтор TCP-сегмента обслуживается транспортом и не запускает шифрование повторно.

14.9 Для криптографических механизмов применяются российские алгоритмы. Управление и передача файлов при tls защищены TLS; при явно выбранном plain применяется открытый TCP. Живая речь передается по UDP без шифрования звука. Защищенная подготовка передает сеансовый ключ подлинности; rtp-alert/5 проверяет метку, group-audio/2 дополнительно проверяет подпись источника. Сам session.start не удостоверяет каждый аудиопакет.

Внесение доверия

14.10 Производственный профиль ДОЛЖЕН определить индивидуальное внесение ключей, роли АРМ и конфигуратора, отзыв и замену, физическое восстановление доступа и аудит без секретов. Защита сетевого канала АРМ–УУ не распространяется автоматически на внутренний интерфейс УУ–ОУ; его границы доверия и меры защиты определяются исполнением изделия. Поставка применяет криптографические средства с требуемым подтверждением соответствия требованиям ФСБ России для конкретного исполнения и условий использования; перечень алгоритмов в спецификации не заменяет такого подтверждения.

14.11 Первичный индивидуальный сервисный ключ устанавливается доверенной производственной процедурой отдельно от общего образа прошивки. Общие заводские пароли, ключи в открытом XML, журналах или обычном экспорте конфигурации запрещены. Рабочие ключи вводит участник с подтвержденными сервисными полномочиями. Ключи в АРМ, конфигураторе, резервных копиях и переносимых контейнерах защищаются средствами на российских алгоритмах с требуемым для поставки подтверждением соответствия требованиям ФСБ России. Формат контейнера определяется применяемым средством защиты; собственный парольный контейнер протокол не вводит.

Хеширование содержимого

14.12 Образы прошивок, содержимое записей и защищаемые манифесты используют hashAlgorithm=streebog256. Хеш вычисляется по точной последовательности байтов объекта от смещения 0 до объявленного размера, без дополнения, нормализации текста и преобразования кодировки. XML hash содержит 64 строчных hex-символа: по два на каждый байт результата в порядке выдачи хеш-функции, без разворота массива. Для составного манифеста хешируются байты его зафиксированного формата, а не произвольная сериализация XML.

14.13 Контроль порядка байтов: для трех байт 61 62 63 (ASCII abc, без перевода строки) hash равен 4e2919cf137ed41ec4fb6270c61826cc4fffb660341e0af3688cd0626d23b481. Скачиваемые digest-vectors содержат также пустое сообщение и пример RFC 6986, переведенный из записи битового вектора в порядок байтов файла. Эти векторы определяют представление hash на проводе независимо от API криптобиблиотеки.

14.14 Совпадение хеша подтверждает целостность относительно доверенного значения, но не право издателя на выпуск образа. Проверка подлинности обновления отделена от проверки хеша и от защиты

транспорта. Если профиль обновления использует подпись, он применяет ГОСТ Р 34.10-2012 со Стрибогом и указывает формат подписываемых данных и доверие к издателю; загрузка по защищенному каналу не объявляется автоматически secure boot.

Подтверждение совместимости

14.15 Паспорт совместимости реализации ДОЛЖЕН назвать TLS-библиотеку и версию, настройки сборки, криптографический backend, источник энтропии, пиковые память/стек и время полного handshake на целевой аппаратуре. Проверяются независимая сторона, неверные Finished/метки, открытые точки, счетчики/TLSTREE/KeyUpdate, отзыв, отказ энтропии и массовое восстановление соединений при одновременном управлении ОУ и работе локальных таймеров. RFC 9367 имеет статус Informational; ссылка на него не означает сертификацию изделия или одобрение криптонаборов IETF. Реализация ДОЛЖНА соблюдать численные бюджеты выбранного профиля при указанной совместной нагрузке.

Параметры защищенного канала

14.16 При выборе режима tls обязательный профиль защиты управляющего ядра - tls13-gost-kuz-mgm/1: TLS 1.3 с российскими криптографическими алгоритмами по Р 1323565.1.030-2020 и RFC 9367. Нормативные параметры приведены ниже и в паспорте security-0.21.json. Обе стороны допускают только этот криптонабор, режим PSK с ECDHE и группу GC256A; отсутствие совпадения завершает handshake до передачи XML. Подмена алгоритма или автоматическое отключение защиты запрещены.

14.17 Идентификатор профиля

tls13-gost-kuz-mgm/1

14.18 TLS cipher suite

TLS_GOSTR341112_256_WITH_KUZNYECHIK_MGM_L, 0xc103

14.19 Шифрование и имитозащита

Кузнечик-MGM, ключ 256 бит, полная метка 16 байт

14.20 Хеширование

Стрибог-256, ГОСТ 34.11-2018 / ГОСТ Р 34.11-2012; hashAlgorithm=streebog256

14.21 Аутентификация

Взаимная, индивидуальный external PSK 256 бит; psk_dhe_ke

14.22 Обмен ключами

GC256A, 0x0022; id-tc26-gost-3410-2012-256-paramSetA; кодирование точек по RFC 9367

14.23 Выработка ключей

TLS key schedule с HMAC/HKDF на Стрибог-256; ключи записей через TLSTREE и KDF_GOSTR3411_2012_256 по RFC 9367/7836

14.24 Размер защищенной записи

record_size_limit=512, включая внутренний тип и padding; XML до 1024 байт передается потоком

14.25 PSK длиной 256 бит выводится по open-alert-trust/2 из случайного индивидуального секрета и индивидуален для пары идентичностей устройства и участника. PSK identity - непрозрачный идентификатор ключа; запись доверия связывает его с deviceId, principalId, ролью, областью и поколением ключа. Оба Finished проверяются до protocol.hello и предметных действий. PSK-only, односторонняя аутентификация и общий ключ парка не соответствуют ядру. Самостоятельная реализация MGM не заменяет согласование TLS, TLSTREE, проверку открытых точек и управление счетчиками.

14.26 observer

Чтение описания, возможностей, состояния, датчиков, журналов и результатов в разрешенной области. Не разрешает acquire/resume и изменяющие команды.

14.27 operator

Права observer, acquire/resume и команды владельца для purpose=real, exercise и technical. Конкретный purpose acquire и session.start должен совпадать; независимые физические условия допуска сохраняются.

14.28 maintainer

Сервисные операции maintenance в разрешенной области. Не разрешает session.start и само по себе не дает права operator или trustAdmin.

14.29 trustAdmin

Управление записями доверия и ключами по профилю commissioning. Не разрешает acquire/resume, session.start или обслуживание maintainer.

14.30 Сочетание ролей

Полномочия объединяются только явным перечнем ролей; одна роль не выводится из другой. Ресурсы, состояние ключа и предметные условия каждой команды проверяются отдельно.

Случайные значения и временные ключи

14.31 Приватные временные ключи GC256A и случайные значения handshake получаются от криптографического генератора с обоснованным источником энтропии. MAC, IP, uptime, bootId и постоянный счетчик не заменяют случайность. При отказе генератора новый защищенный канал не устанавливается. PSK используется только по назначению выбранного профиля и не переносится в иной протокол или файл как ключ шифрования данных.

Ограничения записей и handshake

14.32 Обе стороны ДОЛЖНЫ согласовать record_size_limit=512 по RFC 8449; отсутствие расширения либо иное значение отклоняется. Для TLS 1.3 это длина TLSInnerPlaintext, включая content type и padding, но

без заголовка и метки MGM. Без padding запись несет до 511 прикладных байт. Рамка uint32 BE + XML длиной 1024 байта содержит 1028 байт и требует как минимум трех защищенных записей при максимальном заполнении. Число TCP-сегментов из этого не следует. Граница записи может проходить внутри поля длины или XML-атрибута.

14.33 record_size_limit не ограничивает незашифрованные сообщения начального handshake. Приемник ядра ограничивает TLSPlaintext.fragment незашифрованной записи и каждое сообщение ClientHello/ServerHello/HelloRetryRequest с его четырехбайтовым handshake-заголовком величиной 4096 байт; проверка размера выполняется до выделения буфера. Handshake разбирается порционно с сохранением управления ОУ и обслуживания локальных таймеров. После проверки MGM в прикладной парсер поступают только аутентифицированные байты; ошибка метки закрывает канал, не выдает частичный XML и не продлевает lease владельца.

14.34 Ядро запрещает 0-RTT и TLS session tickets/resumption: каждое новое подключение проходит полный PSK-DHE handshake и текущую проверку допуска. Полный handshake ограничен 60000 мс от принятия TCP. Этот предел относится к подготовке нового канала и не является допустимой задержкой START/STOP на готовом канале. Два дорогих контекста PSK-DHE выделяются только после проверки PSK binder; неаутентифицированный TCP-клиент не занимает эти контексты. До проверки binder применяется отдельный ограниченный прием ClientHello по правилам ниже. Существующий владелец, его heartbeat и session.stop обслуживаются независимо от этой очереди.

14.35 Срок handshake проверяется внутри длительных вычислений в конечных точках прогресса. Дополнительно задается независимый конечный бюджет вычислительной работы на операцию. По исчерпании срока или бюджета расчет прекращается с очисткой временных секретов и освобождением контекста; следующий дорогой этап не начинает-

ся. Паспорт указывает наибольший промежуток между точками прогресса и завершение очистки. АРМ подготавливает канал до команды оператора, показывает его готовность и отменяет неотправленный START при истечении срока намерения. Ожидание нового канала не разрешает запоздалый запуск.

Ротация и отзыв ключей

14.36 Ключ проходит состояния staged, active и revoked. Участник с правом управления доверием подготавливает индивидуальный новый ключ по доверенному сервисному пути; обе стороны подтверждают его установку отдельным пробным handshake до отзыва старого. В записи доверия хранятся поколение, состояние и срок разрешенного перекрытия. Отзыв и старший номер поколения атомарно записываются в энергонезависимую память до отправки подтверждения операции и сохраняются без срока, в том числе после подтверждения и reboot; откат к старой резервной копии не восстанавливает отозванное поколение. Аварийное восстановление доступа выполняется местной авторизованной процедурой с аудитом.

14.37 Ключ staged допускает только пробный handshake и protocol.hello для подтверждения идентичности; управляющие и сервисные команды до перехода active запрещены. Активация требует полномочий действующего администратора доверия либо местной процедуры восстановления. Отозванный ключ не может подтвердить собственную замену.

14.38 После применения отзыва УУ ДОЛЖЕН отклонять новые handshake этим ключом, закрыть его действующие соединения и аннулировать связанные права до следующей изменяющей команды. Отзыв полномочий, на которых основано текущее владение, немедленно прекращает действие его lease и выполняет stop активного исполнения независимо от onControlLoss=finish. Это самостоятельное событие, отлич-

ное от закрытия канала по таймауту; ожидание остатка lease после отзыва запрещено. Отзыв роли не передает сеанс другому участнику; отозванные полномочия не допускают resume. Плановая ротация ключа того же principalId требует нового handshake и resume до отзыва прежнего канала; TLS KeyUpdate не заменяет ротацию идентичности.

14.39 При плановой ротации без потери управления новый active-ключ того же principalId ДОЛЖЕН успешно пройти handshake, затем единственный писатель закрывает прежний управляющий канал и выполняет resume до отзыва старого. После атомарного прикрепления нового канала отзыв только прежнего ключа не аннулирует владение, уже основанное на новом действующем ключе. Отзыв самого права управления principalId действует независимо от ключа. Эта последовательность не отменяет требований к единственному активному экземпляру АРМ.

Первичное доверие и ключи издателя образов

14.40 Чистое УУ с securityPolicy=optional допускает открытый режим по разделу 14. При secureOnly отсутствие доверия запрещает сетевые изменяющие команды и session.start. Локальная авторизованная процедура при подтвержденном off фиксирует deviceId и открытый ключ начального администратора доверия; оператор проверяет его отпечаток (хеш Стрибог-256 от 65 байт 04 || x[32] || y[32], координаты big-endian; результат 32 байта, 64 строчных hex) по независимому доверенному источнику. Подпись, содержащаяся рядом с неизвестным ключом, не заменяет эту проверку. Общий заводской пароль или общий PSK парка не используется. Способ физического доступа и проверки личности оператора указывается в паспорте изделия; сетевой профиль не создает доверие из недоверенного канала.

14.41 Переносимая запись доверия имеет формат open-alert-trust/2 по trust-schema текущей редакции: payload содержит deviceId, generation,

principals, firmwareKeys; envelope содержит payload, authorityKeyId, signature. Подписывается ровно JCS(payload) в UTF-8 без BOM и завершающего перевода строки, без полей envelope. authorityKeyId выбирает заранее закрепленный ключ администратора GC256A; неизвестный ключ не становится доверенным из подписи. Алгоритм — ГОСТ Р 34.10-2012 с кривой GC256A и хешем Стрибог-256. signature содержит 128 строчных hex от r[32] little-endian || s[32] little-endian по единому правилу подписи ниже; DER и автоматическое распознавание кодировки запрещены. generation положителен и строго больше сохраненного; импорт с другим deviceId запрещен. Пакет содержит только открытые параметры и права: PSK, корневые секреты и закрытые ключи в нем запрещены. Подпись обеспечивает подлинность и целостность, но не конфиденциальность состава реестра. Сетевые config.stage и firmware.begin не импортируют доверие.

14.42 principal содержит principalId, keyId, generation, state=staged|active|revoked, roles и resourceIds; для staged/active обязательны rootId и salt (64 строчных hex), для revoked они отсутствуют. rootId выбирает отдельно установленный 256-битный корневой секрет Kpair именно пары deviceId/principalId. Ключи разных пар генерируются независимо криптографическим генератором; общий ключ парка и вывод всех пар из доступного участникам общего секрета запрещены. Роли observer/operator/maintainer/trustAdmin независимы; trustAdmin не разрешает session.start. resourceIds — точный непустой список локальных ресурсов, без wildcard. keyId уникален в хранилище УУ, не переиспользуется и служит TLS PSK identity: ровно 1..32 ASCII-байта token, без завершающего NUL, префикса, hex-декодирования или Unicode-нормализации; регистр значим. Поле identity имеет стандартную длину uint16 BE из TLS 1.3, которая не входит в эти байты; obfuscated_ticket_age=0 для внешнего PSK. ClientHello содержит ровно одну identity. Пробный handshake staged не

разрешает управление; активация оформляется новой подписанной редакцией.

14.43 firmwareKey содержит keyId, generation, state, curve=GC256A, x, y, deviceKinds и hardwareRevisions. x/y - по 64 строчных hex, целые big-endian; проверяются принадлежность кривой, ненулевая точка и порядок подгруппы. Ключ ограничен явно перечисленными типами и ревизиями аппаратуры. keyId прошивки относится только к этому пространству издателей и не является TLS identity. staged допускает проверку пробного образа, active допускает verify/commit, revoked запрещает новый begin/verify/commit. Право обслуживания и знание PSK не дают права подписать образ или добавить издателя.

14.44 Импорт доверия проверяет подпись закрепленного администратора, схему, ссылки, поколения и права; фиксируется атомарно с чтением для проверки до ответа об успехе. Прежние keyId и старшие поколения сохраняются как запреты возврата, даже если отсутствуют в новом пакете; неотозванные записи запрещено молча удалять. Смена материала под прежним keyId запрещена. Переходы staged→active→revoked и staged→revoked необратимы; новая пара ключей получает новый keyId. Обрыв питания оставляет целиком прежнее либо целиком новое состояние. Отзыв издателя перепроверяется непосредственно перед commit и при активации после reboot; проверенный ранее образ не обходит отзыв. Отзыв не выключает уже работающий образ сам по себе; последующая загрузка и восстановление используют только разрешенный образ и сохраненный минимальный номер версии.

14.45 Ротация издателя включает импорт нового staged-ключа, проверку пробного подписанного образа, активацию отдельным пакетом и последующий отзыв старого. До отзыва обеспечивается разрешенный загрузочный и восстановительный образ. Замена корня администратора и восстановление после утраты всех ключей требуют той же локальной проверки независимого отпечатка, подтвержденного off и аудита;

старшие версии, запреты и идентичность не сбрасываются. Аудит содержит поколения, keyId, отпечатки, инициатора и результат, но не PSK и приватные ключи.

14.46 Канонизация пакета доверия: principals и firmwareKeys сортируются по keyId, roles/resourceIds/deviceKinds/hardwareRevisions - лексикографически, повторы запрещены. generation каждой записи положителен и при изменении записи строго увеличивается. Сохраненный материал revoked-ключа издателя остается в журнале отзыва; производный PSK стирается. Отозванный keyId НЕ ДОПУСКАЕТСЯ независимо от возможности повторно вычислить его PSK. rootId/salt/principalId под прежним keyId неизменны; изменение прав и состояния увеличивает generation записи, но не меняет производный ключ. Схема проверяет форму, предметная проверка - диапазон uint64, ссылочную целостность, переходы и подпись. Полномочия trustAdmin разрешают импорт уже подписанного закрепленным корнем пакета, но не заменяют его подпись.

14.47 Kpair устанавливается отдельно от публичного пакета и общей прошивки в защищенное хранилище обоих участников. На чистом УУ первичная установка требует физически контролируемой авторизованной процедуры при off и независимой проверки deviceId/principalId/rootId; обычный сетевой интерфейс первоначальные секреты не принимает. Формат записи для локального импортера: ASCII OAK1 (4 байта), deviceId (16 байт из hex), principalId (16), длина rootId (uint8, 1..32), rootId (ASCII), Kpair (32 байта); всего 70..101 байт, без дополнений. Это ввод через локальный защищенный интерфейс, не незашифрованный переносимый файл. Импортер проверяет полную длину, отсутствие прежнего rootId и соответствие выбранному устройству, сохраняет атомарно и стирает временный буфер. Подтверждение выдается после чтения записи и пробного TLS handshake; обрыв оставляет прежнее либо

полное новое состояние. Хранение этой записи в публичном реестре, журнале, общей прошивке или открытом экспортном файле запрещено.

14.48 PSK пары вычисляется по KDF_GOSTR3411_2012_256 из RFC 7836, пункт 4.5. label — ASCII OpenAlert/TLS-PSK/1 без нуля в конце; seed — Стрибог-256 от JCS объекта с ровно пятью строковыми полями deviceId, principalId, keyId, rootId, salt. Вход HMAC: 0x01 || label || 0x00 || seed[32] || 0x01 || 0x00; ключ HMAC — Kpair, результат — 32 байта PSK. Никакого AES, SHA-256 или парольного материала в этом выводе нет. salt — новые 32 случайных байта для нового keyId. Права и generation записи не входят в seed, поэтому переход staged в active не меняет проверенный ключ. Направления и рабочие ключи соединения выводит только TLS; Kpair не используется как TLS PSK или ключ шифрования файла.

14.49 Компрометация публичного trust-пакета не раскрывает PSK. Компрометация одного Kpair ограничивается данной парой, но раскрывает ее производные PSK: для восстановления заменяются Kpair/rootId и все затронутые keyId с отзывом старых. Простая смена salt при украденном Kpair не восстанавливает доверие. Корневые секреты хранятся отдельно от открытого реестра с индивидуальными правами доступа и защищенными резервными копиями; хранилище, имеющее доступ ко всем секретам, остается отдельной критичной частью системы. Невыгружаемые ключи и аппаратное хранилище допустимы, но не подменяют разграничение пар.

Допуск соединений до выделения дорогого контекста

14.50 Прием TCP/ClientHello и выполнение PSK-DHE имеют разные квоты. Для начального приема выделены четыре короткоживущих места; каждое принимает не более 4096 байт ClientHello, без выделения ECDHE. Первый полный ClientHello ДОЛЖЕН поступить за 3000 мс от приема TCP, вся предварительная проверка — за 8000 мс. Истечение

срока закрывает только этот сокет. При заполнении четырех мест новое соединение отклоняется; непрерывное поступление чужих соединений не продлевает сроки занятых мест. TCP SYN-очередь имеет отдельный конечный предел и защиту от полуоткрытых соединений; ее память не отнимается у действующего управления.

14.51 ClientHello содержит ровно одну PSK identity. До ECDHE УУ проверяет структуру, разрешенный keyId и binder по RFC 8446. Затем УУ обязательно посылает HelloRetryRequest с новым случайным cookie из 32 байт на этом соединении, без изменения выбранной группы. Сохраняются cookie и хеш первого ClientHello для стандартного transcript message_hash; повторный ClientHello и его binder проверяются по полному транскрипту с HelloRetryRequest. Второй HelloRetryRequest запрещен. Cookie принимается только на выдавшем его TCP-соединении, один раз, до исходного срока 8000 мс; закрытие стирает контекст. Повтор записанного первого ClientHello или знание публичной identity не доказывают право занять дорогой слот. Cookie не заменяет binder и Finished.

14.52 После успешной проверки повторного binder выделяется один из двух контекстов PSK-DHE. Пока существует владение, один контекст зарезервирован для текущего principalId, второй — для остальных допущенных участников; посторонний участник не занимает резерв. При отсутствии владения оба доступны допущенным участникам в порядке завершения предварительной проверки. На principalId ДОПУСКАЕТСЯ не более одного незавершенного дорогого контекста, включая все его keyId и TCP-адреса; второй запрос того же участника не занимает другой слот и не вытесняет первый. Полный handshake завершается не позже 60000 мс от приема TCP. При ожидании первого байта также действует срок первого ClientHello 3000 мс; ни одна фаза не начинает общий срок заново. Существующие TLS-соединения, медиа, bulk, взаимодействие с ОУ и таймеры session.stop имеют отдельные ресурсы и

продолжают обслуживаться; отказ выделения контекста не отзывает владельца. Отзыв ключа проверяется повторно перед Finished и допуском команд.

14.53 Приемка воспроизводит два молчащих TCP-клиента, медленный ClientHello, неверный binder, повтор захваченного ClientHello и отказ генератора случайных чисел. Два молчащих клиента не блокируют корректное подключение владельца через оставшиеся места. Дополнительно измеряются пределы частоты проверок binder, памяти и восстановления при нагрузке, заявленной для изделия. Конечный контроллер не гарантирует доступность при насыщении канала или исчерпании всех мест непрерывным потоком атакующих подключений; защита периметра сохраняется. Эти ограничения не оправдывают прежнее занятие двух дорогих слотов молчащими клиентами на 10 секунд.

14.54 Прием ClientHello на канале от 64 кбит/с допускает до 3000 мс от приема TCP до полного сообщения, не более 4096 байт; весь предварительный обмен с cookie завершается за 8000 мс. Дорогие контексты по-прежнему выделяются только после binder и свежего cookie. Полный handshake завершается за 60000 мс от приема TCP. Четыре ограниченных приемных места и резерв дорогого контекста владельца не гарантируют доступность при насыщении входного канала; размеры, очередь и сроки всех фаз проверяются раздельно. Повторы подключения имеют случайную задержку по правилам backoff и не вытесняют действующее управление.

Явный выбор открытого и защищенного обмена

Таблица 15 - Явный выбор открытого и защищенного обмена

Настройка устройства	Допустимое поведение
securityPolicy=optional	Разрешены явно выбранные клиентом TLS и открытый TCP. Наличие ключей не запрещает открытый канал. Чистое устройство без ключей работает по открытому TCP с локально заданными правами.
securityPolicy=secureOnly	Разрешен только TLS. При отсутствии действующих ключей устройство не принимает сетевое управление и сообщает неготовность локально; переход на открытый TCP запрещен.

14.55 В новом пустом хранилище securityPolicy=optional. Настройка «Только защищенный канал» устанавливается при местном вводе; ее отключение ДОПУСКАЕТСЯ только местно при подтвержденном off. Изменение атомарно; переход в secureOnly ДОПУСКАЕТСЯ после успешного пробного TLS с active-ключом, отзывает открытые привязки и останавливает их активные исполнения независимо от onControlLoss=finish. Сброс ключей не сбрасывает secureOnly. Команда открытого канала не меняет эту настройку и не устанавливает доверие.

14.56 В АРМ, конфигураторе, эмуляторе и прошивке режим соединения выбирается явно: tls либо plain. В АРМ это настройка устройства Открытый протокол систем оповещения «Защищенный обмен»: при выключении поля ключей не требуются; состояние показывает «Открытый

канал». Настройка УУ «Только защищенный канал» имеет приоритет над выбором клиента. Ошибка ключа, TLS или таймаут не вызывают автоматического повторения через plain. Открытый режим поддерживает обычные команды управления, оповещения и чтения в пределах локальных прав; он не является только диагностическим режимом.

14.57 Один TCP-порт может принимать оба режима. До разбора первого сообщения соединение имеет состояние undecided: первый байт 0x16 выбирает TLS, 0x00 — открытую четырехбайтовую рамку XML длиной 1..1024; любое другое значение закрывает соединение. При secureOnly вариант 0x00 немедленно отклоняется. Выбор закрепляется на срок соединения: неверный TLS никогда не передается XML-парсеру и наоборот. Возможны отдельные явно настроенные порты с теми же правилами. Открытый клиент сразу передает protocol.hello; защищенный — после завершенного TLS. Оба используют один wire-профиль и размеры XML.

14.58 Открытый канал не проверяет PSK и не доказывает личность отправителя. В protocol.hello его элемент hello содержит обязательный clientId (32 строчных hex), постоянно сохраняемый клиентом; при tls этот атрибут запрещен. УУ использует пару plain/clientId как логическую идентичность открытого клиента, чтобы разделять последовательности и повторы, но не как доказательство полномочий. Все открытые клиенты получают одни локально настроенные plainRoles/plainResourceIds; clientId не выбирает привилегированную учетную запись. В пустой конфигурации базовые роли observer/operator и явно объявленные локальные ресурсы не дают открытого real/exercise/technical: эти назначения требуют отдельного разрешения в локальной политике. АРМ не меняет clientId при обычном переподключении. Кто скопировал clientId, неотличим от прежнего открытого клиента; это ограничение plain, а не защищенная идентификация. IP-фильтр также не объявляется криптографической

аутентификацией. Проверки эпохи, привязки, повторов, занятости, ресурсов и физического результата сохраняются.

14.59 Требования о криптографически аутентифицированном principalId относятся к tls. Для plain вместо него используется явно обозначенная логическая идентичность plain/clientId с описанной границей доверия. Контекст владения и mediaId/transferId фиксирует также режим; открытый канал не может resume, heartbeat, session.stop или продолжить операцию, принятую от TLS-участника. Переход между режимами требует завершения прежнего владения и нового acquire. Это разделение не ограничивает местную аварийную остановку и не делает открытое управление защищенным.

14.60 device.describe сообщает securityPolicy=optional|secureOnly. protocol.hello сообщает security=tls|plain, совпадающий с фактическим управлением. Режим отображается в диагностике. Файлы используют TLS при tls и открытый TCP при plain с привязкой transfer.hello. Живая речь использует UDP: при tls проверяется метка, при plain криптографической проверки нет. Звук не шифруется в обоих режимах. Подпись прошивки и проверка издателя сохраняются независимо от транспорта.

14.61 Режим без ключей является разрешенным эксплуатационным выбором, а не обещанием защиты от подмены. Для tls сохраняются запреты отключения криптографии и все требования TLS-профиля. Проверки открытого режима включают работу чистого устройства, отсутствие запросов ключа в АРМ, отказ plain при secureOnly, отсутствие автоматического понижения защиты и невозможность вмешательства plain в TLS-владение. Изменение касается только Открытый протокол систем оповещения; другие продуктовые протоколы сохраняют свои правила.

Первичный ввод и первое подключение

14.62 Настоящий порядок объединяет правила первичного доверия, open-alert-trust/2 и защищенного соединения для совместимых изделий любых изготовителей. Он применяется при выборе tls. Явно выбранный plain при securityPolicy=optional допускает работу без выдачи ключей; регистрация открытого клиента не устанавливает криптографическое доверие. Первая запись прошивки, внесение доверия и регистрация в АРМ являются отдельными действиями, даже если сервисный инструмент выполняет их в одном задании.

14.63 Администратор доверия определяет допущенных участников, их роли и ресурсы, утверждает подписанные пакеты доверия. Уполномоченный установщик выполняет местные действия и проверяет соответствие выбранному устройству. АРМ и конфигуратор действуют от собственных principalId и только с назначенными правами. Совмещение функций одним инструментом ДОПУСКАЕТСЯ при сохранении этих границ; роль trustAdmin сама по себе не разрешает управление оповещением. Изготовитель определяет физический способ доступа, проверку полномочий установщика и защищенный путь внесения данных в паспорте изделия.

14.64 Шаг 1. Установщик устанавливает соответствие физического устройства его deviceId и проверяет подтвержденный off перед местным внесением доверия. При неизвестном состоянии выхода применяется процедура восстановления устройства; неизвестное состояние не считается выключением. Способ первичного программирования и начальное состояние хранилища определяются паспортом изделия; общее программное обеспечение не содержит индивидуальных секретов. Обычное обновление программы не создает новую идентичность и не сбрасывает сохраненные запреты возврата.

14.65 Шаг 2. Местная авторизованная процедура закрепляет открытый ключ администратора доверия. Установщик сверяет его отпе-

чаток Стрибог-256 по независимому доверенному источнику согласно правилам первичного доверия. Ключ из обнаруженного сетевого объявления или из пакета с подписью этого же неизвестного ключа не принимается как основание доверия. Ключ администратора, ключ издателя прошивки и секрет пары выполняют разные функции; полномочие одного не подменяет остальные.

14.66 Шаг 3. Для выбранных deviceId и principalId создается независимый случайный Kpair длиной 256 бит и назначается rootId. Защищенный контур ввода использует криптографический генератор и устанавливает секрет в защищенные хранилища обоих участников. Локальный импортер получает запись OAK1 установленного формата, проверяет идентификаторы и полную длину, атомарно сохраняет данные и стирает временный буфер. Обычный сетевой интерфейс не принимает первоначальный секрет. rootId, уже присутствующий в хранилище, не используется для перезаписи другим материалом. Завершение записи на одной стороне не означает завершения выдачи паре.

14.67 Шаг 4. Администратор выпускает подписанный пакет open-alert-trust/2 с новым keyId, новым случайным salt, состоянием staged, согласованными rootId и principalId, ролями и точным списком ресурсов. УУ проверяет закрепленного подписанта, deviceId, поколения, схему и связи, затем атомарно фиксирует пакет. Клиент использует соответствующую запись той же пары. Стороны выводят PSK по установленному KDF; секреты не включаются в пакет доверия. Подготовленная запись не дает права управления. Сохранение Kpair и применение staged-записи предшествуют пробному соединению; подтверждение завершенной установки выдается после этого соединения, как установлено для локального импортера.

14.68 Шаг 5. Клиент явно выбирает tls и выполняет полный handshake с подготовленным keyId. Обе стороны проверяют Finished; затем клиент передает protocol.hello профиля bootstrap/1 и сверяет

deviceId, selectedProfile, bootId и security=tls с выбранным устройством и фактическим каналом. Пробное соединение с staged-ключом допускает только handshake и protocol.hello; acquire, session.start и сервисные команды не используются для проверки ключа. Подтверждение относится к конкретным deviceId, principalId, keyId, rootId и salt. Неверная идентичность, несовместимый профиль или ошибка защиты прекращают процедуру без активации и без перехода на plain.

14.69 Шаг 6. После успешного пробного соединения уполномоченный администратор оформляет переход staged в active новой подписанной редакцией пакета и увеличением поколения измененной записи. keyId, rootId, salt и principalId сохраняются; переход состояния не меняет выведенный PSK. Факт успешного handshake сам по себе не изменяет запись доверия. Клиент устанавливает новое полное TLS-соединение действующим ключом и повторяет protocol.hello. Продолжение пробного соединения не заменяет проверку действующего допуска.

14.70 Шаг 7. Через рабочее соединение АРМ читает device.describe, device.capabilities и device.state в пределах назначенных прав, сверяет идентичность, тип, ресурсы, ревизии и ограничения. Регистрация связывает логическую цель с конкретным deviceId, resourceId и bindingGeneration по правилам раздела регистрации. Успех подключения не запускает ОУ и не является подтверждением готовности всех его физических функций. Для управления отдельно выполняются проверки прав, ресурсов и acquire; право прерывания сохраняет установленную связь с происхождением запуска.

14.71 Если выбран режим "Только защищенный канал", местная процедура устанавливает secureOnly после успешного пробного соединения с active-ключом по правилам явного выбора транспорта. Наличие ключа само по себе не меняет optional на secureOnly. Применение secureOnly аннулирует открытые привязки и останавливает их активные исполнения независимо от onControlLoss=finish. Итог ввода различает

записанный секрет, подготовленный ключ, активный ключ и установленное рабочее соединение; это результаты процедуры, а не новые состояния сетевого протокола.

Хранение и учет выдачи ключей

14.72 Для другого principalId на том же устройстве и для другого устройства у того же участника создаются независимые секреты пар. Общая учетная запись или резервный экземпляр АРМ не отменяют требования единственного активного экземпляра участника. Передача полномочий другому участнику выполняется новой регистрацией и назначением прав; копирование чужой записи Kpair не заменяет эту процедуру.

14.73 Хранение и перенос секретов выполняются средствами защиты, установленными разделом безопасности, включая защищенные резервные копии. OAK1 является записью локального импортера и не становится форматом открытого переносимого файла. Секреты исключаются из общего образа прошивки, публичного пакета доверия, обычного экспорта конфигурации и журналов. Учет выдачи содержит идентификаторы участников и ключей, поколения, отпечатки открытых ключей, инициатора и результат; значения Kpair, PSK и закрытых ключей в учет не включаются.

Продолжение процедуры после отказа

14.74 После обрыва связи или питания установщик сначала устанавливает фактически сохраненные идентичности, поколения и состояния через авторизованный местный интерфейс либо доступный действующий сервисный путь. Новое доверие не выводится из прежнего сообщения об успехе или из наличия файла на компьютере. В пределах одного хранилища сохраняется прежняя либо полная новая запись; атомарность двух раздельных участников не предполагается.

14.75 Если секрет записан только на одной стороне, завершается внесение той же пары на вторую сторону через защищенный путь. Уже сохраненный rootId не перезаписывается повторным импортом OAK1. Если соответствующий секрет недоступен либо его происхождение вызывает сомнение, применяются отзыв и местное восстановление с новым Kpair/rootId; случайная замена секрета под прежними идентификаторами запрещена. Незавершенная выдача не дает готовности защищенного соединения.

14.76 Если сохранен staged, повторяется пробное соединение с этой записью; ошибка или таймаут не переводят ее в active. Если подтверждение применения active потеряно, проверяется сохраненное состояние и устанавливается новое соединение с тем же ключом. Уже примененная редакция не выдается за новое изменение; пакет со старым поколением не обходит запрет отката. Отзыв остается действующим и при потере подтверждения. Отказ генератора случайных чисел прекращает создание ключа или нового защищенного соединения без подстановки предсказуемого материала.

Плановая замена ключа

14.77 При плановой замене без компрометации для той же пары ДОПУСКАЕТСЯ использовать действующий Kpair: назначаются новые keyId и salt, затем выполняются подготовка staged, пробное соединение и активация. Старый ключ остается действующим только в разрешенном интервале перекрытия. Изменение generation или TLS KeyUpdate само по себе не является заменой ключа участника.

14.78 При действующем владении новый active-ключ того же principalId проходит handshake. Единственный активный экземпляр APM закрывает прежний управляющий канал, выполняет resume и подтверждает прикрепление к существующей эпохе до отзыва старого ключа. Новый session.start вместо resume запрещен. Отзыв только прежнего

ключа после успешного прикрепления не отзывает владение, основанное на новом ключе. Если переход не завершен, применяются обычные сроки владения и политика потери управления; наличие нового ключа не продлевает их.

Отзыв, утрата ключей и замена устройства

14.79 При компрометации применяется отзыв затронутых ключей и полномочий по правилам безопасности. Отозванные ключи отвергаются, их соединения закрываются, а связанное активное исполнение останавливается независимо от настройки onControlLoss=finish. Плановая ротация сначала прикрепляет новый действующий ключ через resume и не является аварийным отзывом текущей привязки. Если раскрыт Kpair, меняются Kpair/rootId и все производные keyId этой пары; новая соль при прежнем секрете не восстанавливает доверие. Новое доверие не подтверждается отозванным участником.

14.80 При утрате всех действующих ключей или замене корня администратора выполняется местная авторизованная процедура при подтвержденном off с независимой проверкой отпечатка и аудитом. Восстановление сохраняет deviceId, старшие поколения, отозванные идентификаторы, минимальную разрешенную версию и securityPolicy. Восстановление резервной копии не отменяет более поздний отзыв; копия с утраченными сведениями о старших поколениях не становится основанием допуска. Сброс секретов не выключает secureOnly. При отсутствии действующего доверия в secureOnly сетевое управление остается недоступным до завершения местного восстановления.

14.81 Замена физического устройства использует его собственную идентичность и новую привязку ресурсов по разделу регистрации. Перенос совместимой конфигурации не переносит Kpair, PSK, активное владение или сеанс прежнего устройства. Совпадение IP-адреса не подтверждает замену и не позволяет принять ответ нового устройства за

результат прежнего запуска. Восстановленный либо новый экземпляр АРМ также соблюдает запрет вытеснения действующего владельца.

Параметры процедуры в паспорте изделия

14.82 Паспорт изделия ДОЛЖЕН конкретизировать способы выполнения установленной процедуры. Эти сведения определяют местный сервисный путь; они не вводят общие для всех изготовителей разъем, программатор, загрузчик или интерфейс приложения. Паспорт указывает:

14.83 способ первичного программирования, местного входа в сервисный режим, проверки полномочий установщика и подтверждения off;

14.84 способ независимой проверки deviceId и отпечатка администратора, назначения principalId и области ресурсов;

14.85 защищенный путь импорта OAK1 и подписанного пакета доверия, средства хранения и переноса секретов, источник случайности;

14.86 способ чтения сохраненных поколений и состояний без выдачи секретов, признаки завершения каждого шага и порядок продолжения после отказа;

14.87 порядок резервирования, плановой замены, отзыва и местного восстановления с сохранением запретов возврата;

14.88 место хранения аудита, состав доступных оператору результатов и способ явного выбора optional либо secureOnly.

15 Речь, записи и время

Речь передается локальной записью, потоком записи либо микрофоном. Для сети применяется rtp-alert/5 настоящего раздела; для автономного последовательного оповещения - mode=program раздела 26. Управление всеми режимами использует явно выбранный tls либо plain, точные ресурсы и общие критерии физического результата.

Три способа передачи речи

15.1 Локальная запись

Где находится источник: Полное проверенное содержимое доступно на УУ/ОУ до session.start.

Необходимый контракт: Размещение и атомарная публикация содержимого, идентичность и хеш, готовность, запуск одного прохода, естественное окончание, отмена и физический результат. Передача аудио по сети во время исполнения не обязательна.

15.2 Потоковая запись

Где находится источник: Файл хранится на АРМ и передается получателям во время исполнения.

Необходимый контракт: Кроме неизменного содержимого нужны доставка потока с явно выбранным режимом tls либо plain, подготовка получателей, временная шкала, ограниченный буфер, учет потерь и точная граница окончания. Звук не шифруется. Наличие файла на АРМ не делает его локальной записью УУ.

15.3 Микрофон

Где находится источник: Отсчеты создаются разрешенным источником в реальном времени.

Необходимый контракт: Нужны доставка потока и право говорить: выдача, удержание, отпускание и отзыв тангенты, смена источника, предел задержки и реакция на потерю аудио отдельно от потери управления. Полного файла и хеша будущей речи до начала нет.

15.4 Локальный режим mode=recording выбирает неизменяемый объект, доступный УУ или ОУ целиком до session.start: contentId/revision/hashAlgorithm/hash. Один session.start выполняет один проход. Файл с АРМ во время исполнения передается через mode=stream и audio.prepare. Подмена локальной записи URL или сетевым источником в поле contentId запрещена.

Запись на стороне исполнения

15.5 Размещение

Производственная загрузка допустима без сетевого аудиоканала. Сетевое обновление контента использует команды раздела 15: ограниченные блоки, повтор/продолжение, проверка полного объекта, атомарное применение и восстановление после обрыва питания. Сетевая доставка использует content.begin, передачу байтов по bulk/1 и content.commit; до commit объект не доступен для запуска.

15.6 Идентичность

Манифест фиксирует точные байты, размер, кодек и параметры декодирования, длительность либо число отсчетов, неизменные contentId/revision и Стрибог-256. Хеш контейнера и хеш декодированного звука не взаимозаменяемы; объект хеширования объявляется явно.

15.7 Готовность

До воздействия проверены доступность всей записи, ее идентичность, целостность и возможность исполнения на выбранных ресурсах. Полный хеш может быть заранее вычислен при размещении; session.start проверяет актуальность сохраненного доказательства и неизменность объекта. Номер слота или общий режим ОУ не доказывает выбранное содержимое.

15.8 Естественный конец

Один проход завершается только по объявленному критерию окончания именно этой записи. Истечение maxDurationMs, возврат общего idle без достаточного контекста и отсутствие связи не являются доказательством полного воспроизведения.

15.9 Замена и повтор

Принятый сеанс удерживает неизменную версию до конца. Замена слота не меняет исполняемый объект. Новый проход является новым действием; повтор сетевой команды не начинает воспроизведение сначала.

Контракт потока и параметры аудио

15.10 Формат аудио

Точный кодек и вариант, частота, каналы, число отсчетов в пакете, разрешенный последний неполный пакет, порядок байтов и битов, эталонные байты кодирования. Изменение параметров посреди потока без нового согласования запрещено.

15.11 Заголовок и шкала

Побайтовая схема заголовка, длины, номера пакетов и отсчетов, начальные значения, поведение при переполнении, поле полезной длины, границы сообщения и максимальный пакет с защитой. Позиция отсчета задает порядок воспроизведения, а не адрес записи в память устройства.

15.12 Доставка

Один точный набор транспорта и защиты на profileId; направление установления связи, адреса/порты, unicast либо multicast, подготовка каждого получателя и правила маршрутизации. Формулировка "UDP/RTP" не выбирает такой набор.

15.13 Разрешение потока

Защищенное управление связывает поток с bootId, controlEpoch, sessionId, ресурсами, разрешенным источником и mediaId. До помещения в воспроизводимый буфер проверяются разрешенный контекст, свежесть, окно повторов и условия допуска источника выбранного профиля. В режиме tls проверка метки обязательна до приема пакета. В явно выбранном plain источник криптографически не удостоверяется; этот режим не объявляется защищенным. IP, порт и SSRC сами по себе не доказывают подлинность отправителя.

15.14 Начало

Условия готовности тракта, срок первого пригодного пакета, порог стартового буфера, правило запуска воспроизведения и предельная задержка. Первое пришедшее сообщение не назначает источник, контекст

или параметры потока. Готовность каждого получателя подтверждается отдельно.

15.15 Буфер

Начальный и максимальный объем в отсчетах и миллисекундах, окно перестановки, срок опоздания, поведение при переполнении и underrun. Дубликаты не воспроизводятся повторно; поздний пакет не вставляется в уже выданный участок. Отставание не накапливается без предела.

15.16 Потери

Политика восстановления до срока воспроизведения, допустимость повторной передачи/FEC, замены пропуска тишиной или алгоритмом маскирования. Подмененные/пропущенные отсчеты учитываются в результате; восстановление прошлого участка не вызывает позднего повторного звучания.

15.17 Окончание

Конечная граница в той же шкале отсчетов передается по управляющему и медиаканалу выбранного режима; в tls оба аутентифицированы, в plain криптографической проверки нет. Определяются подтверждение ее принятия, идемпотентный повтор, ограниченный срок дренирования и отсутствие новых данных после границы. Потеря источника или таймаут не подменяются штатным EOF.

15.18 Остановка

Адресная отмена отзывает поток, запрещает новые пакеты и отменяет отложенные действия, очищает очереди до аппаратного вывода. Неотменяемый остаток аппаратного буфера входит в измеренный бюджет session.stop; очистка одной сетевой очереди недостаточна.

15.19 Результат

Состояние тракта, принятые и выданные в тракт отсчеты, пропуски/подстановки, принятая граница конца, причины отказа и физи-

ческие наблюдения для каждого получателя. Счетчик отправленных АРМ байтов не является результатом воспроизведения.

15.20 Базовый вариант rtp-alert/5 - G.711 A-law, 8000 Гц, mono, 160 отсчетов на пакет: 20 мс, 50 пакетов/с, 64 кбит/с полезного звука. Дополнительный вариант GSM использует пакетизацию 20, 40 или 80 мс. A-law и mu-law различаются; mu-law в rtp-alert/5 не применяется. Статические payload type и базовые форматы кодеков определены RFC 3551; дополнительные частоты PCMA согласуются по таблице ниже. Собственный четырехбайтовый счетчик перед звуком не является заголовком RTP.

15.21 При базовых 8000 Гц буфер 120 мс соответствует 960 отсчетам: 960 байтам G.711 либо 1920 байтам после декодирования в 16-битный PCM. Это только звуковые данные; сетевые очереди, заголовки, защита, метаданные, стек и буфер аппаратного вывода считаются дополнительно.

15.22 Предел буфера микрофона в rtp-alert/5 равен 1000 мс, начальное накопление - 400 мс. Непрерывное отсутствие пригодного аудио 5000 мс вызывает остановку. Эти значения являются требованиями профиля; изменение чисел требует другого идентификатора медиапрофиля.

Транспорт и multicast

15.23 Локальное воспроизведение

Запись доставляется и проверяется до session.start. Аудиотрафик во время исполнения НЕ ТРЕБУЕТСЯ.

15.24 Речевой UDP unicast

rtp-alert/5: открытые звуковые байты по UDP; при защищенном управлении обязательна метка подлинности. Ключ передается командой audio.prepare внутри TLS. В plain метки нет. Повторная передача звука и скрытая смена режима запрещены.

15.25 Групповая доставка и UDP

Групповые профили group-audio/2 и group-audio-open/1 полностью определены в нормативном приложении Ж настоящего документа; в онлайн-версии — в разделе «Групповая передача речи». Они дополняют базовую одноадресную доставку и требуют поддержки групповой маршрутизации. Защищенный профиль проверяет подпись источника. Сам rtp-alert/5 остается одноадресным.

15.26 Общего ограничения числа УУ протокол не устанавливает. Каждая реализация объявляет проверенные квоты. Один поток для большой группы требует расчета числа соединений, ключей, пакетов и суммарной полосы источника; наличие сети Ethernet само по себе емкость не доказывает. Доверенный узел распределения является отдельной границей полномочий и учета результатов; он не превращает чужой поток в сквозное криптографическое доказательство источника.

Скорость сети и число получателей

15.27 Таблица ниже оценивает PCMA/8000/mono: 160 байт на 20 мс. К каждому звуковому пакету добавляются RTP 32 Б, заголовок OAU5 32 Б, метка 16 Б, UDP 8 Б, IPv4 20 Б и Ethernet 38 Б с преамбулой и межкадровым интервалом. ACK на звук отсутствует. Успешная оценка часов каждые 2 с требует трех датаграмм по 122, 130 и 122 Б, то есть 1496 бит/с суммы направлений. Повторы служебных пакетов и инкапсуляция учитываются дополнительно.

Таблица 16 - Скорость сети и число получателей

УУ	**Только звук, Мбит/с**	**Защищенные данные, Мбит/с**	**Данные и часы, Мбит/с**	**С резервом 30%, Мбит/с**
1	0,064	0,1224	0,1239	0,1611
10	0,64	1,224	1,239	1,6106
100	6,4	12,24	12,3896	16,1065
1 000	64	122,4	123,896	161,0648
4 000	256	489,6	495,584	644,2592
10 000	640	1 224	1 238,96	1 610,648

15.28 Формула данных с меткой: R=(P+32+32+16+8+20+38)*8*1000/T бит/с. С учетом часов сумма двух направлений R+1496. Один PCMA-получатель требует 122,4 кбит/с данных и 123,896 кбит/с с часами; 4000 получателей - 495,584 Мбит/с, с резервом 30% - 644,2592 Мбит/с. Направления полнодуплексного интерфейса проверяются отдельно. В plain отсутствуют 16 байт метки. VLAN, туннели и реальные потери учитываются по трассе.

Медленные каналы

15.29 G.711 A-law

Пакетизация: 20 мс / 160 Б

Звук, кбит/с: 64

Метка/UDP/IP, кбит/с: 107,2

Метка/UDP/Ethernet, кбит/с: 122,4

15.30 GSM 06.10

Пакетизация: 20 мс / 33 Б

Звук, кбит/с: 13,2

Метка/UDP/IP, кбит/с: 56,4

Метка/UDP/Ethernet, кбит/с: 71,6

15.31 GSM 06.10

Пакетизация: 40 мс / 66 Б

Звук, кбит/с: 13,2

Метка/UDP/IP, кбит/с: 34,8

Метка/UDP/Ethernet, кбит/с: 42,4

15.32 GSM 06.10

Пакетизация: 80 мс / 132 Б

Звук, кбит/с: 13,2

Метка/UDP/IP, кбит/с: 24

Метка/UDP/Ethernet, кбит/с: 27,8

15.33 Для суммарного канала 64 кбит/с расчетный вариант - GSM/80: данные 27,8 кбит/с, часы 1,496, сумма 29,296; с резервом 30% - 38,0848. Управлению выделяется не менее 12 кбит/с; остается 13,9152 кбит/с на дополнительные расходы. Фоновые загрузки приостанавливаются либо получают доказанно свободную полосу. Поддержка GSM/80 обязательна при заявлении живой речи на 64 кбит/с; общая поддержка stream этого не обещает.

15.34 GSM 06.10 допускает 20, 40 либо 80 мс: один, два либо четыре последовательных 33-байтовых кадра RFC 3551. PCMA допускает только 20 мс. Выбор фиксируется в audio.prepare до session.start; неподдержанный вариант дает unsupported. При недостаточной полосе с учетом меток подлинности и служебного обмена поток не запускается. Известное сообщение можно доставить целиком и воспроизвести локально; увеличение буфера не исправляет постоянно недостаточную скорость канала.

Задержка микрофона и буфер

15.35 Цель

Минимальная устойчивая задержка: накопление пакета + кодирование + доставка + проверка защиты + jitter-буфер + аппаратный вывод.

Каждая составляющая измеряется; размер буфера не равен всей сквозной задержке.

15.36 Начальная инженерная оценка

Для потоковой записи выбираются разрешенные initialMs/maxMs по измеренному джиттеру. Для микрофона действуют фиксированные 400/1000 мс. Параметры согласуются до session.start и не меняются посреди потока; измеряются задержки на фактическом канале, включая 64 кбит/с.

15.37 Адаптация

Буфер ограничен initialMs/maxMs выбранного rtp-alert/5. Рост задержки для компенсации потерь запрещен; просроченные отсчеты заменяются тишиной с учетом потерь, при достижении таймаута поток останавливается.

15.38 Восстановление потерь

Буфер компенсирует джиттер и перестановку. rtp-alert/5 не повторяет потерянный звук и не имеет FEC: пропуски маскируются тишиной, отражаются в счетчиках и результате. Запоздалые пакеты отбрасываются. Групповой профиль имеет отдельную восстановительную часть.

15.39 Предельное условие

Нулевые потери и минимальная задержка одновременно не гарантируются для произвольной сети. Приемка задает допустимые потери/джиттер и проверяет разборчивость, полноту, задержку и session.stop; выход за эту область не скрывается успешным результатом.

Согласование частоты одноадресной речи

15.40 Базовая частота rtp-alert/5 остается 8000 Гц. Для PCMA дополнительно допускаются 16000 и 32000 Гц: тот же алгоритм A-law применяется отдельно к каждому отсчету. Это расширение частоты дискретизации, а не новый закон сжатия. Дополнительные варианты используют динамические RTP payload type 96 и 97 по таблице; статический PT=8

сохраняет значение PCMA/8000. GSM остается только 8000 Гц. Все варианты монофонические.

Таблица 17 - Согласование частоты одноадресной речи

Кодек / частота	**PT**	**packetMs / N**	**Звук / RTP, байт**	**OAU5 plain / tls, байт**
PCMA / 8000	8	20 / 160	160 / 192	224 / 240
PCMA / 16000	96	20 / 320	320 / 352	384 / 400
PCMA / 32000	97	20 / 640	640 / 672	704 / 720
GSM / 8000	3	20, 40, 80 / 160, 320, 640	33, 66, 132 / 65, 98, 164	97, 130, 196 / 113, 146, 212

15.41 УУ объявляет необязательный атрибут sampleRates в записи mediaProfile с id=rtp-alert/5 и codec=PCMA. Значение является упорядоченным списком через одиночный пробел без повторов: 8000, 8000 16000, 8000 32000 либо 8000 16000 32000. Отсутствие атрибута означает только 8000. Это свойство существующего сочетания id/ security/codec/packetMs; отдельные дублирующие записи для частот не создаются. GSM и групповые профили этот атрибут не используют.

15.42 В audio.prepare атрибут stream.sampleRate необязателен; отсутствие означает 8000. Значение 16000 либо 32000 допускается только для PCMA/20 после объявления этой частоты выбранным mediaProfile. Значение вне 8000/16000/32000 дает invalidArgument, неподдержанное сочетание - unsupported. Старому получателю АРМ отправляет прежний запрос без sampleRate и поток 8000 Гц. Понижение частоты исходного файла выполняется до подготовки отдельной копии; нельзя послать более частые отсчеты в пакетах старого варианта или сменить кодек посреди потока.

15.43 N=packetMs*sampleRate/1000. Все totalSamples, sampleCount, validSamples, sourceSamples, concealedSamples, emittedSamples и RTP timestamp относятся к выбранной частоте. Миллисекунды определяются

как samples*1000/sampleRate. firstTimestamp и timestamp остаются uint32 с прежним правилом переполнения; packetIndex и sequence сохраняются. capturedAtMs, сроки свежести, START/STOP, часы и размеры буфера в миллисекундах не меняются. Полезный хвост END отделен от кодекового дополнения до N; после EOF выводится ровно подтвержденная граница.

15.44 Объем звука в памяти и полоса растут пропорционально частоте. При 120 мс PCMA требуется 960, 1920 либо 3840 байт кодированных отсчетов; PCM16 занимает вдвое больше. Для PCMA/16000 и PCMA/32000 защищенные данные с Ethernet занимают соответственно 186,4 и 314,4 кбит/с, без часов, управления и резерва. Предел пакета не разрешает IP-фрагментацию; MTU проверяется до запуска. Повышение частоты не разрешает увеличивать задержку или сроки остановки.

15.45 Частота согласуется для каждого одноадресного получателя отдельно. Групповые group-audio/2 и group-audio-open/1 и локальные объекты content сохраняют прежние форматы 8000 Гц. Расширение rtp-alert/5 не добавляет им частоты 16000/32000. Аппаратная поддержка дополнительной частоты включает настройку ЦАП, проверку аналогового тракта, своевременное обслуживание UDP и криптографии и измерение START/STOP. Одно объявление sampleRates не является доказательством акустического качества.

15.46 Приемка каждой объявленной частоты включает точные байты A-law, число и длительность отсчетов, неполный хвост, потерю и перестановку пакетов, plain и tls, STOP при поступающем звуке и новый поток 8000 после завершения. Неверный PT и несогласованная частота отбрасываются. Прежний APM без sampleRate и прежнее УУ без sampleRates проверяются отдельно; базовые пакеты PCMA/8000 и GSM остаются побайтно прежними.

Подлинность, возраст и очереди речевых пакетов

15.47 Отсоединение управляющего TCP само по себе не отзывает действующий lease. Уже связанный через BIND поток принятого сеанса продолжает прием до восстановления управления либо истечения lease, если сохраняются остальные условия допуска и свежести. В состоянии detached запрещено связывать новый поток или запускать подготовленный. Источник восстанавливает привязку control.resume к прежним bootId/controlEpoch и читает прежний sessionId; повтор session.start, новая выдача владения и перезапуск шкалы звука не допускаются как способ повторить чтение. UDP и чтение результата не продлевают lease. Отзыв, STOP и истечение lease действуют независимо от переподключения. Неустановленный результат остается неизвестным, а не успешным.

15.48 rtp-alert/5 передает одноадресную речь по UDP. Команды audio.prepare и session.start используют выбранный управляющий канал: TLS при tls, явно открытый TCP при plain. Звуковые байты не шифруются. При tls каждая датаграмма имеет метку HMAC-Стрибог-256, при plain метки нет; автоматическое переключение запрещено. Поле security в mediaProfile обозначает режим управления, а не шифрование UDP. Групповая речь использует group-audio/2 либо group-audio-open/1 и отдельные правила приложения Ж.

15.49 При tls источник создает новый случайный ключ key из 32 байт и передает его как 64 строчных hex в stream команды audio.prepare. При plain key запрещен. Ключ не включается в конфигурацию, журналы, диагностику и долговременное состояние; дедупликация хранит хеш нормализованного запроса. Ключ уничтожается при отмене, завершении, отзыве, истечении подготовки и reboot. Он не применяется для другого mediaId или команд управления. УУ связывает mediaId с bootId, controlEpoch, владельцем, ресурсами и точным маршрутом подготовки.

Таблица 18 - Подлинность, возраст и очереди речевых пакетов

Смещение / длина	**Заголовок UDP rtp-alert/5**
0 / 4	ASCII OAU5.
4 / 1	flags: бит 0 равен 1 только при tls; бит 1 задает направление: 0 от источника, 1 от УУ. Остальные биты нулевые. Режим обязан совпадать с подготовкой.
5 / 1	kind: 1 BIND, 2 READY, 3 CLOCK_REQUEST, 4 CLOCK_RESPONSE, 5 AUDIO, 6 CLOCK_ACK.
6 / 2	payloadBytes: uint16 BE, точная длина следующего тела, 0..672; предел AUDIO определяется согласованной частотой.

Окончание таблицы 18

Смещение / длина	**Заголовок UDP rtp-alert/5**
8 / 16	mediaId: 16 байт из 32 hex, выданных audio.prepare.
24 / 8	sequence: uint64 BE, отдельный счетчик каждого направления. Первый пакет 1; каждый новый пакет увеличивает счетчик, переполнение запрещено.
32 / payloadBytes	Открытое тело. AUDIO содержит один RTP-пакет; другие виды имеют фиксированные тела ниже.
32+payloadBytes / 16	Только при tls: первые 16 байт HMAC-Стрибог-256 над заголовком и открытым телом. При plain поле отсутствует.

15.50 Ключ метки K=HMAC-Стрибог-256(key, ASCII "OpenAlert rtp-alert/5 auth" || mediaId[16]); нулевой завершающий байт к строке не добавляется. Оба направления используют K, направление включено в проверяемый заголовок. Полная датаграмма содержит ровно 32+payloadBytes+16 байт при tls либо 32+payloadBytes при plain; максимум 720 либо 704 байта для PCMA/32000. Для PCMA/8000 сохраняются 240 либо 224 байта. IP-фрагментация, составные пакеты, XML и TCP-обрамление запрещены. MTU тракта обязан вмещать пакет с UDP/IP и применяемой инкапсуляцией.

15.51 Получатель сначала проверяет существование разрешенного mediaId, состояние подготовки, маршрут, длину, направление и метку; затем окно sequence из 64 пакетов относительно старшего принятого номера. Неверная метка не сдвигает окно и не закрывает действующий поток. Дальний корректный номер сдвигает окно; старые номера ниже окна отбрасываются. Повтор AUDIO не воспроизводится. Повтор

BIND или ответа часов может вызвать повтор служебного ответа, но не освежает часы, lease или срок подготовки. Изменение режима и ключа в пакете запрещено.

15.52 BIND: источник к УУ

Пустое тело. Первый допустимый пакет связывает UDP с живой подготовкой. УУ отвечает READY. Источник повторяет ту же датаграмму не чаще одного раза в 250 мс, не более 10000 мс. Повтор не создает нового разрешения и не вытесняет поток.

15.53 READY: УУ к источнику

Пустое тело; mediaId находится в заголовке. Ответ на повтор BIND ограничен одним в 200 мс. session.start остается отдельной командой владельца.

15.54 CLOCK_REQUEST: УУ к источнику

challenge:uint64 BE, 8 байт. Первый challenge равен 1; новые оценки используют возрастающие номера без переполнения. Одновременно ожидается один ответ.

15.55 CLOCK_RESPONSE: источник к УУ

challenge:uint64 BE || sourceMs:uint64 BE, 16 байт. sourceMs снимается перед первой отправкой ответа в монотонной шкале источника. Повтор сохраняет исходные байты и отметку.

15.56 CLOCK_ACK: УУ к источнику

challenge:uint64 BE, 8 байт. УУ подтвердило допустимую оценку часов. До первого ACK источник не отправляет AUDIO; потерянный ACK восстанавливается повтором ответа часов.

15.57 AUDIO: источник к УУ

Ровно один RTP-пакет 32..672 байта по настоящему разделу. Нет подтверждений и повторных передач звука; END согласуется с надежной командой audio.finish.

15.58 После BIND УУ немедленно запрашивает часы. Пока нет ответа, запрос повторяется не чаще одного раза в 250 мс с тем же

challenge и исходным u0; после превышения 1000 мс начинается новая попытка с новым challenge. При действующей оценке новая попытка начинается через 2000 мс. Источник повторяет неизменный CLOCK_RESPONSE каждые 250 мс до ACK или нового challenge. Старые и непрошенные ответы не обновляют оценку. Потеря служебной датаграммы не вызывает повторной передачи аудио.

15.59 Часы имеют дискретность не хуже 1 мс и относительное расхождение не более 500 ppm. u0 - монотонное время УУ перед первой отправкой запроса; u2 - время после получения и проверки ответа; s1 - sourceMs. При 0 <= u2-u0 <= 1000 мс смещение часов УУ относительно источника находится в [u0-s1, u2-s1]. Интервал расширяется с каждой стороны на D=1+ceil(500*(now-u0)/1000000) мс. Все разности вычисляются знаковым расширенным типом, без uint64 wrap. Оценка действует до u0+10000 мс; новый допустимый ответ заменяет ее. До первой оценки RTP не принимается; истечение prepareDeadlineMs дает prepareTimeout.

15.60 capturedAtMs в RTP - монотонное время первого отсчета в шкале sourceMs. Микрофон учитывает аппаратный буфер захвата; запись формируется в реальном темпе. Подменять отметку временем отправки задержанного пакета запрещено. Отметки не убывают; нарушение связи отсчетов и времени дает mediaClockFault. Верхняя оценка возраста A=now-capturedAtMs-(u0-s1-D), нижняя B=now-capturedAtMs-(u2-s1+D). Требуется -2 <= A <= 3000 мс; отрицательное B при допустимом A само по себе не является отказом. Проверка выполняется после метки и повторно для выдачи отсчетов с учетом очереди ЦАП. Устаревший звук не воспроизводится.

15.61 Просроченный пакет учитывается как latePackets; доказанный пропуск отражается как потеря и маскирование. Потеря допустимой оценки часов прекращает сеанс с mediaClockFault. Отсутствие пригодного аудио 5000 мс дает sourceUnavailable для микрофона либо mediaIncomplete для записи. После END оценка часов остается обяза-

тельной для хвоста, а новые ответы часов не продлевают срок его выдачи. session.stop немедленно отзывает прием и очищает еще не выданный звук, включая корректный END.

15.62 Предел свежести 3000 мс, срок оценки часов 10000 мс, отсутствие пригодного аудио 5000 мс, сроки подготовки, владения и предельной длительности имеют разные назначения. Ни один из них не является таймером остановки из-за уровня аудиоочереди. Успешный TCP/TLS handshake, heartbeat и ответы часов не заменяют поток звука, не продлевают lease и не подтверждают работу или выключение ОУ.

15.63 Источник ограничивает очередь пятьюдесятью звуковыми пакетами и возрастом первого отсчета 1000 мс, включая кодирование и сетевой стек. Датаграмма передается атомарно; при невозможности ограничить очередь источник прекращает поток и запрашивает session.stop через управление. Приемник хранит звук только в пределах initialMs/maxMs, обслуживает UDP ограниченными порциями и сохраняет сроки управления. Переподключение или поздний BIND не возобновляет завершенный либо отозванный mediaId. Новый сеанс требует новых key/mediaId, подготовки и session.start. Согласованный sourceSwitch вместо нового session.start использует audio.start в том же сеансе.

15.64 Потеря первого либо другого пакета начального накопления не блокирует весь поток. При наличии пригодного звука окно от позиции 0 до последнего принятого отсчета размером initialMs разрешает начать выдачу с маскированием дыр. Полный короткий поток начинает выдачу сразу; неполный короткий поток с известным концом ожидает initialMs от первого пригодного пакета, затем выдает доступные отсчеты с учетом потерь. Отсутствие пригодного звука не разрешает запуск. Потеря отражается в concealedSamples и предупреждении mediaPacketLoss, но сама по себе не запрещает завершение с succeeded/completed при полном проходе известной границы и подтвержденном off.

15.65 При предварительном декодировании приемник сохраняет ограниченное окно перестановки. Пропуск позади принятой границы после расходования этого окна маскируется без ожидания полного опустошения PCM. К каждому пропуску не добавляется пауза следующего обслуживания основным циклом; выдача продолжается по часам ЦАП. Еще неизвестный хвост не объявляется потерянным заранее. Маскированные позиции учитываются однократно, а поздние байты не воспроизводятся задним числом.

15.66 Микрофон непрерывно передает пакеты с согласованным packetMs, включая кодированную тишину. VAD/DTX запрещены. Таймаут 5000 мс относится к отсутствию пригодных пакетов, а не к амплитуде; ответ часов не заменяет звук.

15.67 Приемка проверяет открытые звуковые байты, изменение метки и тела, чужой ключ, mediaId, маршрут и направление, попытку смены режима, повторы, перестановку и потерю датаграмм. Отдельно проверяются потеря BIND/READY/ответа часов/ACK, RTT до 1000 мс, задержка звука и часов, предельный дрейф, отзыв, STOP и 10 секунд кодированной тишины. Подлинность, возраст, полнота воспроизведения и физическое выключение проверяются раздельно при заявленной нагрузке.

15.68 В режиме tls команды управления зашифрованы. UDP-звук остается открытым; метка каждого пакета защищает его подлинность и целостность, а номер и время ограничивают повторы и задержку. Один защищенный session.start без проверки медиапакетов этих гарантий не дает. В plain криптографической защиты нет; IP, порт, SSRC или mediaId ее не заменяют. Для заранее проверенной локальной записи сетевого аудио во время исполнения нет, однако доверие к загрузке, неизменность хранения и достоверность физической обратной связи остаются самостоятельными условиями.

Воспроизведение, тангента и остановка

15.69 Подготовка принята

Ресурсы, параметры и контекст зафиксированы; выход еще не считается воспроизводящим. Подготовка имеет конечный срок, а ready - конечный срок применимости.

15.70 Первый пригодный пакет

Учитывается только после проверки разрешения потока, привязки UDP-источника, номера и допустимого возраста пакета, а в режиме tls также его метки подлинности. Подтверждает прием данных, но не готовность всего буфера и не акустическое воспроизведение.

15.71 Буфер готов

Воспроизведение начинается по согласованному правилу. Его наблюдение отделено от факта выдачи отсчетов программой; отсутствующая обратная связь явно ограничивает доказательство.

15.72 Потеря или underrun

Пропуски маскируются и учитываются без перезапуска контента. Потерянные слова не считаются доставленными. Полный проход известной границы с подтвержденным off допускает completion=complete и succeeded/completed с предупреждением mediaPacketLoss.

15.73 Штатный EOF потоковой записи

Фиксируется конечное число действительных отсчетов; буфер доигрывается только до этой границы и в пределах срока дренирования. Завершение этого прохода и доставка всех исходных отсчетов проверяются раздельно; маскирование утраченного хвоста не скрывает потери.

15.74 Отпускание тангенты / отзыв источника

Отпускание передает audio.finish: ограниченный хвост до sampleCount доигрывается. Принудительный отзыв выполняется session.stop/cancel и отбрасывает хвост. Завершение mediaId не освобождает владение УУ. После полного off новое нажатие создает новую

подготовку и сеанс; при согласованном sourceSwitch подготовка и audio.start продолжают прежний сеанс.

15.75 session.stop/session.cancel, истечение срока или отзыв сеанса

Прекращаются прием и будущая выдача звука, очищается тракт и подтверждается выключение. Режим дренирования штатного EOF не используется для принудительной отмены. Поздние команды и пакеты старого потока не затрагивают новый.

15.76 Reboot или возврат сети

Прежнее разрешение не восстанавливается автоматически. Новое включение требует актуального контекста и повторной подготовки; запоздалое аудио отбрасывается.

15.77 Тангента управляется владельцем сеанса. До session.start(mode=stream) обязательны audio.prepare(source=microphone), подтвержденный prepared и готовая привязка медиаканала. Подготовка ДОПУСКАЕТСЯ заранее в пределах ее срока, без включения выходов и захвата речи. Если нажатие поступило во время подготовки или ее обновления, АРМ может сохранить одно действующее намерение до готовности; повторное нажатие не требуется и не создает второй START. Отмена, окончание сеанса и потеря права источника снимают ожидающее намерение. При согласованном startup=outputReady захват и передача начинаются только после outputReady=true из audio.state данного mediaId/sessionId. До этого отпускание тангенты выполняет session.stop/cancel, без захвата и накопления речи; после начала передачи штатное отпускание передает audio.finish. При согласованном sourceSwitch конец источника оставляет речевой режим для следующей тангенты; ручная остановка файла использует audio.pause. Новый audio.prepare и audio.start продолжают тот же сеанс, а полный отбой выполняется session.stop/cancel. Физический источник может быть внешним устройством АРМ, но сетевым отправи-

телем и носителем полномочий остается владелец. Передача слова другому участнику выполняется после конца и off прежнего сеанса. Потерянный prepared или accepted не разрешает начинать новый запуск; применяется чтение либо адресная отмена по общей модели.

15.78 Подготовка истекает через 60000 мс от создания, запуск ограничен acceptBeforeMs и prepareTimeoutMs, поток - maxDurationMs и таймаутом отсутствия данных 5000 мс. Повторы не продлевают эти сроки. session.stop отзывает медиаразрешение и очищает буфер в stopBeginMs.

15.79 Потеря управляющего соединения, истечение lease и потеря аудио являются разными событиями. Для локальной конечной записи и программы применяется зафиксированная stop/finish. Для stream всегда применяется stop при истечении lease; хвост неполученной записи не обещается к автономному воспроизведению. Гарантия окончания сообщения при потере АРМ требует локальной записи в mode=recording или program.

15.80 Для микрофона потеря разрешения запрещает последующий звук; конкретный срок обнаружения потери источника задается профилем. Автоматическая подстановка резервной записи ДОПУСКАЕТСЯ только как заранее разрешенное действие с точной версией контента и отдельным результатом, а не как продолжение якобы полученной речи.

15.81 При переходе "сирена -> речь" подтверждается завершение предыдущего воздействия, сохраняется происхождение запуска, подготавливается новый тракт и отзываются прежние отложенные команды. Частичная готовность группы обрабатывается по заранее выбранной политике с результатом каждого получателя; готовность части не выдается за готовность всей группы.

15.82 Дренирование медиабуфера и draining владения - разные состояния разных автоматов, их нельзя отождествлять.

Приемочные испытания речи

15.83 M01: контент

Обрыв загрузки, подмена версии и неверный хеш не приводят к воспроизведению другого или неполного объекта. Производственное размещение и сетевое обновление проверяются отдельно.

15.84 M02: подготовка

Тракт готов, но аудио нет: истекает срок первого пакета, успешное воспроизведение не объявляется. Неподготовленный получатель не открывает выход по первым медиабайтам; UDP не принимается.

15.85 M03: порядок и потери

Дубликаты, перестановки, пропуск середины и хвоста, поздний пакет и переполнение буфера не создают повторного/запоздалого звучания или ложного полного результата.

15.86 M04: точное окончание

Неполный END-пакет несет validSamples. Задержка audio.finish на 200 мс, его опережение, потеря и повтор не меняют границу. Короткий поток меньше initialMs и пустой поток имеют определенные исходы; потерянный хвост с известной границей дает счетчики потерь и mediaPacketLoss. Успешное завершение прохода не означает доставку утраченного звука.

15.87 M05: отмена

session.stop во время подготовки, воспроизведения и дренирования прекращает звук в измеренном бюджете; пакеты из сетевой и аппаратной очередей не возобновляют его.

15.88 M06: тангента

Отпускание, потеря источника, два претендента, поздняя выдача права и повтор старого поколения не разрешают новую передачу.

15.89 M07: независимые каналы

ДОЛЖНЫ проверяться отдельно: потеря управления при живом аудио, потеря аудио при живом управлении, полный обрыв и возврат связи. finish не объявляет доставленным неполученный хвост.

15.90 M08: этапы и группа

Поздний session.stop сирены не выключает новый речевой этап; отказ части приемников виден по каждому устройству и не заменяется общим успехом.

15.91 M09: защита

В tls отвергаются неверная метка, чужой principal, старый mediaId и UDP-инъекция. В plain проверяется только объявленная привязка и явно показывается отсутствие криптографической защиты. Переход tls в plain после ошибки запрещен; режимы проверяются отдельно.

15.92 M10: ресурсный бюджет

Сквозная задержка, буфер, время криптографии, память, число сокетов и задержка session.stop не превышают установленные пределы при одновременном управлении ОУ, опросе и нагрузке объявленного числа получателей.

15.93 Полное сообщение означает подтвержденное прохождение требуемого интервала без неучтенных пропусков на заявленном уровне наблюдения. Маскирование потери или выдача тишины не делает недоставленные слова доставленными. Цифровая полнота, работа усилителя и акустическое свидетельство сообщаются раздельно; слышимость и понимание населением из сетевых квитанций не выводятся. Критерии речевого оповещения охватывают управление, выбранный медиарежим, полноту отсчетов и физические выходы.

Согласованное время и групповой запуск

Синхронный групповой запуск требует качества часов, допуска по времени, правил опоздания и результата частичной готовности. Профиль 0.21 не предостав-

ляет синхронный запуск по UTC; такой запрос отклоняется с unsupported и не заменяется немедленным.

15.94 Профиль согласованного времени фиксирует общую шкалу, источник синхронизации, максимальную погрешность и время удержания точности после потери источника. Название NTP или PTP само по себе не задает достигнутую точность. В межузловом поручении исходный абсолютный срок сохраняется: посредник НЕ ДОЛЖЕН продлевать его назначением нового полного TTL. При переводе на локальный монотонный таймер используется консервативный остаток до срока с учетом верхней границы ошибки часов; при недостоверной границе срок не считается проверенным. Монотонный maxDurationMs принятого исполнения считается отдельно от срока приема и коррекции настенных часов.

Смена источника в одном речевом сеансе

15.95 Необязательное расширение sourceSwitch=true объявляется только для индивидуального mediaProfile rtp-alert/5 вместе со startup=outputReady. АРМ выбирает его атрибутом holdOutput="true" в первоначальном audio.prepare. Без объявления или без выбора действует прежнее завершение потока с отбоем. Групповые профили расширение не используют. Параметры кодека и передача 8 кГц остаются прежними.

15.96 После полного EOF и выдачи последнего отсчета на выход согласованный источник переходит в audio.state state=paused, sourceResult=completed. Команда audio.pause со stream mediaId/sessionId переводит источник в paused, sourceResult=operatorStop, немедленно очищает его сжатую очередь и еще не выданный PCM. Обе команды сохраняют controlEpoch, sessionId, startSeq, подтверждение речевого режима и первоначальный предельный срок исполнения. В paused выход остается в речевом режиме и получает тишину; это не подтвержде-

ние off. Потери завершенного источника сохраняются как mediaPacketLoss в результате сеанса.

15.97 Следующий audio.prepare включает holdOutput=true и sessionId текущего сеанса. Он допустим только после paused в running, от того же владельца, для тех же ресурсов. Создается новый mediaId, свежий ключ при tls, отдельные счетчики, декодер, буфер и синхронизация; разрешена смена согласованных кодека и частоты, в том числе запись 16/32 кГц на микрофон 8 кГц. Состояние нового источника prepared уже содержит sessionId. После BIND/READY команда audio.start со stream mediaId/sessionId разрешает новый источник без повторной аппаратной команды включения. Ответ status=ok, started mediaId/sessionId/startSeq; audio.pause отвечает аналогичным paused. Захват и отправка AUDIO начинаются только после outputReady=true данного нового mediaId. Подготовка и ее истечение не включают и не выключают ОУ.

15.98 audio.pause и audio.start являются обычными изменяющими командами владельца с bootId, controlEpoch и cmdSeq. Дубликат возвращает сохраненный ответ без повторного действия. Старый mediaId, ключ и UDP-маршрут после паузы не принимают звук; запоздалый EOF или команда старого источника не меняют новый. После замены прежний audio.state может вернуть notFound: APM сохраняет его итог до замены. Срок новой подготовки равен 60000 мс, audio.start после него отклоняется. Подготовка может обновляться новым mediaId, пока источник не запущен; это не продлевает maxDurationMs сеанса. Все запросы сохраняют предел 1024 байта и 12 атрибутов на элемент.

15.99 Таймеры отсутствия звука и синхронизации источника не применяются только к намеренной паузе и новой неиспользованной подготовке. Срок владения, первоначальный maxDurationMs, свежесть аппаратного состояния, защита оборудования и контроль физического тактирования продолжают действовать. Потеря управления, reboot, неисправность, session.stop/cancel и аварийный сброс завершают все

исполнение с обычным подтверждением off. Нельзя автоматически переиздавать session.start после сомнительного ответа на переключение. Пауза до подтверждения первого включения не создает доказательство running: исходный срок подготовки и свежие аппаратные подтверждения обязательны.

Форматы речевого профиля rtp-alert/5

15.100 Для ОУ с предварительной аппаратной подготовкой rtp-alert/5 поддерживает согласованный порядок startup=outputReady. УУ объявляет startup="outputReady" в соответствующем mediaProfile; АРМ выбирает его одноименным атрибутом stream команды audio.prepare только при таком объявлении. Отсутствие атрибута сохраняет прежний порядок; иное или неподдержанное значение получает unsupported. Групповые профили этот атрибут не используют. Подготовка audio.prepare по-прежнему не включает выходы.

15.101 При выбранном startup=outputReady перед session.start требуется завершить UDP BIND/READY. Принятый session.start немедленно инициирует подготовку речевого выхода без ожидания AUDIO. Ответ audio.state обязательно содержит outputReady=true|false, включая prepared и finished. true разрешено только в starting/running после свежего достоверного наблюдения требуемого речевого режима, полученного после отправки аппаратной команды; промежуточный режим подготовки и отправленная команда не подтверждают готовность. В остальных состояниях передается false. PCM до подтверждения речевого режима запрещен. outputReady не заменяет критерий running, подтверждение исполнения или акустический контроль.

15.102 АРМ до outputReady=true поддерживает привязку, синхронизацию часов и lease, но не захватывает микрофон, не считывает исходную речь с продвижением времени и не отправляет AUDIO. После подтверждения начинает источник с первого отсчета. Проверка выполняет-

ся по событию готовности либо чтением audio.state с интервалом не более 100 мс; фиксированная задержка вместо проверки запрещена. При тайм-ауте подготовки, finished или отмене источник не запускается, выполняются адресная остановка и чтение результата. Срок подготовки из паспорта ОУ, границы свежести источника и maxMs не увеличиваются. Таймер накопления начального буфера и установления частоты отсчитывается от первого принятого звукового пакета, а не от времени accepted: ожидание аппаратной готовности не расходует начальный буфер.

15.103 mediaId включается в каждую UDP-датаграмму и определяет контекст подготовки. После session.stop или reboot разрешение отзывается; старый пакет не принимается в новом mediaId. Привязка и часы используют бинарные датаграммы rtp-alert/5 без media.hello.

15.104 Файл локального контента содержит только байты кодека, без WAV, RTP и иных контейнеров: PCMA - один байт A-law на отсчет, sampleCount=bytes; GSM - последовательность 33-байтовых кадров RFC 3551, bytes кратно 33 и 160*(bytes/33-1) < sampleCount <= 160*bytes/33. Последний GSM-кадр дополнен тишиной до 160 отсчетов; лишние отсчеты не воспроизводятся. Частота 8000 Гц, один канал. Хеш вычисляется над всеми передаваемыми байтами файла, включая дополненный последний кадр; длительность равна sampleCount/8000 секунд.

15.105 Потоковый профиль rtp-alert/5 использует RTP версии 2 с обязательным расширением, описанным ниже, без CSRC и padding. PCMA: payload type 8/96/97 для 8000/16000/32000 Гц соответственно, один канал, 20 мс, 160/320/640 байт отсчетов. Дополнительные частоты требуют объявления sampleRates и согласования sampleRate; без них остается 8000 Гц. GSM: payload type 3, 8000 Гц, один канал, 20, 40 или 80 мс, соответственно 33, 66 или 132 байта в формате RFC 3551. Изменение кодека, пакетизации, SSRC или маршрута требует нового mediaId; во время передачи изменение запрещено. Согласованный sourceSwitch

допускает замену в паузе того же сеанса по отдельным правилам. Профиль объявляет поддержку каждого варианта отдельно.

15.106 audio.prepare / stream

Изменяющая команда владельца с обычным cmdSeq. stream: source=recording|microphone, codec=PCMA|GSM, packetMs=20|40|80, sampleRate=8000|16000|32000 (необязателен, по умолчанию 8000), ssrc=8 строчных hex, firstSeq=0..65535, firstTimestamp=0..4294967295; key=64 строчных hex обязателен только при tls и запрещен при plain; для recording обязателен totalSamples, для microphone запрещен. PCMA допускает только 20 мс; GSM только 8000 Гц. Внутри stream: resources из 1..4 resource id/bindingGeneration; один route; один buffer.

15.107 route

UDP и unicast заданы профилем, transport/delivery в route отсутствуют; sourceAddress/destinationAddress - полные IPv4; sourcePort/destinationPort=1..65535; ttl=1..255. Маршрут проверяется по разрешенной конфигурации. Подлинность при tls доказывается меткой, а не IP-адресом. Для multicast применяется отдельный профиль.

15.108 buffer

initialMs=40|60|80|200|400; maxMs=80|120|160|400|1000, maxMs >= initialMs. Для microphone: initialMs=400, maxMs=1000. Время включает только буфер после проверки пакета. Отсутствие этих полей запрещено.

15.109 Подготовленный ответ

status=ok, prepared mediaId (32 hex), expiresAtMs, configRevision, capabilityRevision. Подготовка не включает выходы; занимает ограниченный буфер на 60000 мс, не продлеваемых повтором. mediaId связан с владельцем, эпохой, маршрутом и точными ресурсами. Один владелец имеет не более одного неиспользованного mediaId на УУ.

15.110 session.start mode=stream

Обычные поля session.start и resources; дочерний stream содержит только mediaId. УУ проверяет его срок, эпоху, ресурсы и ревизии. После

accepted mediaId не применяется для нового session.start; его замена в том же сеансе допускается только согласованным sourceSwitch. Для stream onControlLoss ДОЛЖНО быть stop.

15.111 audio.state

Чтение: selection mediaId. Ответ audio с mediaId и sessionId после принятого session.start, state=prepared|buffering|playing|draining|paused|finished, receivedPackets, lostPackets, latePackets, duplicatePackets, emittedSamples, bufferedMs и lastPacketAtMs. Все счетчики uint64; счетчики отсутствовавших событий равны 0; дополнительно clock по правилам частоты ниже. Чтение не продлевает lease.

15.112 audio.finish

Изменяющая команда владельца: stream mediaId, lastPacketIndex и sampleCount. Индексы расширенные, начиная с 0; sampleCount - полное число отсчетов источника. Ответ status=ok, eof mediaId/lastPacketIndex/sampleCount; несовпадающий повтор дает conflict. EOF согласуется с END и запрещает новые индексы выше lastPacketIndex; после finished совпадающая граница подтверждается без изменения результата. В состоянии prepared до session.start команда отклоняется с unsafeState.

15.113 Отмена

session.stop/cancel останавливает связанное исполнение и отзывает mediaId; поступающие пакеты больше не декодируются и не выводятся. audio.start разрешен только для следующего источника согласованного sourceSwitch; команды audio.stop нет. Непринятая подготовка отменяется audio.discard с prepared mediaId; ответ status=ok, discarded mediaId. Активный mediaId через discard не отзывается: unsafeState.

15.114 packetIndex передается явно в расширении rtp-alert/5 и проверяется вместе с sequence/timestamp по формулам ниже. Индексы не превышают 2^31-1; окно перестановки 64 пакета относительно старшего принятого индекса. Дубликаты и просроченные пакеты не воспроизводятся. Кодековая длина фиксирована; validSamples последнего

пакета отделяет полезный хвост от дополнения тишиной до декодирования на выход.

15.115 Источник открывает UDP-сокет на согласованном sourceAddress/sourcePort и передает BIND на destinationAddress/destinationPort. После READY и подтверждения часов можно отправлять RTP по правилам ниже: 192/352/672 байта для PCMA при 8000/16000/32000 Гц и 65/98/164 байта для GSM/20/40/80. Подготовка и привязка не включают выходы; первоначально требуется session.start, для следующего источника согласованного sourceSwitch требуется audio.start. Медиа и служебные датаграммы не продлевают управляющий lease. Неверный пакет отбрасывается без вытеснения разрешенного источника.

15.116 После accepted и, при startup=outputReady, подтверждения выхода поток накапливается до initialMs, кроме полного короткого потока по правилу ниже. Истечение prepareTimeoutMs до running дает prepareTimeout. Буфер и обработка потерь подчиняются правилам конечной границы и частоты ниже; для rtp-alert/5 FEC, повторная передача RTP и автоматическая смена кодека запрещены. Принудительный session.stop удаляет все еще не выданные отсчеты, включая корректный END.

15.117 Граница конца передается в END-пакете и дублируется audio.finish. Ее точная проверка, порядок при запаздывании управляющего сообщения, пустой и короткий поток определены ниже. Ни одно ожидание EOF не продлевает prepareTimeoutMs/maxDurationMs. После полного off повторное нажатие тангенты создает новый mediaId и новый сеанс; согласованный sourceSwitch создает новый mediaId и продолжает прежний сеанс через audio.start. Старый mediaId не возобновляет звук.

15.118 Сетевая загрузка локального контента использует content.begin/commit/status/cancel с административным правом и обычным cmdSeq у изменений; данные передаются bulk/1. begin содержит

content id/revision/codec/bytes/sampleCount/hashAlgorithm/hash и отдельный transfer profile/blockBytes/windowBlocks/windowRoundTripMaxMs. codec=PCMA|GSM, bytes:1..33554432. Ответ upload id/receivedBytes/remainingBudgetMs; новая транзакция имеет receivedBytes=0, повтор возвращает сохраненный префикс и прежний ID. commit содержит upload id и ДОПУСКАЕТСЯ после полного приема; проверяет длину, формат, sampleCount и Стрибог-256, затем атомарно публикует неизменный contentId/revision. Синхронный ответ content id/revision/state=committed либо accepted с operation.id=upload.id и последующим content.status. status является чтением selection uploadId и сообщает состояния и поля bulk/1 в разделе 16. cancel с upload id допустим до атомарного commit, включая verifying, ответ upload id/state=cancelled. Действующая ревизия не стирается при незавершенной загрузке, совпадающие id/revision с другими байтами дают conflict. Заводское размещение создает тот же манифест и результат проверки байтов.

Эталонные трассы окончания и синхронизации

15.119 late-eof

Входные условия: PCMA: полный пакет index=0, END index=1 с validSamples=40; audio.finish приходит через 200 мс после END.

Обязательный исход: sourceSamples=200, concealedSamples=0; 120 отсчетов дополнения не выдаются как звук источника; поздний finish подтверждает границу 200.

15.120 full-boundary

Входные условия: Микрофон отправил полный пакет index=0 без END, затем остановился.

Обязательный исход: END index=1, validSamples=0, нагрузка 0; sourceSamples=160, второго звукового пакета нет.

15.121 short-stream

Входные условия: END index=0, validSamples=40 приходит до накопления initialMs=60.

Обязательный исход: Начать выдачу после получения полного короткого потока, потребить 40 отсчетов источника; не ждать 60 мс.

15.122 missing-final

Входные условия: Микрофон уже running: полные пакеты index=0..2; END index=3 с validSamples=40 утрачен; получен надежный EOF sampleCount=520.

Обязательный исход: sourceSamples=520, concealedSamples=40; при confirmed исполнения и off в срок succeeded/completed с fault mediaPacketLoss, effect=degraded, severity=warning. Потеря не скрыта и не объявлена доставленной.

15.123 clock-drift

Входные условия: Потерь нет, джиттер доставки до 20 мс; дрейф +200 ppm в течение 600 с, затем отдельные трассы -200 ppm и изменения 1 ppm/с. Повтор при разных фазах пакетного пополнения аппаратной очереди.

Обязательный исход: Нет удаления/повтора пакетов, underrun, переполнения и ложного mediaClockFault; sourceSamples отделен от счетчика ЦАП. Заполнение включает аппаратную очередь и остается в пределах maxMs; коррекция ограничена ±500 ppm. Постоянное смещение заполнения не считается расхождением частот.

15.124 eof-after-payload-retirement

Входные условия: Выданы все действительные отсчеты короткого последнего пакета либо принят пустой END после полных пакетов. Память полезной нагрузки освобождена, затем приходит совпадающий audio.finish; отдельно противоречивый sampleCount.

Обязательный исход: Совпадающий конец подтверждается без лишних отсчетов, потерь и mediaIncomplete. Сохраненные validSamples

и граница не заменяются полной длиной пакета. Противоречивый конец отклоняется; завершенная история не переписывается.

Конечная граница, короткий поток и частота воспроизведения

15.125 RTP

V=2, P=0, X=1, CC=0. Обычный заголовок 12 байт. За ним 16-битный идентификатор 0x534f и 16-битная длина 4 (четыре 32-битных слова без четырехбайтового заголовка расширения). Все многобайтовые целые big-endian.

15.126 16 байт данных расширения

packetIndex:uint32, validSamples:uint16, flags:uint16, capturedAtMs:uint64. Единственный флаг END=0x0001; остальные биты нулевые. Полный заголовок RTP с расширением - 32 байта.

15.127 Обычный пакет

END=0, validSamples=N=packetMs*sampleRate/1000, при отсутствии sampleRate используется 8000. Полезная нагрузка: PCMA 160, 320 либо 640 байт для 8000, 16000 либо 32000 Гц; GSM 33, 66 или 132 байта. Индекс начинается с 0; sequence=(firstSeq+packetIndex) mod 65536, timestamp=(firstTimestamp+N*packetIndex) mod 2^32. marker=1 только при packetIndex=0.

15.128 Последний пакет

END=1, 1<=validSamples<=N. Нагрузка той же кодековой длины с дополнением тишиной, но декодер передает только validSamples. sampleCount=packetIndex*N+validSamples. Этот пакет задает границу до выдачи собственного хвоста независимо от доставки audio.finish.

15.129 Маркер границы

END=1, validSamples=0 и длина нагрузки 0 допустимы у микрофона после последнего уже отправленного полного пакета: packetIndex равен следующему индексу, sampleCount=packetIndex*N. Маркер не является звуковым пакетом и не увеличивает lostPackets при его утрате. При

packetIndex=0 поток пуст: выходы не включаются, итог cancelled/ operatorStop с подтвержденным off.

15.130 Для source=recording audio.prepare содержит обязательный totalSamples:uint64>0; для microphone это поле запрещено. У записи заранее известны lastPacketIndex=floor((totalSamples-1)/N) и допустимые отсчеты хвоста; END и audio.finish ДОЛЖНЫ совпасть с этой границей. Для микрофона конец определяется END либо заранее поступившим audio.finish. Если полный последний пакет уже отправлен без END, источник передает маркер границы с нулем отсчетов; повторная передача прежнего индекса с измененными флагами запрещена. lastPacketIndex в audio.finish обозначает индекс END, включая такой маркер. При sampleCount=0 индекс равен 0; иначе допустимы обычный хвост с (lastPacketIndex*N)<sampleCount<=(lastPacketIndex+1)*N либо маркер sampleCount=lastPacketIndex*N. У записи обязателен END в последнем звуковом пакете, отдельный нулевой маркер не используется. Два разных конца дают failed/mediaIncomplete до завершения; после finished противоречивое сообщение дает conflict и не переписывает историю. audio.finish повторяет lastPacketIndex/sampleCount, подтверждается eof; доставка после воспроизведения корректного END допустима и возвращает ту же границу в течение resultRetentionMs, даже если сеанс уже finished. Отсутствие audio.finish при полученном корректном END не является потерей звука.

15.131 УУ никогда не выдает дополнение последнего кодекового кадра как отсчеты источника. Короткий непустой поток меньше initialMs начинает выдачу, как только получены конечная граница и все его пакеты; ожидание полного стартового буфера в этом случае запрещено. При потерях начальное окно считается от позиции 0 до последнего принятого отсчета, включая доказанные пропуски. Неполный короткий поток с известным концом и хотя бы одним пригодным звуковым пакетом начинает выдачу через initialMs от первого пригодного пакета с учетом маски-

рования потерь. Отсутствие пригодного звука к prepareDeadlineMs дает prepareTimeout. В остальных случаях старт при накоплении окна не менее initialMs; округление вверх до целого кодекового пакета ДОПУСКАЕТСЯ только в пределах maxMs. Пустой поток не требует running и не считается ошибкой подготовки.

15.132 Освобождение памяти уже выданного пакета не изменяет принятую границу его действительных отсчетов. До завершения проверки конца УУ сохраняет сведения, необходимые для проверки sampleCount, validSamples и повторов. Поздний audio.finish с ранее принятой границей, в том числе после полного вывода последнего пакета или пустого маркера, не создает дополнительных отсчетов, потерь или mediaIncomplete. Противоречивый конец по-прежнему отклоняется; удаление полезной нагрузки не разрешает считать короткий хвост полным кодековым пакетом.

15.133 Потерянная позиция учитывается как lostPackets и замена тишиной только если ее принадлежность источнику доказана totalSamples, EOF либо наличием более позднего пакета. За еще неизвестным концом УУ может выдавать аппаратную тишину, но не начисляет фиктивные отсчеты источника и потери. При появлении более позднего пакета пропущенные позиции до него классифицируются один раз; поздние байты не воспроизводятся задним числом. Потери UDP сами по себе не останавливают поток. При завершении или остановке с concealedSamples>0 результат содержит fault code=mediaPacketLoss, effect=degraded, severity=warning. При потерянном END надежный EOF задает диапазон; его утраченные позиции маскируются, а полный проход до этой границы с подтвержденным выключением допускает succeeded/completed с предупреждением. Это не подтверждает доставку утраченного звука. Отсутствие пригодных пакетов 5000 мс без полного конца дает mediaIncomplete для записи и sourceUnavailable для микрофона. При неизвестном выключении итог unknown/stopUnconfirmed.

Все ожидания ограничены prepareTimeoutMs/maxDurationMs и не задерживают session.stop.

15.134 RTP timestamp описывает шкалу источника; часы ЦАП не считаются тождественными ей. Приемник ДОЛЖЕН поддерживать непрерывное воспроизведение при постоянном относительном расхождении частот до ±200 ppm, а также плавном изменении до 1 ppm/c внутри этого диапазона. Используется подстройка частоты ЦАП или непрерывный ресемплинг; коррекция ограничена ±500 ppm. Сброс timestamp, периодическое удаление целого пакета и повтор уже потребленного пакета для регулировки буфера запрещены. Алгоритм оценки не нормируется, но результат обязан удовлетворять указанным границам и испытательным трассам.

15.135 В отсутствие потерь, при джиттере доставки до 20 мс и указанном дрейфе УУ удерживает заполнение без underrun и без превышения maxMs. В оценку заполнения входят декодированные, но еще не выведенные отсчеты аппаратной очереди. Коррекция по среднему заполнению ограничена ±500 ppm. Изменение уровня очереди, насыщение коррекции и временный underrun сами по себе не являются доказательством расхождения частот и не запускают таймер остановки. После паузы обслуживания или потери пакета выдача продолжается с плавной коррекцией и учетом утраченных позиций. Переполнение установленной емкости либо нарушение допустимой связи времени и отсчетов дает failed/mediaClockFault; старые данные не вытесняются молча. Проверка свежести, срока допустимой оценки часов и работоспособности кодека сохраняется. Изменение initialMs/maxMs во время сеанса запрещено; бесконечный рост задержки не допускается.

15.136 audio.state сохраняет emittedSamples как число отсчетов, выданных ЦАП, включая ресемплирование и аппаратную тишину до off. Дополнительный дочерний clock содержит sourceSamples, concealedSamples, rateCorrectionPpm и state=locking|locked|outOfRange.

sourceSamples - число однократно потребленных позиций источника внутри известной границы, включая позиции замены; concealedSamples - сколько из них заменены из-за утраты, без добавленных кодековых отсчетов и тишины за неизвестным концом. rateCorrectionPpm - int от -500 до 500. При succeeded/completed конечного потока sourceSamples=sampleCount. concealedSamples=0 означает отсутствие маскирования потерь; при concealedSamples>0 обязательно предупреждение mediaPacketLoss, которое клиент показывает отдельно от отказа оборудования. emittedSamples может отличаться из-за коррекции часов. При ресемплинге длина считается в шкале источника, полнота прохода не требует равенства счетчикам ЦАП и не означает отсутствие потерь.

15.137 Счетчики пакетов, отсчетов, повторов, поздних пакетов и потерь обязательны в прежнем назначении. Запрещен только вывод об отказе частоты из длительного смещения уровня очереди или насыщения регулятора без независимого доказательства нарушения. clock.state описывает состояние алгоритма приемника: locked само по себе не является измерением точности аппаратного генератора и не подтверждает слышимость. Сравнение накопленных отсчетов с временем обслуживания программного цикла не заменяет измерение тактирования ЦАП.

Изоляция отказов подключения медиаканала

15.138 До успешного BIND датаграммы не связаны с подготовкой. УУ проверяет разрешенный mediaId, текущую эпоху и владельца, точный маршрут и подлинность пакета. Неверный пакет отбрасывается без очистки звука, изменения результата или отзыва действующего mediaId. Знание mediaId не доказывает полномочия источника.

15.139 BIND атомарно связывается с подготовкой. Его корректный повтор только восстанавливает служебный ответ. Новый источник, маршрут или mediaId не наследует полномочия действующего сеанса. STOP, отзыв, смена контекста и таймаут завершают прием; запоздалые

пакеты не возобновляют звук. Ошибка аутентификации не останавливает разрешенный поток.

15.140 Штатный session.stop/cancel отзывает поток и прекращает воспроизведение без требования передать оставшуюся часть файла, END или audio.finish. Отсчеты после принятой границы остановки и сброшенный по ней буфер не учитываются как потерянный звук. Для незавершенного файла completion=interrupted; при подтвержденном начале на всех ресурсах и off в срок ДОПУСКАЕТСЯ succeeded/operatorStop. Уже зафиксированные до остановки mediaIncomplete, sourceUnavailable, mediaClockFault и аппаратные отказы сохраняются и запрещают успех. Потери до границы остановки сохраняются в счетчиках и предупреждении mediaPacketLoss; они сами по себе не запрещают succeeded/operatorStop.

16 Конфигурация, контент и обновление

16.1 Изменения проходят подготовку, проверку и явное применение. config.stage не меняет действующую конфигурацию; config.commit атомарно фиксирует и применяет проверенные локальные правила получателя; configRevision и appliedRevision становятся равны. Команды физической перенастройки ОУ и перезапуск в этот пакет не входят. Несколько получателей применяют пакет независимо и имеют отдельные результаты. Значение onControlLoss хранится на конкретном приборе и не переопределяется произвольным session.start. После применения выполняется чтение фактической конфигурации.

16.2 Сеть меняется с подтверждением через новый путь и возвратом к прежним параметрам при истечении срока. Потеря ответа commit требует чтения результата по идентичности операции, а не слепого повторного применения. Секреты не возвращаются обычным config.get.

Обновление и восстановление

16.3 Прошивка принимается малыми блоками с проверкой совместимости, целостности и подлинности образа. Загрузка, проверка, активация и результат загрузчика различаются. В активном оповещении недопустимы перезапуск, смена профиля ОУ и изменение задействованного контента. Сервисная передача НЕ ДОЛЖНА занимать неограниченную очередь перед session.stop.

16.4 Конфигурация и контент фиксируются атомарно: действующая копия сохраняется до проверки и commit-маркера новой; после потери питания выбирается последняя целая зафиксированная ревизия. Допустимый откат выполняется как новая ревизия уполномоченного автора. Загрузчик сохраняет исполнимый образ либо независимый режим приема образа при обрыве любой операции записи. Если такая гарантия не обеспечивается аппаратурой, удаленная активация прошивки ДОЛЖНА быть запрещена.

16.5 Изменяющие сервисные команды допускаются участнику с административным правом, при владении purpose=maintenance и подтвержденном off затрагиваемых ресурсов. Чтения config.get, firmware.status и content.status владения не требуют. Изменения используют обычный cmdSeq; одна активная транзакция изменения на УУ. Транзакция переживает смену эпохи по правилам продолжения, но старая эпоха больше не меняет ее. Сервис уступает реальной тревоге по разделу 5; секретные ключи в конфигурацию не входят.

16.6 При config.get с configId снимок фиксирует выбранный пакет и его configRevision/appliedRevision. entry.id представляет положительный порядковый номер в канонической десятичной записи, начиная с 1, одинаковый для обеих facet этой записи. Обход идет от корня: свойства объекта в лексикографическом порядке имен, элементы массива в исходном порядке. facet=path имеет type=string и путь от корня JSON: сегменты отделены знаком /, имена полей сохранены, индексы массивов

начинаются с 0; например /targets/0/port. Отдельная запись пути массива перед его элементами имеет facet=value/type=counter и число элементов, включая 0. Объекты не создают отдельной записи значения. Вместе эти записи сохраняют структуру пакета, включая пустые коллекции.

16.7 Для скалярного поля facet=value содержит декодированное значение. Неотрицательные целые поля имеют type=counter, перечисления type=enum, activeValue type=bool; идентификаторы, токены, хеши и schemaVersion имеют type=string. Значение parameter.value использует type из своего parameter. Обе facet передаются полностью, частями до 64 байт UTF-8 без разрыва символа; part начинается с 1, parts одинаково во всех частях одной facet. Пустое значение передается одной частью с пустым text. Строки parameter.value содержат только символы XML 1.0: табуляцию, переводы строк и допустимые символы от U+0020; NUL, остальные управляющие символы ниже U+0020, непарные суррогаты, U+FFFE и U+FFFF запрещены. JSON-экранирование не отменяет это ограничение.

16.8 До staged получатель проверяет, что полные снимки принятой конфигурации и каталога программ помещаются в объявленные пределы страниц и времени чтения. Недостаточная емкость дает noSpace до commit; обрезка пакета или каталога запрещена. configRevision является счетчиком состояния получателя и отличается от revision линии configId внутри JSON. Новый commit присваивает configRevision=baseRevision+1 и тот же appliedRevision; исчерпание uint64 дает conflict. Повтор уже принятой команды или транзакции возвращает ее решение без нового увеличения счетчика.

16.9 config.get

Чтение selection с configId либо transactionId или snapshotId/cursor. Ответ configuration: snapshotId, configRevision, appliedRevision, more/cursor; записи entry id/facet/part/parts/type/text для разрешенных параметров либо transaction id/state/receivedBytes/totalBytes/hash/

remainingBudgetMs. Путь и значение до 128 байт передаются отдельными частями по 64 байта; обе facet и все части обязательны по разделу 7.

16.10 config.stage

transaction id/baseRevision/totalBytes/hashAlgorithm/hash и transfer profile/blockBytes/windowBlocks/windowRoundTripMaxMs. id:32 hex; totalBytes:1..1048576. Создает прием bulk/1 без изменения текущей конфигурации; байтов block в XML нет. Ответ transaction id/ state=receiving/receivedBytes/remainingBudgetMs. Повтор возвращает текущий префикс/состояние. Полный файл проходит validating и получает staged только после проверки длины, хеша, JSON-пакета и прав всех изменений; config.get читает прогресс и исход.

16.11 config.commit / config.cancel

transaction id. commit требует staged, совпадения baseRevision с текущей и отсутствия исполнения; атомарно фиксирует и применяет локальные правила получателя, без записи параметров ОУ; точная граница и исходы определены ниже. Ответ committed id/configRevision/ appliedRevision. cancel удаляет только незафиксированную транзакцию, ответ cancelled id. Точный повтор использует сохраненное решение; cancel после commit дает conflict.

16.12 network.stage

network id/address/prefix/gateway/confirmMs. id:32 hex, IPv4, prefix:0..32, confirmMs:10000..60000. Прежняя конфигурация хранится до подтверждения; ответ staged id/expiresAtMs. Новый адрес активируется как пробный, старый путь сохраняется на период подтверждения, если аппаратно допустимо.

16.13 network.confirm / network.cancel

network id. confirm принимается через новый адрес до expiresAtMs, с действующего управляющего соединения того же principalId и того же явно выбранного режима tls либо plain, что network.stage. Перед confirm

новый канал проходит protocol.hello и control.resume действующей эпохи; если она истекла, тот же участник получает новое control.acquire purpose=maintenance. Чужой principalId, смена режима, прежняя привязка и старый адрес не подтверждают переход. Для tls principalId аутентифицирован, для plain остается логическим идентификатором. confirm фиксирует новые параметры; cancel или истечение срока восстанавливает прежние. Ответ network id/state=committed|cancelled. Повтор, новое соединение и смена эпохи не продлевают expiresAtMs. При невозможности надежного возврата network.stage отклоняется до смены адреса.

16.14 time.set

clock utcMs/errorMs; uint64, errorMs:0..500. Изменяет UTC и оценку его погрешности; uptimeMs, lease, временная шкала и уже вычисленные сроки исполнения неизменны. Ответ clock utcMs/errorMs.

16.15 firmware.begin

image id/deviceKind/hardwareRevision/version/bytes/hashAlgorithm/hash/signature/keyId и transfer profile/blockBytes/windowBlocks/windowRoundTripMaxMs. id:32 hex; version uint64; bytes:1..33554432; подпись - 128 строчных hex, ГОСТ Р 34.10-2012, 256-битный вариант и GC256A, над Стрибог-256 манифеста. keyId выбирает предварительно внесенный доверенный ключ издателя. Ответ upload id=image.id/state=receiving/receivedBytes=0/remainingBudgetMs; байты поступают bulk/1.

16.16 firmware.verify

upload id. Проверяет размер, хеш байтов, подпись манифеста, совместимость device.kind/hardwareRevision и разрешенную version. Ответ upload id/state=verified либо status=accepted с operation id=upload.id. firmware.status сообщает verifying/verified/failed; failed содержит reason. Проверка ограничена 60000 мс и выполняется порциями до 100 мс. Правило владения во время проверки полного файла исключает прерыва-

ние обычным idle-сроком 10000 мс, но сохраняет lease и отзыв в пользу real. Отзыв владения во время verifying возвращает complete без продления общих сроков.

16.17 firmware.commit / firmware.status

commit: upload id, только verified и off; ответ accepted с operation id. Активируется только образ с сохраненным путем восстановления. status: selection operationId; ответ firmware operationId/state=receiving|complete|verifying|verified|activating|booted|failed|cancelled, receivedBytes/version/reason. booted подтверждается новым bootId и фактической version после загрузки, квитанция загрузчика не является booted.

16.18 Манифест образа - JCS JSON с полями deviceKind, hardwareRevision, version, bytes, hashAlgorithm="streebog256", hash. Все целые числа представлены десятичными строками. Подпись кодируется как 32-байтовое r, затем 32-байтовое s, оба в little-endian, в строчных hex. Производитель поддерживает монотонный минимальный разрешенный номер версии; откат ниже него дает rollbackDenied. Для передачи применяется расчетный бюджет и idleTimeoutMs по таблице ниже; повтор не сдвигает сроки. Состояния операции сохраняются до явной отмены или истечения, а результат commit - по общему resultRetentionMs.

16.19 При firmware.begin и firmware.verify устройство ДОЛЖНО проверять наличие ключа издателя, его состояние staged либо active, разрешенные deviceKind и hardwareRevision, а также минимальную разрешенную версию. Отсутствующий, отозванный или не подходящий по области действия ключ дает invalidSignature; версия ниже допустимой дает rollbackDenied. Непосредственно перед firmware.commit эти проверки ДОЛЖНЫ выполняться повторно по действующему реестру и текущему минимуму версии; ключ издателя ДОЛЖЕН находиться в состоянии active. Ранее полученное verified не отменяет последующий отзыв ключа или повышение минимума версии. Проверка подписи при

firmware.verify и проба staged-ключа не разрешают установку этим ключом до его активации.

Идентичность сервисной транзакции

16.20 Одна незавершенная транзакция изменения на УУ; она связана с principalId и неизменным манифестом, а не с эпохой. firmware upload.id= image.id=operation.id во всех стадиях от begin до booted; content использует один upload.id для приема, проверки и commit; config использует transaction.id. Повтор begin/stage своего участника проверяет весь манифест и параметры bulk/1, возвращает прежний ID и текущий receivedBytes без обнуления срока. Изменение любого поля дает conflict. Параметры канала, долговременные ACK, сроки и продолжение определены ниже в bulk/1. Терминальный failed/cancelled освобождает staging и сохраняет итог не менее resultRetentionMs; повтор ID с другим манифестом не превращается в новую транзакцию.

16.21 firmware.cancel: upload id, обычный cmdSeq владельца maintenance и право обновления. В receiving/complete/verifying/verified отменяет прием/проверку, возвращает upload id/state=cancelled и освобождает незакоммиченный образ; в activating/booted/failed - conflict; cancelled возвращает сохраненный итог. Повтор возвращает сохраненный итог. firmware.verify принимает complete, синхронный ответ verified либо accepted с operation.id; firmware.status всегда содержит operationId/ state/receivedBytes/version/remainingBudgetMs; в verifying дополнительно progressBytes, в failed - reason (код ошибки), в иных состояниях reason отсутствует. remainingBudgetMs=0 для терминального результата, иначе остаток действующего срока. commit допустим только в verified, задает activating и сохраняет ID в журнале загрузчика; booted относится к фактической новой version и bootId.

16.22 config.commit атомарен только для конфигурации самого получателя: хранение полного JSON-пакета и переключение локального

набора правил. В 0.21 пакет не содержит команд записи параметров ОУ, секретных ключей или изменения собственной аппаратной идентичности. Открытые ключи источников защищенной групповой речи задаются только в mediaGroups по специальным правилам этого профиля. Все ссылки, доступность локальных объектов и возможность применить правила без записи параметров в ОУ проверяются до commit. Успешный ответ committed содержит configRevision=appliedRevision новой редакции. Отказ до атомарной фиксации сохраняет обе старые редакции. Потеря ответа после фиксации выясняется через config.get/transaction.id. Групповой импорт на несколько узлов не атомарен: каждый узел имеет собственный итог. Физическая перенастройка ОУ этим пакетом не выполняется и не объявляется примененной.

Оконная передача файлов bulk/1

16.23 При объявлении сетевого размещения контента, транзакционной конфигурации или обновления прошивки обе стороны ДОЛЖНЫ поддерживать bulk/1. Управляющие команды создают, проверяют и фиксируют объект; его байты передаются отдельным TCP-соединением того же явно выбранного режима tls либо plain. В open-alert-draft/0.21 команды content.write и firmware.write отсутствуют, config.stage не содержит block. Использование 64-байтового обмена прежних редакций внутри 0.21 запрещено; неподдержанный action получает unsupported по общему правилу cmdSeq. Устройства без указанных файловых функций не обязаны реализовывать bulk/1.

16.24 transfer.capabilities выполняется на управляющем соединении без владения, с правом чтения возможностей. Запрос не содержит дочерних элементов. Ответ status=ok содержит единственный transfer: profile="bulk/1", port, maxBlockBytes, maxWindowBlocks, maxInFlightBytes. port:1..65535 задает отдельный от управления и медиа TCP-порт того же аутентифицированного УУ; иной адрес и переадресация запрещены. До-

пустимые maxBlockBytes:4096,8192,16384,32768,65536; maxWindowBlocks:4..64; maxInFlightBytes:16384..4194304 и не меньше 4*maxBlockBytes. Изменение этих возможностей меняет capabilityRevision. Минимальная совместимая пара: blockBytes=4096, windowBlocks=4.

16.25 firmware.begin, content.begin и config.stage содержат рядом с предметным манифестом единственный transfer с profile="bulk/1", blockBytes, windowBlocks, windowRoundTripMaxMs. blockBytes выбирается из указанного ряда, не больше maxBlockBytes; windowBlocks:4..maxWindowBlocks, произведение не больше maxInFlightBytes. windowRoundTripMaxMs:1..10000 — объявленный бюджет передачи и долговременного подтверждения полного окна, включая сеть, TLS, накопление ACK, Flash, планирование и необходимое восстановление владения. Невозможность принять точные параметры дает unsupported до резервирования; скрытого уменьшения блока или окна нет. Параметры входят в неизменный манифест транзакции; повтор begin/stage с другими параметрами дает conflict. Подписанный манифест образа сохраняет прежний состав и не включает параметры транспорта.

16.26 УУ резервирует staging-хранилище до подтверждения начала. Окно определяет число байтов в полете, а не обязательный массив такого размера в SRAM: приемник может порционно записывать проверенные данные выбранного режима (в tls — только после аутентификации MGM) в неактивную область и использовать ограниченный буфер записи. Непроверенная метка TLS не дает права публиковать либо подтверждать данные. Паспорт изделия задает память TLS/парсера/записи, размер очередей, геометрию Flash и максимальную непрерываемую операцию. Ресурсы управляющего канала, взаимодействия с ОУ и отмены резервируются независимо; недостаточность ресурсов отклоняет начало до изменения действующего объекта.

Привязка и ограждение файлового соединения

16.27 Файловый TCP-порт принимает тот же режим соединения и principalId, что управление: TLS 1.3 с проверкой ключей для tls, открытый TCP для plain; отдельная политика доверия не подменяет управляющую. Для tls сохраняются record_size_limit=512 и запреты 0-RTT/fallback. Бинарный блок может пересекать любое число TLS-записей и TCP-сегментов; размер блока не увеличивает лимит записи. Первое прикладное сообщение — обычная рамка uint32 BE + XML до 1024 байт: request v/profile/id/action="transfer.hello"/bootId/controlEpoch, один transfer id/kind. kind=config|content|firmware; id равен transaction.id для config и upload.id для остальных. profile равен выбранному open-alert-draft/0.21. protocol.hello и cmdSeq на этом соединении не передаются.

16.28 Привязка требует receiving, действующего управляющего соединения того же principalId, текущих bootId/controlEpoch, владения maintenance, соответствующего административного права и отсутствия иной файловой привязки. УУ атомарно связывает транзакцию с конкретными экземплярами управляющего и файлового соединений. До успеха ошибка закрывает только входящий канал, не меняет транзакцию, ее срок или действующее соединение. Ответ — обычная response v/profile/to/bootId/status=ok, один ready id/blockBytes/windowBlocks/receivedBytes/remainingBudgetMs. Величины блока и окна точно совпадают с begin; receivedBytes — устойчивый префикс на момент привязки. Ответ передается полностью до первого бинарного кадра. После него XML на файловом соединении запрещен.

16.29 Потеря или замена привязанного управляющего соединения, отзыв/освобождение/истечение эпохи либо прав, cancel, reboot и срок транзакции закрывают ее файловую привязку. Перед каждой долговременной фиксацией префикса повторно проверяются экземпляры соединений, principalId, bootId и controlEpoch. Запись, еще не зафиксированная при отзыве, не становится подтвержденной позже. Текущий сер-

висный шаг ограничен 100 мс; отзыв и ответ real сохраняют предел 500 мс. Медленный отправитель, заполненный TCP-буфер и ожидание ACK не удерживают эту границу. Поздний callback старого соединения не закрывает и не изменяет новую привязку. После разрыва файлового соединения тот же участник выполняет новый transfer.hello; если управляющее соединение и maintenance-эпоха еще действуют, они сохраняются. При утрате владения сначала получается новое maintenance-владение. Старый файловый канал не возобновляется.

Двоичные кадры и подтверждения

Таблица 19 - Двоичные кадры и подтверждения

Смещение	Поле	Представление
0..3	magic	4 байта ASCII OAB1: 4f 41 42 31.
4	type	uint8: 1 DATA, 2 ACK, 3 ERROR.
5	flags	uint8=0; неизвестные биты запрещены.
6..7	reserved	uint16=0.
8..11	payloadBytes	uint32 big-endian: число байтов после 20-байтового заголовка.
12..19	offset	uint64 big-endian: смещение DATA или устойчивый receivedBytes для ACK/ ERROR.

16.30 single-tail

Поля: DATA; offset=0

Эталонные байты hex:

4f4142310100000000000003000000000000000616263

16.31 tail-after-block

Поля: DATA; offset=4096

Эталонные байты hex:

4f4142310100000000000003000000000000001000010203

16.32 prefix-ack

Поля: ACK; offset=4096

Эталонные байты hex:

4f4142310200000000000000000000000000001000

16.33 stale-epoch

Поля: ERROR; offset=4096

Эталонные байты hex:

4f4142310300000000000002000000000000010000004

16.34 Каждый кадр состоит ровно из заголовка и payloadBytes байт. Проверка type, flags, reserved, размера, смещения, переполнения и окна выполняется до выделения памяти и записи полезной нагрузки. Неполный заголовок/блок при EOF не считается DATA. DATA передается только АРМ → УУ; ACK и ERROR только УУ → АРМ. Неверное направление, magic, тип, зарезервированное поле либо длина закрывают файловое соединение. Получатель может отправить ERROR, если заголовок достоверно разобран и отправка не задерживает управление; после ошибки продолжение на этом соединении запрещено. Невыдача ERROR не разрешает угадывать исход — клиент читает статус на управляющем канале.

16.35 DATA имеет payloadBytes=blockBytes, кроме последнего блока, содержащего остаток 1..blockBytes; offset кратен blockBytes, offset+payloadBytes<=totalBytes. После hello первый offset равен полученному receivedBytes. Следующий DATA начинается точно после предыдущего целиком принятого DATA; пропуски, перекрытия и повтор на одном соединении запрещены. TCP уже обеспечивает порядок и повторную доставку; приложение не пересылает блок повторно поверх живого потока.

После обрыва прием возобновляется только от устойчивого префикса нового hello, включая случай потерянного ACK. При receivedBytes=totalBytes hello НЕ ДОПУСКАЕТСЯ: клиент читает состояние и выполняет проверку/commit.

16.36 ACK имеет payloadBytes=0, offset=receivedBytes; он подтверждает весь непрерывный префикс, переживающий потерю питания. receivedBytes кратен blockBytes либо равен totalBytes. ACK не подтверждает хеш всего файла, подпись, staged, verified или commit. УУ поддерживает отдельно следующий принимаемый offset и устойчивый receivedBytes; после разрыва отбрасывает незафиксированный хвост. На одном соединении ACK не уменьшается и не превышает конец отправленных данных. Повтор ACK допустим, но не является продвижением. ACK старого экземпляра соединения игнорируется клиентом.

16.37 Пусть A — последний полученный клиентом ACK, а перед первым DATA A=receivedBytes из hello. Конец каждого отправляемого DATA ДОЛЖЕН быть не больше min(totalBytes,A+blockBytes*windowBlocks). Все разрешенные окном блоки могут отправляться без ожидания отдельных квитанций. УУ не уменьшает согласованное окно во время привязки; при недостатке ресурсов использует ограниченное TCP-противодавление, сохраняя сроки управления, либо закрывает канал с сохранением устойчивого префикса. УУ фиксирует и подтверждает поступивший непрерывный префикс, не ожидая заполнения окна: при полученном DATA прогресс ACK ДОЛЖЕН укладываться в windowRoundTripMaxMs от первого неподтвержденного DATA с учетом оговоренного сетевого бюджета. После долговременной фиксации ACK ставится на отправку не позднее 20 мс. Совокупный срок все равно ограничен windowRoundTripMaxMs; таймеры ACK не складываются сверх него.

16.38 Отправитель сохраняет время начала отправки каждого неподтвержденного DATA. Ни один такой блок не остается без долго-

временного подтверждения дольше windowRoundTripMaxMs; частичный ACK не обновляет время остальных блоков. Дополнительно для расчетных групп до S*W байт полный оборот от начала отправки группы до ACK ее конца укладывается в заявленный бюджет окна, включая сериализацию. При превышении срока самого старого блока клиент закрывает только файловое соединение, читает status/get, восстанавливает владение при необходимости и привязывается заново. Эти действия не увеличивают общий срок. Проверка связи выполняется по управляющему каналу; DATA, ACK и transfer.hello не продлевают lease. Сохраненный новый префикс ДОЛЖЕН освежать maintenance idleDeadlineMs до uptimeMs+10000, ограниченного сроком транзакции; чтения, повторы и пустое соединение не освежают его. Это не препятствует немедленному отзыву maintenance в пользу real.

16.39 ERROR имеет payloadBytes=2, payload — uint16 big-endian с кодом: 1 invalidArgument, 2 conflict, 3 forbidden, 4 staleEpoch, 5 resultExpired, 6 storageFault, 7 busy, 8 unsafeState. offset сообщает последний достоверно сохраненный префикс, после кадра соединение закрывается. Неизвестный код закрывает соединение и требует чтения статуса без изменения принятого префикса у клиента. Если носитель больше не позволяет подтвердить ранее сохраненный префикс, состояние транзакции становится failed/reason=storageFault; успешный ACK запрещен. Непривязанный канал не может создать ERROR для действующей транзакции.

16.40 Неполный заголовок либо DATA ДОЛЖЕН закончиться в пределах windowRoundTripMaxMs от первого байта этого кадра; при превышении приемник закрывает канал, не подтверждая хвост. Для TLS-handshake действуют общие сроки защищенного транспорта, а первый XML transfer.hello ДОЛЖЕН поступить полностью не позднее 10000 мс после handshake. Приемник ограничивает число незавершенных подключений, общий объем их памяти и частоту handshake по заранее за-

данным квотам; эти ресурсы не берутся из резерва управляющего канала. При отсутствии ожидающих данных общий и idle-сроки транзакции продолжают отсчитываться.

Сроки, устойчивость и завершение передачи

16.41 Для B байт, размера блока S и окна W число блоков N=ceil(B/S), число расчетных окон K=ceil(N/W), transferBudgetMs=K*windowRoundTripMaxMs+600000. Все вычисления выполняются без переполнения. Бюджет свыше 604800000 мс отклоняется до резервирования. Это срок передачи при объявленных условиях, а не обещание скорости произвольной сети. Срок включает паузы без владения и переподключения; повтор begin/hello не обнуляет его. idleTimeoutMs=600000 считается от последнего впервые устойчиво сохраненного префикса. При истечении незакоммиченная передача удаляется, дальнейшее продолжение дает resultExpired.

16.42 Получатель сохраняет ID, principalId, манифест, устойчивый префикс и расход бюджета атомарно с соответствующими данными до ACK. Последующее стирание/перезапись хвоста не разрушает подтвержденный префикс, даже если он делит физический erase-block с хвостом. Используются staging и журнал либо эквивалентная схема с проверяемым восстановлением; геометрия и износ носителя входят в паспорт реализации. НЕ ТРЕБУЕТСЯ отдельная синхронная запись метаданных на каждый DATA: ДОПУСКАЕТСЯ фиксация пакета принятых блоков в пределах окна и срока ACK. Действующий образ, конфигурация и опубликованный контент не перезаписываются приемом.

16.43 Расход монотонного бюджета записывается до каждого подтверждения нового префикса и не реже одного раза в 60000 мс во время работы. После reboot к последнему сохраненному расходу добавляется 60000 мс; недостоверный журнал дает resultExpired. Время выключенного питания не входит в бюджет работы. Новый idle-срок не превышает

min(600000,remainingBudgetMs). Перед продолжением сверяются целостность подтвержденных данных и журнал; несоответствие дает failed/storageFault, а не молчаливое уменьшение receivedBytes. Старые bootId/controlEpoch и файловые привязки не восстанавливаются.

16.44 При receivedBytes=totalBytes content и firmware переходят receiving → complete; файловое соединение закрывается после последнего ACK. config переходит receiving → validating и проверяет длину, Стрибог-256, JSON-схему, ссылки и права за время не более 60000 мс, порциями до 100 мс; результат staged либо failed с reason. config.get transactionId возвращает receiving|complete|validating|staged|committed|cancelled|failed, receivedBytes/totalBytes/hash/remainingBudgetMs, progressBytes только в validating и reason только в failed. content.commit и firmware.verify выполняют свои проверки только после полного приема; content.commit при длительной проверке отвечает accepted с operation.id=upload.id, content.status показывает verifying/progressBytes и затем committed либо failed/reason. Проверка занимает не более 60000 мс и не задерживает управление. content.status содержит upload id/state=receiving|complete|verifying|committed|cancelled|failed/receivedBytes/totalBytes/remainingBudgetMs; failed требует reason, verifying — progressBytes.

16.45 От полного приема до commit действует общий срок 600000 мс, включая проверку, потерю владения и ожидание. Переход в validating/verifying не начинает новый срок. Истечение удаляет только незакоммиченные данные. cancel ДОПУСКАЕТСЯ до атомарной фиксации и останавливает проверку; гонка cancel/commit имеет единственную точку решения и сохраненный итог. Для config/content отказ проверки оставляет прежнюю конфигурацию/ревизию действующей; failed не может стать committed повторным чтением. Ошибки проверки используют invalidArgument, conflict, invalidSignature, incompatible, rollbackDenied либо

storageFault по предметной причине. При полном чтении нет обязательного ожидания EOF для перехода в complete/validating.

Владение во время проверки полного файла

16.46 Переход в config.validating, content.verifying или firmware.verifying является активной ограниченной работой, а не ожиданием первого session.start. До начала работы УУ ДОЛЖНО проверить действующее maintenance-владение и атомарно установить внутренний verificationDeadlineMs=min(uptimeMs+60000, срок транзакции, срок после полного приема). Эти имена состояний обозначают соответствующую транзакцию; нового сетевого поля verificationDeadlineMs нет. На время проверки публикуемый ownership.idleDeadlineMs равен этому фиксированному сроку. Ранее установленный десятисекундный idle-срок не прерывает начатую проверку. Повтор команды, чтение статуса, heartbeat и изменение progressBytes НЕ ДОЛЖНЫ сдвигать verificationDeadlineMs. Перезапуск того же обработчика внутри действующей попытки также не начинает срок заново.

16.47 Lease управляющего канала остается независимым: controlLeaseMs=30000, heartbeatIntervalMs=5000. Проверка файла сама не продлевает lease. Отзыв в пользу real, потеря эпохи или прав и cancel останавливают проверку на ближайшей границе сервисного шага до 100 мс; решение о real-владении и ответ остаются в пределах 500 мс. Превышение verificationDeadlineMs дает failed/reason=resultExpired и освобождает незакоммиченное хранилище; поздний обработчик не публикует результат. При совпадении срока и завершения сначала применяется истечение: атомарная фиксация результата разрешена только при uptimeMs<verificationDeadlineMs. Более ранний срок транзакции или срока после полного приема имеет тот же приоритет.

16.48 После успешной проверки, отказа проверки либо cancel, если maintenance-эпоха еще действительна, начинается обычное ожидание:

idleDeadlineMs=uptimeMs+10000, ограниченное оставшимся сроком незавершенной транзакции. Завершение проверки не освежает lease. Состояния complete, staged и verified сами по себе не удерживают владение. Сохраненный терминальный результат не удерживает staging и доступен по общим правилам хранения результата.

16.49 При отзыве или потере maintenance-эпохи незавершенная validating/verifying прекращается на той же границе сервисного шага и возвращается в complete с сохраненными байтами и прежними сроками транзакции и после полного приема. Отложенный старый обработчик не фиксирует результат и не выполняет commit. config.get сообщает state=complete; повтор config.stage с тем же манифестом при новом владении снова запускает validating. Для content и firmware повторяются соответственно content.commit и firmware.verify новым обычным cmdSeq. Такая новая попытка получает собственные не более 60000 мс, ограниченные прежними сроками транзакции и после полного приема; эти общие сроки не продлеваются. Уже законченные staged/verified сохраняются; их commit заново проверяет текущую эпоху, права, off и применимость. Атомарно зафиксированный committed/activating не отменяется последующим отзывом и выясняется через статус.

16.50 bulk-verification-long

Входные условия: Полный файл, действующая maintenance-эпоха; проверка длится 20000 мс, heartbeat через 5000 мс.

Обязательный исход: config получает staged, firmware — verified, content — committed; прежний idle-срок 10000 мс не прерывает работу. После завершения idle не более 10000 мс.

16.51 bulk-verification-deadline

Входные условия: Проверка не завершилась за 60000 мс; heartbeat, чтения, повторы и progressBytes продолжаются.

Обязательный исход: failed/resultExpired в установленный срок; эти события не сдвигают срок проверки. Завершение на самой границе срока уже опоздало.

16.52 bulk-verification-lease

Входные условия: Проверка дольше 30000 мс без допустимого продления lease владельцем.

Обязательный исход: Истечение lease прекращает проверку на границе шага; файл возвращается в complete, старый обработчик не фиксирует результат.

16.53 bulk-verification-real

Входные условия: Допущенный acquire real при t=15000 мс незавершенной проверки.

Обязательный исход: Сервис прекращается до t=15100, новая эпоха и ответ до t=15500; старое завершение не дает staged/verified/ committed.

16.54 bulk-verification-restart

Входные условия: Отзыв эпохи в проверке; повтор под новым maintenance-владением, до общего срока остается 12000 мс.

Обязательный исход: Повторная проверка получает не более оставшихся 12000 мс, не новые общие 600000 мс. Повтор в той же попытке не меняет ее срок.

Таблица 20 - Расчет бюджета передачи bulk/1

Расчетный пример	**Блок / окно / оборот окна**	**Окна / бюджет**
1048576 байт	4096 байт / 4 / 100 мс	64 / 606400 мс
33554432 байт	4096 байт / 4 / 100 мс	2048 / 804800 мс
33554432 байт	4096 байт / 4 / 600 мс	2048 / 1828800 мс
33554432 байт	65536 байт / 64 / 100 мс	8 / 600800 мс

16.55 При 32 МиБ, S=4096, W=4 и обороте окна 100 мс расчетная передача занимает 204800 мс (3 мин 24,8 с) до общего запаса; при 600 мс — 1228800 мс (20 мин 28,8 с). Предельная полезная скорость в этой модели S*W/T составляет соответственно 163840 и примерно 27307 байт/с. В сравнении с 64 байтами и одним запросом при том же времени полного оборота количество последовательных ожиданий сокращается в 256 раз. Фактическая скорость ограничена также полосой, TLS и накопителем; паспорт не подменяет измерение. Более производительные изделия используют согласованные большие блоки и окна в тех же границах bulk/1.

16.56 Кодировки ключа зависят от поля и не взаимозаменяемы. JSON x/y и publicX/publicY — по 64 строчных hex от 32-байтового целого big-endian. Отпечаток закрепляемого ключа вычисляется как Стрибог-256 от 04 || xBE[32] || yBE[32]. TLS key_share GC256A по RFC 9367, пункт 6.1.1.3 передает xLE[32] || yLE[32] без байта 04. Подпись передает rLE[32] || sLE[32]; координаты ключа не являются половинами подписи. Преобразование выполняется отдельно для каждого поля; разворот всего 64-байтового ключа меняет также порядок координат и запрещен. Векторы содержат все три представления и отпечаток одного тестового ключа.

16.57 Подпись манифеста образа и пакета доверия кодируется одинаково с RFC 9367, пункт 5.3: r[32] || s[32], обе половины little-endian; полученные 64 байта представляются 128 строчными hex. Хеш трактуется как целое little-endian по модулю порядка подгруппы; ноль заменяется единицей. Старое прикладное s big-endian || r big-endian не принимается; автоматическое распознавание формата и DER-оболочка запрещены. Адаптер СКЗИ явно преобразует документированный выход своего API в этот формат ровно один раз. signature-vectors содержат JCS, хеш, GC256A, r/s, положительные подписи и отрицательные перестановки; прежний формат включен в отрицательные примеры.

17 Сторонние сервисы и расширения

17.1 Внешние сервисы подключаются адаптерами к АРМ. Единая модель содержит предупреждение, ревизию, запуск, ветвь, область ресурсов, срок и происхождение. Для каждого адаптера задаются principalId, разрешенные операции, преобразование данных, дедупликация, подтверждения, таймауты и аудит.

Таблица 21 - Сторонние сервисы и расширения

Вид интеграции	**Граница ответственности**
CAP / EDXL и обмен предупреждениями	АРМ интерпретирует содержание, территорию, обновление и отмену. CAP не заменяет команды усилителю.
REST, webhook, MQTT, очереди сообщений	Способ доставки адаптера. Семантика повторов и происхождение не определяются названием транспорта.
Телефония / SIP / источники речи	АРМ проверяет право источника и преобразует поток в объявленный медиапрофиль.
АСУ, диспетчеризация, мониторинг	Отдельные права наблюдения и запуска, качество измерений, ограничение частоты и обратная доставка результата.
ОУ другого производителя	Согласованный с изготовителем ОУ интерфейс, каталог ресурсов, критерии исполнения и профиль испытаний.

17.2 Нельзя заранее гарантировать поддержку любого внешнего API. Расширяемость задается контрактом адаптера, версиями и

capabilities. Неизвестная обязательная функция отклоняется целиком; расширение не может обходить право остановки или менять смысл существующего поля.

17.3 УУ передает уведомления по согласованной подписке eventsPush либо через чтение журнала events.read с общими правилами курсора и качества. В управляющем профиле 0.21 инициативные изменяющие запросы УУ к АРМ запрещены. Адаптер внешнего сервиса получает самостоятельную идентичность и права; ответ УУ не является командой другому прибору.

18 Проверки и критерии готовности

Соответствие речевой части определяется для каждого объявленного режима: локальной записи, потока записи и микрофона. Критерии M01-M10 раздела 15 применяются к полному тракту от источника до выходов; прием команды и готовность декодера не заменяют критерии полноты сообщения и физического результата.

18.1 Оценка соответствия выполняется по обязательным требованиям выбранного профиля и критериям настоящего раздела. Для каждого условия фиксируются входная последовательность, допустимые ответы, изменения состояния и физическое наблюдение. Повтор, перестановка, потеря сообщения и перезапуск проверяются как самостоятельные случаи. Отчеты и сведения об используемых средствах испытаний являются документами реализации.

18.2 Повтор / потерянный ответ / reconnect

session.start не исполняется повторно; журнал результата отличает accepted от текущего running.

18.3 Отмена до session.start и поздняя отмена

Поздний session.start не выполняется, новый сеанс не останавливается старой отменой.

18.4 Сеть, питание и зависшее ОУ

stop/finish выполняются в заданной области; unknown не превращается в off; автоматический рестарт отсутствует.

18.5 Датчики и опрос

Устаревание, отсутствие, таймаут и авария различаются; полнота опроса не выдается за исправность.

18.6 Парсер / транспорт / защита

Разрывы записей, неверные длины, подделка, повторы и перегрузка не приводят к действию до полной проверки.

18.7 Нагрузка

Память, максимальный шаг цикла и задержка session.stop не превышают объявленные пределы при одновременном опросе, аудио и сервисной передаче.

18.8 Питание при обновлении

Загрузчик, конфигурация и контент восстанавливаются после обрыва питания в любой точке изменения на каждой объявленной аппаратной ревизии.

18.9 Рисунок сирены

Длительности фаз, начальная фаза, паузы, последний неполный цикл, пределы подготовки и выключения соответствуют одной редакции аппаратного профиля; одинаковый pattern не подменяет проверку одинакового сигнала.

18.10 Границы таймеров

Таймаут отправки после свежего продления lease только отсоединяет канал; stop/finish применяется при истечении оставшегося lease. Отзыв действующих полномочий проверяется отдельно и действует немедленно. Разрыв канала не сбрасывает lease; resume до и после его истечения различается; draining не допускает новый session.start. Повторы не продлевают подготовку, исполнение и остановку. При совпадении сроков действует заданный порядок обработки.

18.11 Последовательные сеансы

При подтвержденном off и действующих полномочиях переходы сирена -> запись и проверка -> оповещение укладываются в пределы

паспорта при занятой истории. Испытываются все поддержанные переходы, максимальный всплеск и длительная заявленная частота, включая смену эпох. В пределах нагрузки нет busy только из-за истории; все прежние результаты доступны до истечения их срока. Переполнение за пределами нагрузки не вытесняет их и не вызывает воздействие.

18.12 Хранение результата

В одной загрузке результат доступен весь resultRetentionMs, в том числе после release; reboot дает staleBoot старому запросу и запускает восстановление по постоянному журналу без повторного session.start. Полный архив испытывается только при заявленном отдельном профиле.

18.13 Ресурсный бюджет определяется для целевого состава выпускаемой прошивки: код и загрузчик, конфигурация и ключи, постоянный журнал, пиковые SRAM/стек/SDRAM, сетевые соединения, криптография и тракт ОУ при одновременной нагрузке. Совместное размещение прежних протоколов и служебных приложений не является требованием открытого протокола. Первичное заводское программирование и последующее эксплуатационное обновление имеют отдельные процедуры; наличие первого не доказывает восстановление после сбоя второго. Емкость и сроки подтверждаются измерениями изделия, а не размером прежнего образа.

Дополнительная приемка

18.14 Для телеметрии обязательны сценарии изменения качества, пропуска событий, неполных снимков и запуска теста; для интеграции - происхождение, делегирование и частичный результат. Дополнительные сценарии U01-U12 устанавливают критерии взаимодействия АРМ и ОУ. Метод испытания ДОЛЖЕН сохранять проверяемые предпосылки, событие и ожидаемый результат.

18.15 U01-U03: этапы и речь

Запоздавший session.stop сирены не выключает следующий этап; отмена в паузе запрещает повтор; устаревшее разрешение тангенты не открывает поток.

18.16 U04-U06: доставка до ОУ

Потеря подтверждения ОУ не вызывает второй session.start; резервный путь не дублирует неизвестный запуск; фазовые и общие показания разных вариантов ОУ не смешиваются.

18.17 U07-U09: наблюдения

Перестановка/повтор/пропуск страниц не меняют адресацию и полноту; активная до подключения авария видна без нового фронта; повтор эпизода или снимок не перезапускают сценарий.

18.18 U10-U12: пределы и совместимость

Таймаут программы не выдается за успешное окончание; два независимых АРМ одинаково трактуют профиль и отказ неизвестной функции; session.stop укладывается в измеренный бюджет под нагрузкой.

Проверка инициативных уведомлений

18.19 Для eventsPush проверяются отсутствие notification до подписки, доставка сработки и возврата без events.read, неизменный повтор после потери notificationAck, отсутствие повторного применения события, две независимые подписки, замена и отмена, чужой курсор и bootId, восстановление после обрыва, отзыв прав и явный gap при переполнении. Отдельно проверяются plain/tls, старый клиент, старое УУ, целостность XML при частичной записи, граница 1024 байта, приоритет STOP и сохранение сроков START/STOP и непрерывности речи под нагрузкой. Успех модели формата не заменяет аппаратную приемку.

18.20 Приемка 0.21 включает: UDP BIND с чужим ключом и верным mediaId во время звучания; повторную привязку того же владельца;

поздний пакет старой подготовки; технический session.start с maxDurationMs=30001 и учебный session.start с maxDurationMs=1080001; тревогу во время короткого и длительного учения с проверкой независимого тракта; отказ записи результата после off; edge и level с активным входом при reboot; удержание уровня после законченной программы; снятие уровня до освобождения ресурса. Для level проверяются новый физический отсчет и новый sessionId; для edge отсутствие фронта запрещает запуск. Успешный снимок с failed-проверкой либо неподтвержденным off ДОЛЖЕН отклоняться независимо от счетчиков записей.

19 Контракт управляющего ядра

19.1 Управляющий профиль open-alert-draft/0.21 задает конечное исполнение и единственного владельца. Форматы и межполевые ограничения настоящего раздела обязательны совместно с временной шкалой раздела 8 и правилами резервирования cmdSeq раздела 6. Разрешение функции определяется объявленными возможностями и полномочиями участника; синтаксически корректное сообщение не отменяет проверку состояния.

Подключение

19.2 1. Транспорт

АРМ явно выбирает tls либо plain по настроенному адресу и порту УУ. Для tls профиль доверия известен при регистрации; прикладные команды до аутентификации не принимаются. Для plain ключи не требуются, но securityPolicy=secureOnly запрещает подключение.

19.3 2. Начальный обмен

Первый запрос: profile=bootstrap/1, action=protocol.hello, список 1-8 поддерживаемых профилей. УУ пересекает предложение клиента со своим разрешенным списком и выбирает профиль по собственному настроенному порядку приоритета; порядок клиента не влияет на выбор. Возвращаются selectedProfile, deviceId, security=tls|plain и bootId. Запре-

щенный локальной политикой профиль не участвует в выборе; клиент также проверяет свой разрешенный список. У ядра 0.21 единственный профиль open-alert-draft/0.21, другие профили допускаются только отдельными спецификациями. При отсутствии пересечения - incompatibleProfile и закрытие.

19.4 3. Фиксация

На данном соединении последующие запросы используют только selectedProfile. Повтор того же hello допустим до первой предметной команды и возвращает тот же выбор; смена списка требует нового соединения.

19.5 4. Наблюдение

device.describe/capabilities/state; АРМ проверяет kind, ресурсы, ревизии, часы uptimeMs и сроки профиля. Чтение не требует acquire.

19.6 5. Управление

acquire с сохраненными acquireSeq/instanceId, затем session.start. При потере канала - восстановление контекста и resume, без повторного физического запуска.

Прием запуска

19.7 Оболочка

v=1, profile, id, action=session.start, bootId, controlEpoch, cmdSeq обязательны. ID - 32 строчных hex, cmdSeq - uint64 > 0.

19.8 session

Ровно один элемент: mode=siren|recording|program|stream|test, purpose=real|exercise|technical, acceptBeforeMs, maxDurationMs, configRevision и capabilityRevision обязательны. Значений по умолчанию нет.

19.9 acceptBeforeMs

uint64 в монотонной шкале УУ данного bootId. session.start принимается только при uptimeMs < acceptBeforeMs <= uptimeMs + 30000.

Повтор принятого session.start возвращает решение до повторной проверки срока.

19.10 maxDurationMs

Целое 100..86400000 мс от принятия, с подготовкой и паузами. Для test и purpose=technical верхняя граница 30000; для purpose=exercise — 1080000 с отдельными условиями допуска длительных учений. Исчерпание инициирует остановку с maxDurationExceeded.

19.11 resources

1-4 элемента resource: id длиной 1-32 ASCII-символа, bindingGeneration uint64. ID уникальны. Четыре ссылки любого режима с максимальными разрешенными длинами полей ДОЛЖНЫ помещаться в 1024 байтах компактного XML без отступов.

19.12 siren

Ровно один элемент siren: pattern=continuous|intermittent, stopMode=duration|command. Для duration обязателен durationMs=100..3600000 и durationMs <= maxDurationMs; для command durationMs запрещен. Рисунок сигнала определяется аппаратным профилем ниже; durationMs отсчитывается от running и включает паузы.

19.13 recording

Ровно один элемент recording: contentId, revision uint64, hashAlgorithm=streebog256 и hash из 64 строчных hex. Один проход; естественный конец отличается от maxDurationExceeded.

19.14 test

Ровно один элемент test: testId и testRevision uint64; purpose ДОЛЖНО быть technical. Эффекты и критерии заданы полным каталогом tests.list по разделу 12; testId/testRevision фиксируются до воздействия.

19.15 program

Ровно один program id/revision; полный неизменный план из programs.list, правила исполнения и результата раздела 26.

19.16 stream

Ровно один stream mediaId из audio.prepare; onControlLoss=stop, правила rtp-alert/5 раздела 15.

19.17 Ревизии

Все ресурсы, поколения и ревизии проверяются до воздействия. Их несовпадение дает conflict. Отсутствие обязательного поля не заменяется значением из старой карточки АРМ.

19.18 Порядок приема session.start: полное обрамление и лимиты; синтаксис и общая оболочка; аутентифицированные права и контекст; дедупликация/расходование cmdSeq; предметная схема и межполевые условия; актуальные ревизии и возможности; срок приема; занятость и готовность; сохранение решения accepted; передача задания исполнителю. После принятия физические отказы отражаются в результате сеанса. Утрата ответа не разрешает новый session.start с новым номером.

19.19 АРМ рассчитывает acceptBeforeMs из свежего uptimeMs и конечного бюджета доставки, учитывая RTT, локальное прошедшее время и границу дрейфа. При ретрансляции исходный срок поручения не продлевается. Если соответствие шкал нельзя ограничить, запуск с таким сроком не принимается как своевременный. maxDurationMs и таймеры остановки не зависят от коррекции UTC.

Физический рисунок сирены

19.20 continuous

Непрерывная подача сигнала без программных пауз. Частотный состав, модуляция, разгон и выбег определяются опубликованным профилем ОУ; continuous не означает постоянное мгновенное значение акустического сигнала.

19.21 intermittent

Циклический рисунок из фаз сигнала и пауз. Профиль ДОЛЖЕН задавать их порядок и длительности, начальную фазу, допуски, поведение

при остановке и точную редакцию нормативного описания, если он на него ссылается. Одного названия "прерывистая сирена" недостаточно.

19.22 Исполнитель рисунка

Профиль указывает, формирует ли рисунок само ОУ или драйвер УУ, чем подтверждается выбор рисунка и как начало его первой фазы связано с критерием running. Общий ответ ОУ "режим сирены" сам по себе не доказывает выбранный рисунок.

19.23 Длительность и последний цикл

При stopMode=duration durationMs - интервал от running, включая паузы; по его истечении начинается stopping, последний неполный цикл не продлевается до полного. При command штатную остановку задает адресный session.stop. Для речевого ОУ с signalKind=builtIn естественный конец собственного цикла также определяется свежим переходом из подтвержденного режима сирены в покой без аварии; после него УУ подтверждает off по профилю и завершает сеанс с succeeded/completed. Покой до подтверждения running, иной режим, устаревший ответ и авария не доказывают завершение. Во время остановки новые фазы включения запрещены. Задержка физического выключения ограничена stopBeginMs/stopConfirmMs и критерием off профиля.

19.24 Предел всего сеанса

maxDurationMs идет от acceptedAtMs и имеет приоритет при одновременном истечении с durationMs. Для stopMode=duration АРМ оставляет бюджет подготовки сверх durationMs. Для command достижение maxDurationMs является защитным отказом; применяется onControlLoss=stop. Равенство durationMs=maxDurationMs допустимо по схеме, но не обещает полный интервал running или успешное завершение.

19.25 Привязка до session.start

При вводе в эксплуатацию АРМ и УУ получают одно и то же опубликованное описание сигнала с идентификатором и редакцией для каж-

дого ресурса. Оно связано с deviceId, resourceId, bindingGeneration и принятыми configRevision/capabilityRevision. Смена рисунка меняет соответствующие ревизии; для активного сеанса профиль неизменен.

19.26 Граница ядра

session.start выбирает один из двух рисунков аппаратного профиля; произвольные длительности фаз в session.start не передаются. profiles.get предоставляет параметры и идентификатор описания сигнала. Полное физическое описание входит в паспорт изделия при вводе в эксплуатацию; автоматическое угадывание по одному pattern запрещено.

19.27 Неподдержанное исполнение

Допустимый XML не обещает любую длительность на любом ОУ. УУ ДОЛЖЕН отклонить неподдержанный рисунок или длительность с unsupported до воздействия. Молчаливое округление, замена рисунка или ожидание полного цикла сверх срока не допускаются; ограничения и допуски публикуются в профиле.

Чтение и снимки

19.28 device.describe

Запрос без дочернего элемента. Ответ device: deviceId, kind=vau|siren, version, uptimeMs, securityPolicy=optional|secureOnly; отдельный элемент info protocolName="Открытый протокол систем оповещения". manufacturer/softwareDeveloper необязательны и обозначают фактического изготовителя и разработчика ПО. bootId в оболочке.

19.29 device.capabilities

Первая страница без параметров; продолжение selection snapshotId/cursor. capabilities содержит capabilityRevision и целые записи mode, command, resource, limit и mediaProfile. Конец каталога определяет more=false.

19.30 device.state

Первая страница без параметров; продолжение selection snapshotId/cursor. state содержит stateRevision, eventSeq, capabilityRevision, uptimeMs, resourceCount, alarmCount, sessionCount. Первая страница начинается ровно с configuration, ownership, recovery; продолжения их не повторяют. resource, session и alarm передаются целыми записями.

19.31 inventory.get

Необязательный selection с одним из pointId/resourceId/ouId; при продолжении фильтр сохраняется и добавляется snapshotId/cursor. inventory содержит capabilityRevision/bindingGeneration и целые записи point, enum, threshold, condition.

19.32 sensors.read

selection содержит 1-8 уникальных point id; при продолжении набор сохраняется. snapshot содержит capabilityRevision/bindingGeneration, stateRevision, eventSeq, evaluatedAtMs и complete.

19.33 events.read

events bootId/afterSeq; ответ events nextSeq/more/gap/oldestSeq и записи event. При gap=true записей нет, more=false, nextSeq=afterSeq; клиент получает новый снимок.

19.34 Страницы

Все контейнеры страниц, кроме events, содержат snapshotId и more; cursor обязателен только при more=true. Запрос продолжения содержит одновременно snapshotId/cursor. Снимок и критерии выборки неизменны, срок хранения снимка 30000 мс; после истечения - snapshotExpired.

19.35 Контекст чтения привязан к bootId, полученному bootstrap. Перезагрузка закрывает канал и требует нового bootstrap; явно переданный старый bootId получает staleBoot. Сводка session передается только на первой странице session.get; продолжения не повторяют ее. Приня-

тые ревизии и время не меняются от чтения следующей страницы. Для чтения нового состояния начинается новый снимок.

19.36 configuration обязательна на первой странице device.state независимо от поддержки config.get. configRevision - текущая сохраненная редакция, appliedRevision - фактически примененная, onControlLoss - действующая политика примененной редакции. При несовпадении редакций session.start получает unsafeState. Для первого session.start АРМ берет configRevision и capabilityRevision из согласованного состояния и повторно проверяет их при запуске; несвежая ревизия дает conflict/detail. Ревизия присваивается любой смене политики или параметров запуска. Разные appliedRevision и configRevision не дают АРМ права угадывать будущую политику.

19.37 Ответ device.describe содержит ровно один device и один info с единственным атрибутом protocolName="Открытый протокол систем оповещения". info не содержит наименования компании, рекламы или обязательной ссылки. Поле protocolDeveloper отсутствует в формате 0.21 и не является условием соответствия. Идентификаторы фактических manufacturer/softwareDeveloper необязательны. Условия свободного использования приведены в приложении Е; независимая реализация не обязана показывать сведения об издателе спецификации.

19.38 ownership действующей эпохи дополнительно содержит purpose; при отсутствии сеанса - idleDeadlineMs. Снимок device.state фиксирует все ресурсы и текущий сеанс, а также полный набор незавершенных аварийных условий: active и unknown. Счетчики resourceCount/alarmCount/sessionCount относятся ко всему снимку, sessionCount равен 0 или 1. Ресурс, сеанс и авария передаются ровно один раз за все страницы. configuration/ownership/recovery передаются ровно один раз на первой странице. Метаданные контейнера, кроме more/cursor, повторяются без изменений. ownership в held/detached/draining содержит principalId и controlEpoch; free/released их не содержит;

для blocked допустима либо эта пара, либо отсутствие обоих полей. Отсутствие аварии означает отсутствие актуального условия только после проверки последней страницы и всех счетчиков; частичное чтение не очищает аварии в АРМ.

Аварии

19.39 Идентичность

conditionId - устойчивый ID условия (token 1-32), pointId - точка каталога, alarmId - уникальный 32-hex ID эпизода. В снимке не более одного эпизода для conditionId; alarmId не используется повторно.

19.40 Состояние и качество

state=active|cleared|unknown, severity=info|warning|critical; validity, freshness, observedAtMs, changedAtMs, continuity обязательны. active/cleared требуют valid/fresh; потеря достоверного наблюдения дает unknown, а не cleared. Время - монотонные миллисекунды текущего bootId; changedAtMs <= observedAtMs.

19.41 Снимок и журнал

state содержит active/unknown. events.read: тип alarm.raised несет ровно один alarm state=active, alarm.cleared - state=cleared, alarm.qualityChanged - active/unknown того же эпизода, без подтверждения снятия. Изменение состояния и публикация события атомарно получают stateRevision и eventSeq. Подтвержденное снятие исключает эпизод из следующих снимков.

19.42 Восстановление

В одном bootId alarmId сохраняется при разрыве связи и переполнении журнала. После reboot достоверно восстановленный эпизод сохраняет ID; временные метки начинаются в новой шкале. При утрате доказанной преемственности создается новый ID с continuity=unknown, без утверждения о новом фронте. continuity=continuous требует сохра-

ненного происхождения. Восстановленный снимок сам по себе не создает повторное внешнее действие.

Каталог точек

19.43 point facet=identity

id, facet, resourceId, ouId обязательны; parentId и sourceId необязательны.

19.44 point facet=meaning

id, facet, kind, type обязательны; unit и scale необязательны.

19.45 point facet=source

id, facet, sourceKind, presence, support обязательны. Поддержка не подменяет физическое наличие.

19.46 point facet=acquisition

id, facet, readMethod, maxAgeMs обязательны.

19.47 enum

pointId, code и label обязательны. Уникальность (pointId, code). Только для type=enum, как минимум одна запись. Строка label не более 64 декодированных байт UTF-8; этот предел учитывает расширение XML-сущностей при кодировании.

19.48 threshold

pointId, conditionId, operator, value, hysteresis, severity обязательны. Только int/counter; value и hysteresis - XML decimal без экспоненты, hysteresis >= 0. Порог сравнивает переданное числовое значение до применения scale. conditionId уникален в каталоге.

19.49 Полный каталог содержит четыре facet для каждой выбранной точки; enum и threshold не повторяют их поля. Записи связываются через pointId и общий snapshotId/capabilityRevision/bindingGeneration. Фильтр pointId/resourceId/ouId применяется ко всем частям описания, а не только к identity. Для порога gt/ge условие снимается при значении ниже value-hysteresis; для lt/le - выше value+hysteresis; eq/ne требуют

hysteresis=0 и снимаются при ложном сравнении. Значения enum и пороги не являются командами воздействия.

19.50 В ядре sensors.read возвращает полный выбранный набор: complete=true на всех страницах обозначает полноту зафиксированного набора, more=false - конец передачи. Недоступное показание передается записью с validity=unavailable и reason; не удаляется из набора. Неизвестный pointId отклоняет запрос с notFound. Запрошенная точка передается ровно один раз; посторонние точки запрещены. Изменение качества после фиксации снимка появляется в новом снимке или событии.

19.51 Полнота определяется всей последовательностью запросов и ответов снимка. Обязательны первая страница без cursor, неизменные критерии выборки и метаданные, точная цепочка snapshotId/cursor, отсутствие циклов, повторов и лишних страниц, конечное more=false и равенство счетчиков/множества выбранных точек. Для session.get проверяются счетчики resourceCount/checkCount/observationCount/ executionCount/faultCount, ссылки и условия успеха; сводка сеанса есть только на первой странице. Для inventory.get проверяются все facet и ссылки enum/threshold. Проверка одного ответа не доказывает полноту многостраничной передачи.

19.52 Ниже приведен закрытый словарь значений профиля. Файл vocabulary-0.21.json содержит те же перечисления и коды ошибок. Произвольные значения вместо перечислений запрещены. conflict, forbidden и busy ДОЛЖНЫ содержать detail из conflictDetail, forbiddenDetail и busyDetail соответственно; прочие ошибки detail не передают. startBlockedByCancel ДОЛЖЕН содержать cancelSeq. Для failed/ inconclusive/skipped/pending в check обязателен reason; skipped означает, что проверка не выполнялась. pending разрешен только до finished.

Результат сеанса

19.53 Контейнер result

snapshotId, stateRevision, mode, evaluatedAtMs, resourceCount (1-4), checkCount, observationCount, executionCount, faultCount и more обязательны. cursor только при more=true. Счетчики относятся ко всему неизменному снимку.

19.54 Сводка session

Ровно одна на первой странице, отсутствует в продолжениях. Все принятые идентификаторы, ревизии, времена и onControlLoss сохраняются. Выбор по startSeq требует исходного controlEpoch.

19.55 resource

id, bindingGeneration, output, evidence, observedAtMs обязательны. Набор точно совпадает с принятым session.start; каждый ресурс передается один раз.

19.56 check

id, resourceId, phase=baseline|running|stopped, status обязательны; reason обязателен кроме passed. ID уникален во всем результате. pending означает ожидание выполнения; в finished запрещен.

19.57 observation

id, resourceId, pointId, phase, sourceId, sourceKind, type, validity, freshness, observedAtMs обязательны. value обязателен для valid; reason обязателен для invalid/unavailable. Всего не более 12 атрибутов.

19.58 План activationPoll

До воздействия tests.list фиксирует testId/testRevision, полный набор checks (id/resourceId/phase) и observations (id/resourceId/pointId/phase/type/sourceId/sourceKind), критерии и сроки. Для каждого ресурса предусмотрена хотя бы одна проверка. Наблюдения покрывают все три фазы для каждого ресурса. Все объявленные проверки и наблюдения этого теста обязательны для общего успеха.

19.59 Неполученное наблюдение

Запись сохраняет свой ID и фазу с validity=unavailable, freshness=never, observedAtMs=0 и reason. До исполнения check=pending; после завершения невыполненная проверка skipped. Исчезновение записи не заменяет отрицательный результат.

19.60 Условия успеха

succeeded требует off с допустимым evidence для всех ресурсов и passed для всех checks. Для test все observations также valid/fresh на момент соответствующей фазы. Свежесть исторического измерения здесь оценивается в фазе теста; чтение результата позже не превращает его в текущее показание. В finished причина completed требует succeeded, operatorStop допускает succeeded только по критериям подтвержденной работы раздела 8; в остальных случаях применяется cancelled либо unknown, maxDurationExceeded требует failed. Подтвержденное выключение после исчерпания предела не делает поручение выполненным.

19.61 Проверка полноты

Получатель проверяет страницы, счетчики и внутренние ссылки, сверяет результат с исходными session.start/accepted и фиксированным манифестом теста. Одни счетчики не доказывают, что сам отправитель не исключил ожидаемую проверку.

19.62 Для siren/recording/program/stream записи check и фазовые observation необязательны; resourceCount положителен. Для test каталог tests.list, полный набор критериев и фазовые наблюдения обязательны. Результат связывается с сохраненным accepted исходного session.start и его неизменным планом. Полный снимок содержит одну session, одну timeline и одну completion; execution и fault обязательны по счетчикам и правилам раздела 8.

19.63 Первая страница state обязательно содержит configuration, ownership, recovery в этом порядке; первая страница result обязательно содержит session. Если предметные записи не помещаются, допускают-

ся только этот обязательный набор и метаданные контейнера при more=true. Пустой контейнер вместо обязательного набора запрещен. Каждая следующая непоследняя страница ДОЛЖНА содержать полезную запись; сводка в продолжениях запрещена. УУ ДОЛЖЕН проверять кодируемость всех объявленных записей с оболочкой и cursor до объявления возможностей. Запись не дробится на произвольные фрагменты XML. Максимальные длины полей и расширение XML-сущностей учитываются при кодировании; session.start поддерживает четыре ресурса при лимите 1024 байта.

Типы показаний

19.64 bool

Только true или false.

19.65 int

Каноническое десятичное int64: -9223372036854775808..9223372036854775807; без плюса, ведущих нулей, -0 и экспоненты.

19.66 counter

Каноническое десятичное uint64: 0..18446744073709551615.

19.67 enum

Код token 1-32 из enum данного pointId в согласованном каталоге.

19.68 type / scale

Каждое показание содержит type, совпадающий с каталогом. scale - положительный десятичный множитель без экспоненты, до 32 ASCII-символов, при отсутствии 1. Отображаемое значение равно value * scale; пороги сравнивают исходное value.

19.69 freshness=never означает отсутствие полученного измерения: value отсутствует, observedAtMs=0, reason обязателен. type обязателен и для sensor.qualityChanged. Получатель проверяет лексическую форму и диапазон, связывает полный каталог и readings по bootId/

capabilityRevision/bindingGeneration, проверяет тип и принадлежность enum. Нельзя принять новый тип из показания как неявное изменение каталога. evidence имеет закрытый словарь: none, commandSent, reportedMode, electrical, acoustic, localConfirmation. none/commandSent не разрешают off. Допустимость остальных доказательств, их источник и свежесть определяются аппаратным профилем и не выводятся из одного имени.

19.70 В снимке показаний observedAtMs ДОЛЖНО быть не больше evaluatedAtMs; changedAtMs, если присутствует, не больше observedAtMs. Для freshness=fresh возраст evaluatedAtMs - observedAtMs ДОЛЖЕН быть не больше maxAgeMs из acquisition каталога. Если показание повторяет maxAgeMs, значение ДОЛЖНО совпадать с каталогом. Получатель применяет maxAgeMs из каталога независимо от наличия этого атрибута в показании. Исторические observations результата проверяются относительно своей фазы, а не времени чтения результата.

19.71 Для аварии observedAtMs относится к последнему полученному наблюдению источника. changedAtMs фиксирует последний подтвержденный источником переход состояния условия. Переход качества в unknown при потере источника не обновляет эти времена; его порядок задается eventSeq. Если наблюдений еще не было, оба времени равны 0 и freshness=never. Чтение кеша и пересчет качества не создают наблюдения. Подтверждение control.release ДОЛЖНО содержать исходный controlEpoch; leaseRemainingMs в channel.ping, acquire и resume ДОЛЖНО быть не больше 30000.

События

Таблица 22 - События

Событие ядра	Ровно одно обязательное тело
session.started / session.finished	session со state=running / finished соответственно; завершение требует result/reason. Событие сообщает сводку, полный результат читается через session.get.
sensor.qualityChanged / resource.changed	point с обязательными типом и качеством / resource с output, evidence и observedAtMs.
alarm.raised / alarm.cleared / alarm.qualityChanged	alarm по правилам эпизода выше.
inventory.changed	Без дочерних записей; новая capabilityRevision в event требует перечитать каталог.

19.72 session.execution и session.fault содержат execution и fault соответственно по разделу 19. Неизвестный тип события в ядре запрещен; poll.finished и прочие расширения требуют другого согласованного профиля. Курсор сдвигается только после проверки тела. Пустое session.finished не является допустимой квитанцией о завершении.

19.73 При поддержке речи device.capabilities передает записи mediaProfile с id, security=tls|plain, codec=PCMA|GSM и packetMs. Одна запись задает один поддержанный вариант; ключ уникальности — все четыре поля. Для rtp-alert/5 PCMA требует 20 мс, GSM допускает 20/40/80; для group-audio/2 и group-audio-open/1 фиксированы GSM и 100 мс звука на часть, security соответственно tls и plain. Объявление command audio.group.prepare не заменяет объявления точного

медиапрофиля. Необъявленное сочетание получает unsupported до выделения подготовки.

Лимиты реализации

19.74 Начальный обмен выбирает точный профиль с фиксированными транспортными пределами; встречного предложения меньших лимитов в hello 0.21 нет. Ответ device.capabilities ДОЛЖЕН содержать все обязательные команды и каждое обязательное имя limit из таблицы ниже ровно один раз за полный каталог. Для постоянных значений ТРЕБУЕТСЯ точное совпадение. Прочие значения задают паспорт изделия и применимость развертывания, а не обещание универсальной емкости.

Таблица 23 - Транспортные лимиты и бюджеты реализации

limit name	**Единица**	**Значение**
maxXmlBytes	bytes	1024
heartbeatIntervalMs	ms	5000
controlLeaseMs	ms	30000
frameAssemblyMs	ms	10000
writeProgressMs	ms	10000
ordinaryResponseMs	ms	500
cancelAdmissionMs	ms	100
snapshotLifetimeMs	ms	30000
maxSnapshots	count	1..18446744073709551615
maxSnapshotsPerPrincipal	count	1..18446744073709551615
snapshotMaxPages	count	1..18446744073709551615
snapshotReadMaxMs	ms	1..27000

Окончание таблицы 23

limit name	Единица	Значение
eventCapacity	count	1..18446744073709551615
eventRateMax	events/s	1..18446744073709551615
eventBurstMax	count	0..18446744073709551615
eventDrainMinPerSec	events/s	1..18446744073709551615
stopBeginMs	ms	1..1000
stopConfirmMs	ms	1..10000
prepareTimeoutMs	ms	1..18446744073709551615
recoveryTimeoutMs	ms	1..18446744073709551615
resultRetentionMs	ms	30000..18446744073709551615
cmdSeqWindow	count	4096
technicalDurationMaxMs	ms	30000
exerciseDurationMaxMs	ms	1080000
snapshotReplace	bool	0..1
eventsPush	bool	0..1

19.75 stopBeginMs <= stopConfirmMs; maxSnapshotsPerPrincipal <= maxSnapshots. Квота учитывает все незавершенные снимки аутентифицированного участника, а не только одно соединение. При исчерпании квоты новый снимок получает busy; уже выданный живет 30000 мс от фиксации первой страницы без продления чтением. В пределах одной загрузки результат сеанса доступен не менее resultRetentionMs от finished независимо от срока конкретного снимка; активный сеанс не вытесняется. Новый session.start не сокращает эту гарантию и НЕ ДОЛЖЕН ожидать освобождения истории в пределах заявленной нагрузки. Емкость истории и последовательные переходы нормируются отдельно ниже. Гарантии постоянного журнала и чтения после reboot различаются по разделу 8. Кеш решений cmdSeq - отдельное хранилище.

19.76 Дополнительная возможность snapshotReplace объявляется как limit name=snapshotReplace value=1. Только при ее наличии АРМ МОЖЕТ передать selection replaces=<snapshotId> в первом запросе нового device.state, sensors.read или session.get. Это явное подтверждение, что прежний снимок больше не нужен читателю. replaces НЕ совмещается с snapshotId/cursor и не повторяется в продолжениях. Прежний снимок ДОЛЖЕН принадлежать тому же аутентифицированному участнику, режиму защиты, команде и нормализованной выборке, включая controlEpoch; несовпадение возвращает invalidArgument без освобождения снимка. После проверки УУ освобождает указанное место перед выделением нового, в том числе при полной квоте. Неизвестный или уже истекший идентификатор не освобождает чужих мест и обрабатывается как новое чтение. УУ сохраняет связь request id замены с ее результатом на срок жизни нового снимка. Точный повтор того же request id, выборки и replaces возвращает первоначальный новый snapshotId без еще одного освобождения или выделения; изменение содержимого при том же request id дает conflict. Поэтому потеря ответа не приводит к busy и не требует использовать освобожденный курсор. Без replaces каждый снимок сохраняет гарантированные 30000 мс; незавершенные чтения других наблюдателей не вытесняются. Результат исполнения и журнал событий этим подтверждением не удаляются.

19.77 Фоновое наблюдение датчиков использует начальный снимок с границей eventSeq. При объявленном eventsPush=1 АРМ включает инициативную доставку events.subscribe; без нее сохраняется events.read. Частое пустое чтение журнала при работающей подписке не требуется. Изменение resource.changed или качества требует чтения актуальных точек, если событие не содержит достаточных показаний. Начальная синхронизация и восстановление после gap обновляют индикацию, но не создают новый физический фронт для автоматического запуска сценария. Фоновые чтения ДОЛЖНЫ уступать командам управле-

ния между ответами; session.stop и session.cancel сохраняют отдельный слот. Частота наблюдения и выделения снимков ДОЛЖНА укладываться в объявленные квоты, включая одновременное чтение результата речи. При отсутствии snapshotReplace клиент ограничивает фоновые чтения, не расходуя резерв управления.

История результатов и последовательные этапы

19.78 Емкость истории

УУ ДОЛЖЕН указать число одновременно сохраняемых полных результатов и срок хранения resultRetentionMs. До принятия session.start резервируется место его будущего результата. Снимки, кеш cmdSeq и постоянный журнал не заменяют эту емкость.

19.79 Нагрузка последовательных сеансов

Паспорт ДОЛЖЕН задать верхнюю границу числа завершений в любом скользящем окне длиной resultRetentionMs, с учетом всплесков, всех режимов, владельцев и смен эпох. Средней частоты без ограничения всплеска недостаточно. Емкость ДОЛЖНА покрывать эту границу и отдельно результат каждого одновременно принимаемого исполнения; в ядре такое исполнение одно. Если фактическое хранение дольше resultRetentionMs, невытесняемая часть истории также включается в расчет.

19.80 Решение по следующему session.start

Когда полномочия действуют, off подтвержден, обязательные проверки пройдены и нагрузка не превышена, занятая история сама по себе не допускает busy или ожидание истечения resultRetentionMs. Решение и постановка ответа на отправку укладываются в ordinaryResponseMs. Это еще не подтверждение начала звука.

19.81 Время перехода между этапами

Паспорт развертывания ДОЛЖЕН задать конечный допустимый интервал от подтверждения off предыдущего этапа до running следующего

для каждого поддержанного перехода. Он включает получение результата и решение АРМ, разрешенную паузу программы, доставку следующего session.start, ordinaryResponseMs и подготовку до prepareTimeoutMs. Предел выбирается по требованиям сценария и подтверждается испытанием при заявленных канале и нагрузке; срок хранения истории не является допустимой добавочной паузой.

19.82 Превышение заявленной нагрузки

Если безопасно зарезервировать результат нельзя, session.start получает busy до принятия намерения и воздействия. УУ не сокращает гарантированное хранение и не вытесняет активный результат. Повтор того же cmdSeq возвращает прежнее решение; новая попытка требует нового номера и проверки состояния. АРМ учитывает отказ как нарушение сроков сценария, а не скрывает его неограниченным ожиданием.

19.83 Пример расчета емкости: при гарантированном пределе 60 завершений в любом окне 30000 мс, resultRetentionMs=30000 и одном принимаемом исполнении ТРЕБУЕТСЯ не менее 61 места. Число включает резерв результата уже принятого сеанса; это расчетный пример, а не обязательный размер для всех изделий. Параметры этой таблицы входят в опубликованный паспорт изделия и развертывания; новые XML-атрибуты limits редакцией документа не вводятся.

Машинный словарь

19.84 profile

Значения: bootstrap/1 / open-alert-draft/0.21

Смысл: profile

19.85 bool

Значения: true / false

Смысл: bool

19.86 mode

Значения: siren / recording / test / program / stream

Смысл: mode

19.87 purpose

Значения: real / exercise / technical

Смысл: purpose

19.88 policy

Значения: stop / finish

Смысл: policy

19.89 phase

Значения: starting / running / stopping / finished

Смысл: phase

19.90 output

Значения: off / commandedOff / onObserved / unknown / fault

Смысл: output

19.91 result

Значения: succeeded / failed / cancelled / unknown

Смысл: result

19.92 ownershipState

Значения: free / held / detached / draining / blocked / released

Смысл: Состояние владения

19.93 recoveryState

Значения: reconciling / ready / localInterventionRequired

Смысл: Восстановление после сбоя

19.94 checkStatus

Значения: passed / failed / inconclusive / skipped / pending

Смысл: Итог проверки; pending только для незавершенного сеанса. Все статусы кроме passed требуют reason.

19.95 presence

Значения: present / absent / expectedMissing / unknown

Смысл: Физическое наличие точки

19.96 readMethod

Значения: cached / refreshable / checkOnly

Смысл: Способ получения показания

19.97 sourceKind

Значения: localInput / ouReply / controllerMetric / derived

Смысл: Источник данных

19.98 validity

Значения: valid / invalid / unavailable

Смысл: Достоверность значения

19.99 freshness

Значения: fresh / stale / never

Смысл: Свежесть наблюдения

19.100 facet

Значения: identity / meaning / source / acquisition

Смысл: Часть описания точки

19.101 pointType

Значения: bool / enum / int / counter

Смысл: Тип значения точки

19.102 support

Значения: supported / unsupported

Смысл: Поддержка точки, отдельно от ее наличия

19.103 alarmState

Значения: active / cleared / unknown

Смысл: Состояние аварийного эпизода

19.104 severity

Значения: info / warning / critical

Смысл: Категория важности аварии

19.105 continuity

Значения: continuous / unknown

Смысл: Подтвержденная непрерывность эпизода

19.106 thresholdOperator

Значения: lt / le / gt / ge / eq / ne

Смысл: Сравнение значения с порогом

19.107 conflictDetail

Значения: requestMismatch / configRevisionMismatch / capabilityRevisionMismatch / bindingChanged / contentRevisionMismatch / instanceMismatch / operationStateMismatch

Смысл: Машинная причина conflict

19.108 errorCode

Значения: incompatibleProfile / staleAttachment / unsupportedVersion / unsupported / invalidArgument / forbidden / staleBoot / staleEpoch / busy / unsafeState / startWindowExpired / confirmationWindowExpired / startBlockedByCancel / conflict / resultExpired / notFound / snapshotExpired / cursorExpired / sourceUnavailable / storageFault / incompatible / incomplete / noSpace / invalidSignature / rollbackDenied

Смысл: Коды отказа приема; причины результата не являются кодами отказа

19.109 digestAlgorithm

Значения: streebog256

Смысл: Стрибог-256 для содержимого и образов; 32 байта, 64 строчных hex-символа

19.110 evidence

Значения: none / commandSent / reportedMode / electrical / acoustic / localConfirmation

Смысл: Уровень доказательства: нет, отправленная команда, режим ОУ, электрическое/акустическое наблюдение, местное подтверждение. off запрещен для none/commandSent.

19.111 eventType

Значения: session.started / session.finished / resource.changed / sensor.qualityChanged / inventory.changed / alarm.raised / alarm.cleared / alarm.qualityChanged / session.execution / session.fault

Смысл: События ядра; обязательное тело зависит от типа. poll.finished задается отдельным профилем.

19.112 observationPhase

Значения: baseline / running / stopped

Смысл: До воздействия, при воздействии, после остановки.

19.113 limitName

Значения: maxXmlBytes / heartbeatIntervalMs / controlLeaseMs / frameAssemblyMs / writeProgressMs / ordinaryResponseMs / cancelAdmissionMs / snapshotLifetimeMs / maxSnapshots / maxSnapshotsPerPrincipal / snapshotMaxPages / snapshotReadMaxMs / eventCapacity / eventRateMax / eventBurstMax / eventDrainMinPerSec / stopBeginMs / stopConfirmMs / prepareTimeoutMs / recoveryTimeoutMs / resultRetentionMs / cmdSeqWindow / technicalDurationMaxMs / exerciseDurationMaxMs / snapshotReplace / eventsPush

Смысл: Лимиты capabilities; snapshotReplace и eventsPush необязательны. Значения и единицы заданы паспортом ядра. Неизвестные имена требуют другого согласованного профиля.

19.114 resultReason

Значения: completed / operatorStop / controlLostStop / maxDurationExceeded / prepareTimeout / stopDispatchTimeout / stopConfirmTimeout / sourceUnavailable / driverFault / storageFault / contentMismatch / mediaIncomplete / hardwareProtection / testFailed / stopUnconfirmed / observationLost / recoveryUncertain / mediaClockFault / equipmentFault / programAborted

Смысл: Закрытый словарь причин. programAborted допустим только у stageResult со state=notStarted и без result; у session и finished-этапа запрещен. Остальные пары result/reason заданы временной моделью.

19.115 ownershipPurpose

Значения: real / exercise / technical / maintenance

Смысл: Назначение эпохи; START совпадает с ним, maintenance не допускает START.

19.116 executionStatus

Значения: pending / confirmed / notExecuted / unknown

Смысл: Историческое подтверждение начала воздействия; confirmed не стирается остановкой или поздним отказом.

19.117 completionState

Значения: pending / complete / interrupted / unknown

Смысл: Полнота принятого задания отдельно от подтверждения начала.

19.118 stopMode

Значения: duration / command

Смысл: Сирена заданной длительности либо до штатной адресной остановки; maxDurationMs обязателен в обоих случаях.

19.119 forbiddenDetail

Значения: roleDenied / scopeDenied / ownerMismatch

Смысл: Машинная причина отказа по полномочиям; сведения о чужом исполнении не раскрываются.

19.120 busyDetail

Значения: ownershipHeld / executionActive / localControl / serviceActive / capacityExhausted

Смысл: Причина занятости; START не ставится в очередь.

19.121 faultEffect

Значения: none / degraded / failed / unknown

Смысл: Влияние диагностического условия на выбранное исполнение, независимо от важности severity.

19.122 faultCode

Значения: powerLost / phaseLost / batteryLow / lineOpen / lineShort / amplifierFault / ouLinkLost / outputMismatch / sourceLost / overTemperature / hardwareProtection / storageFault / mediaIncomplete / mediaPacketLoss / mediaClockFault / driverFault / sourceUnavailable / contentMismatch / maxDurationExceeded / prepareTimeout / stopDispatchTimeout / stopConfirmTimeout / testFailed / observationLost / vendorFault

Смысл: Код нарушения во время исполнения; конкретная область и влияние передаются отдельно.

19.123 responseStatus

Значения: ok / accepted / error

Смысл: responseStatus

19.124 testEffect

Значения: none / readBus / switchOutput / audible / pauseAudio / resetInterface / writeStorage

Смысл: testEffect

Записи подтверждения и полноты исполнения

19.125 result

Дополнительно executionCount=0..512 и faultCount=0..65535; неизменны на всех страницах. Ноль разрешен только для program без звучащих этапов; в остальных случаях executionCount=1..512. Вместе с остальными полями и cursor не более 12 атрибутов.

19.126 execution

resourceId, status, criterionId обязательны; stageIndex/passIndex только для program. При confirmed обязательны sourceId, evidence и observedAtMs; evidence не none/commandSent. При другом статусе evidence=none, sourceId/observedAtMs отсутствуют. Ключ — resourceId/

stageIndex/passIndex, уникальный в снимке; resourceId принадлежит session.start.

19.127 completion

Ровно одна запись в полном session.get: state=pending|complete|interrupted|unknown. pending только до finished; complete означает достижение естественной границы всего сигнала, файла, потока, теста или программы, interrupted — подтвержденное прекращение до нее, unknown — полнота не доказана. Отдельно от execution и result; при operatorStop незавершенный файл имеет interrupted даже при succeeded. Для siren stopMode=command штатный session.stop после работы либо подтвержденный конец собственного цикла речевого ОУ с signalKind=builtIn означает complete; при duration остановка до durationMs означает interrupted.

19.128 fault

id (32 hex), code, resourceId, effect, severity, occurredAtMs обязательны; conditionId и alarmId при каталожном условии, clearedAtMs только при подтвержденном снятии; stageIndex/passIndex только для program. Код берется из faultCode. Для внутреннего отказа контроллера resourceId=controller; это специальная область, не дополнительная цель session.start. ID уникален; записи сохраняются до конца хранения результата.

19.129 executionCriterion в profiles.get

resourceId, mode, criterionId, evidence, sourceId, maxAgeMs обязательны. Ровно один критерий для каждого поддержанного сочетания ресурс/режим; он связан с capabilityRevision/configRevision и описанием профиля. Evidence none ДОПУСКАЕТСЯ только при невозможности подтвердить начало и не разрешает confirmed. Условия для конкретной записи/потока, источника и свежести ДОЛЖНЫ быть опубликованы; критерий активного сеанса неизменен.

19.130 События session.execution / session.fault

Ровно одно тело execution / fault с дополнительным sessionId (32 hex). В session.get sessionId берется из единственной сводки и в этих записях не повторяется. Событие execution публикуется при первом confirmed; оно не означает finished.

19.131 Полнота session.get требует точных множеств execution согласно session.start/плану, одной completion, точных executionCount/ faultCount и уникальных ключей. Положительный итог не выводится по первой странице. confirmed связывается с принятой ревизией, конкретным ресурсом и проходом; acceptedAtMs <= observedAtMs <= evaluatedAtMs, наблюдение относится к фазе работы данного прохода до stoppingAtMs. Историческая свежесть оценивается в момент подтверждения, а не чтения. Все необходимые доказательства off проверяются отдельно по resource и timeline.

19.132 Сирена поддерживает два явно объявленных режима длительности: stopMode=duration с обязательным durationMs=100..3600000 либо stopMode=command без durationMs. Оба требуют конечного maxDurationMs. Отсутствие stopMode недопустимо. При command штатный конец задает session.stop либо подтвержденный конец собственного цикла речевого ОУ с signalKind=builtIn по требованиям аппаратного профиля сигнала; maxDurationExceeded остается защитным отказом. Для command применяется onControlLoss=stop независимо от настройки finish. В автономной program сирена всегда использует stopMode=duration; command внутри программы запрещен. Разрешенные stopMode объявляются параметрами profiles.get; неподдержанный вариант получает unsupported до воздействия.

Таблица 24 - Записи подтверждения и полноты исполнения

Уровень результата	**Закрытые значения и смысл**
Оболочка ответа УУ и node-alert/2	ok — завершено синхронное действие; accepted — принято длительное действие; error — отказ команды. Принятие не доказывает физическое исполнение.
Допуск цели admission	pending|accepted|rejected|unknown — отдельная ось, а не второй словарь статусов оболочки.
Ошибки bulk/1	Коды 1..8 являются компактным кодированием имен общего словаря; соответствие bulkErrorCodes задано vocabulary и проверяется вместе с бинарными векторами. Значения не переиспользуются.

20 Семейство профилей и совместимость

20.1 Протокол задает общую предметную модель и несколько профилей: АРМ-АРМ, АРМ-УУ, драйвер ОУ, медиа и обслуживание. Состояния, происхождение запуска и смысл результата едины; способы доставки и ресурсные пределы могут различаться. Устройство не обязано поддерживать все профили.

Таблица 25 - Семейство профилей и совместимость

Паспорт профиля	**Обязательное содержание**
Идентичность и версии	profileId, версия схемы, зависимости от других профилей, правило совместимости и необходимые расширения.
Предметный контракт	Обязательные команды, состояния, ошибки, роли, ресурсы и критерии фактического результата.
Пределы	Байты сообщения, страницы, глубина, число контекстов, очереди, длительность операций и квоты. Ограничения имеют измеримое основание.
Время канала	Период heartbeat, порог потери владельца, таймер неполного кадра, таймер передачи, бюджет session.stop и условия восстановления. Учитываются RTT, потери и нагрузка.
Доказательство совместимости	Позитивные и негативные векторы, сценарии отказа, аппаратные ревизии и результат испытаний.

20.2 После установления защищенного контекста стороны выбирают точный общий профиль с фиксированными лимитами. Если общего профиля нет, управление не начинается. Лимиты конкретного изделия читаются через capabilities и проверяются на применимость; hello не согласует их произвольное уменьшение. Нельзя молча сменить режим, отключить защиту или использовать неизвестное обязательное расширение. Согласование выполняется через protocol.hello профиля bootstrap/1 по разделу 19.

20.3 Capabilities описывают поддержанные функции, согласованный профиль - разрешенный договор обмена, а readiness - возможность выполнить конкретный план сейчас. Эти три понятия различаются. Смена capabilityRevision или привязки требует повторной проверки применимости плана.

Версии и совместимость

20.4 v="1" обозначает оболочку сообщения и сохраняется независимо от номера веб-редакции. profile определяет точную схему и семантику; стороны ДОЛЖНЫ отвергать несовместимый профиль до воздействия. Примеры этой редакции используют open-alert-draft/0.21. Изменение схемы или обязательного поведения получает новый идентификатор профиля. Реализация ДОЛЖНА соответствовать выбранному контракту целиком.

20.5 Матрица каталога различает обязательные команды ядра и дополнительные расширения. Если расширение объявлено, реализация ДОЛЖНА выполнять весь его контракт и зависимости: например, tests.list для mode=test, семейство poll.start/get/cancel для заданий опроса и config.stage/commit/cancel для транзакционной настройки. Отклонение неподдержанного режима является штатным результатом поддержанной команды session.start.

20.6 Для независимых производителей публикуются обязательное ядро и отдельные опциональные профили, реестр имен/ошибок, схемы, тестовые векторы и правила неизвестных полей. Обязательные расширения согласуются до исполнения. Необязательные расширения допускаются только в определенных схемой местах; неизвестный элемент не становится произвольной командой. Таблица совместимости включает разные АРМ и драйверы ОУ, ограниченные функции, обновление версий и отрицательные случаи. Закрытые особенности конкретного АРМ не являются условием базовой совместимости.

Соединение и проверка устройства

20.7 Для базового постоянного TCP-канала запросы, ответы и события идут в обе стороны по этому соединению; отдельный обратный сокет НЕ ТРЕБУЕТСЯ самим фактом отправки события. Очереди и корреляция направлений остаются ограниченными. Последовательность команд одного УУ не запрещает АРМ параллельно обслуживать другие приборы.

20.8 АРМ ДОЛЖЕН проверить описание устройства, его функциональный тип и объявленные возможности до запуска. Типы ВАУ и Сирена задают назначение ОУ; разрешение конкретного режима определяется capabilities и полномочиями участника. Один и тот же ОУ имеет одного действующего владельца управления.

20.9 busy является штатным отказом при занятости другим сеансом или владельцем. Получатель ответа сохраняет действующее исполнение; новый запрос запуска не становится отложенным заданием, которое включит ОУ после освобождения.

Изменение обязательного поведения

20.10 Идентификатор управляющего контракта этой редакции - open-alert-draft/0.21. Любое изменение обязательного поведения, диапазона, значения по умолчанию, правил дедупликации, полномочий, таймеров или достаточности результата требует нового идентификатора профиля, даже при неизменном XML. Редакция текста при прежнем профиле может исправлять опечатки, ссылки и пояснения, не меняющие множество допустимых трасс. Выбор через protocol.hello фиксирует весь контракт; автоматическое смешивание правил разных профилей запрещено.

21 Логические цели, реестр и маршрутизация

Таблица 26 - Логические цели, реестр и маршрутизация

Сущность	Смысл
targetId	Непрозрачный постоянный 128-битный ID логической точки оповещения в АРМ. Регион и муниципалитет не зашиваются в его байты.
deviceId	Идентичность физического УУ; в tls подтверждается ключом, в plain криптографически не подтверждена. Замена прибора меняет эту идентичность.
Привязка	targetId -> ответственный nodeId, deviceId, ouId/resourceId, bindingGeneration и registryRevision.
Локальный locator	Область сети, транспорт, адрес и порт. Это способ добраться до прибора, а не основание его полномочий.
Маршрут между АРМ	Разрешенные соседние узлы, версия пути, hopLimit и исходный срок приема. Происхождение поручения сохраняется.

21.1 Два одинаковых частных IP в разных областях сети допустимы. Команды и результаты различаются по логической цели, идентичности прибора и области маршрутизации. Два несовместимых назначения одного targetId или одного физического ресурса образуют конфликт, который выявляется до автоматического применения.

21.2 Вышестоящий АРМ адресует разрешенную цель; нижестоящий разрешает ее через свой реестр. УУ проверяет локальную ссылку ouId/resourceId и поколение привязки. Полный реестр всей системы в малом контроллере не хранится. Перенос цели на замененный УУ выполняется явно, с отзывом старого назначения, без переноса активного сеанса.

Если старое ОУ не умеет удостоверять собственный ID, identityEvidence=commissionedBinding означает подтвержденную при наладке привязку. Криптографическая идентичность Ethernet-части не доказывает правильность подключения исполнительного тракта или идентичность неаутентифицированного усилителя.

21.3 Циклы маршрутов запрещены; hopLimit уменьшается на каждом переходе и не превышает 16. Межузловое поручение содержит deadlineUtcMs и clockErrorMs; uint64 передаются десятичными строками. Каждый узел допускает поручение только при upperBoundUtcMs < deadlineUtcMs, где upperBoundUtcMs - его текущее UTC плюс установленная верхняя погрешность не более 500 мс. Неизвестная погрешность либо превышение 500 мс дает clockUncertain без воздействия. Посредник сохраняет исходный deadlineUtcMs и не выдает новый полный TTL. АРМ переводит оставшийся гарантированный срок в acceptBeforeMs шкалы УУ с учетом RTT, погрешности и прошедшего локального времени.

Межузловой формат node-alert/2

21.4 АРМ—АРМ использует отдельное TLS 1.3-соединение с криптографическим профилем раздела 14, индивидуальным PSK пары узлов и заранее разрешенными nodeId. Адрес, port и networkScope задаются реестром при вводе; multicast и открытый запасной канал запрещены. Рамка uint32 BE длины и UTF-8 JSON: 1..65536 байт, глубина до 12, не более 4096 узлов JSON, повтор ключа запрещен. Тело соответствует node-schema-0.21.json, неизвестные поля отвергаются. Первый запрос node.hello; hello.nodeId сверяется с TLS principalId, ответ hello фиксирует node-alert/2. Срок handshake 20000 мс от приема TCP; сборка/отправка кадра — 20000 мс при нижнем ориентире канала 64 кбит/с. Кадр максимальной длины 65536 байт занимает 8192 мс без транспортных расходов и не подчиняется пределу 1024 байта конечного УУ; предел обра-

ботки обычного запроса 500 мс, длительное исполнение возвращает accepted. Чтение не поддерживает владение УУ.

21.5 В таблицах каждое поле занимает отдельную строку. Знак ? после имени обозначает необязательное поле; остальные поля обязательны. Варианты объекта приведены отдельно. Целые в строке передаются в канонической десятичной записи; обе границы диапазона включены. Предел в байтах UTF-8 применяется к декодированному значению; непарные суррогаты ЗАПРЕЩЕНЫ. Для массива указаны минимальное и максимальное количество элементов и тип элемента. Дополнительные межполевые условия приведены в правилах соответствующего профиля.

Таблица 27 - Тело запроса node.hello

Поле	**Тип и допустимые значения**
nodeId	Строка: 32 символа [0-9a-f]
profile	Точное значение: "node-alert/2"

Таблица 28 - Тело запроса run.start

Поле	Тип и допустимые значения
originNodeId	Строка: 32 символа [0-9a-f]
originSeq	Целое в строке: 1..18446744073709551615
runId	Строка: 32 символа [0-9a-f]
branchId	Строка: 32 символа [0-9a-f]
deadlineUtcMs	Целое в строке: 0..18446744073709551615
clockErrorMs	Целое в строке: 0..500
hopLimit	Целое в строке: 1..16
path	Массив 1..16: Строка: 32 символа [0-9a-f]; без повторов
purpose	Строка; одно из: real, exercise
planId	Строка: 1..32 символа [A-Za-z0-9_.:-]
planRevision	Целое в строке: 1..18446744073709551615
maxDurationMs	Целое в строке: 100..86400000
targetDigest	Строка: 64 символа [0-9a-f]
targets	Массив 1..256: selectionTarget

Таблица 29 - Тело запроса run.cancel

Поле	Тип и допустимые значения
originNodeId	Строка: 32 символа [0-9a-f]
originSeq	Целое в строке: 1..18446744073709551615
runId	Строка: 32 символа [0-9a-f]
branchId	Строка: 32 символа [0-9a-f]
deadlineUtcMs	Целое в строке: 0..18446744073709551615
clockErrorMs	Целое в строке: 0..500
hopLimit	Целое в строке: 1..16
path	Массив 1..16: Строка: 32 символа [0-9a-f]; без повторов

Таблица 30 - Тело запроса run.get — вариант 1

Поле	Тип и допустимые значения
originNodeId	Строка: 32 символа [0-9a-f]
originSeq	Целое в строке: 1..18446744073709551615
runId	Строка: 32 символа [0-9a-f]
branchId	Строка: 32 символа [0-9a-f]

Таблица 31 - Тело запроса run.get — вариант 2

Поле	Тип и допустимые значения
originNodeId	Строка: 32 символа [0-9a-f]
originSeq	Целое в строке: 1..18446744073709551615
runId	Строка: 32 символа [0-9a-f]
branchId	Строка: 32 символа [0-9a-f]
snapshotId	Строка: 32 символа [0-9a-f]
cursor	Строка: 1..32 символа [A-Za-z0-9_.:-]

Таблица 32 - Тело запроса result.read — вариант 1

Поле	Тип и допустимые значения
originNodeId	Строка: 32 символа [0-9a-f]
originSeq	Целое в строке: 1..18446744073709551615
runId	Строка: 32 символа [0-9a-f]
branchId	Строка: 32 символа [0-9a-f]
targetId	Строка: 32 символа [0-9a-f]

Таблица 33 - Тело запроса result.read — вариант 2

Поле	Тип и допустимые значения
originNodeId	Строка: 32 символа [0-9a-f]
originSeq	Целое в строке: 1..18446744073709551615
runId	Строка: 32 символа [0-9a-f]
branchId	Строка: 32 символа [0-9a-f]
targetId	Строка: 32 символа [0-9a-f]
snapshotId	Строка: 32 символа [0-9a-f]
cursor	Строка: 1..32 символа [A-Za-z0-9_.:-]

Таблица 34 - Тело запроса catalog.get — вариант 1

Поле	Тип и допустимые значения
—	Пустой объект {}

Таблица 35 - Тело запроса catalog.get — вариант 2

Поле	Тип и допустимые значения
snapshotId	Строка: 32 символа [0-9a-f]
cursor	Строка: 1..32 символа [A-Za-z0-9_.:-]

Таблица 36 - Тело запроса observations.get

Поле	Тип и допустимые значения
targetId	Строка: 32 символа [0-9a-f]
pointIds	Массив 1..8: Строка: 1..32 символа [A-Za-z0-9_.:-]; без повторов
deadlineUtcMs	Целое в строке: 0..18446744073709551615
clockErrorMs	Целое в строке: 0..500
hopLimit	Целое в строке: 1..16
path	Массив 1..16: Строка: 32 символа [0-9a-f]; без повторов

Таблица 37 - Объект ответа hello

Поле	**Тип и допустимые значения**
nodeId	Строка: 32 символа [0-9a-f]
selectedProfile	Точное значение: "node-alert/2"
maxBytes	Точное значение: "65536"

Таблица 38 - Объект ответа admission

Поле	**Тип и допустимые значения**
originNodeId	Строка: 32 символа [0-9a-f]
originSeq	Целое в строке: 1..18446744073709551615
runId	Строка: 32 символа [0-9a-f]
branchId	Строка: 32 символа [0-9a-f]
state	Строка; одно из: accepted, cancelled
targetCount	Целое в строке: 0..256

Таблица 39 - Объект ответа targetResult

Поле	**Тип и допустимые значения**
targetId	Строка: 32 символа [0-9a-f]
admission	Строка; одно из: pending, accepted, rejected, unknown
state	Строка; одно из: pending, starting, running, stopping, finished, notStarted
result	Строка; одно из: none, succeeded, failed, cancelled, unknown
reason	Строка: 1..32 символа [A-Za-z0-9_.:-]
execution	Строка; одно из: pending, confirmed, partial, notExecuted, unknown
completion	Строка; одно из: pending, complete, interrupted, unknown
sourceNodeId	Строка: 32 символа [0-9a-f]
deviceId	Строка: 32 символа [0-9a-f]
bootId?	Строка: 32 символа [0-9a-f]
sessionId?	Строка: 32 символа [0-9a-f]

Таблица 40 - Объект ответа runResult

Поле	**Тип и допустимые значения**
originNodeId	Строка: 32 символа [0-9a-f]
originSeq	Целое в строке: 1..18446744073709551615
runId	Строка: 32 символа [0-9a-f]
branchId	Строка: 32 символа [0-9a-f]
snapshotId	Строка: 32 символа [0-9a-f]
evaluatedAtUtcMs	Целое в строке: 0..18446744073709551615
more	Логическое значение: true или false
cursor?	Строка: 1..32 символа [A-Za-z0-9_.:-]
targetCount	Целое в строке: 0..256
state	Строка; одно из: pending, running, finished
targets	Массив 0..32: targetResult

Таблица 41 - Объект ответа rawResult

Поле	**Тип и допустимые значения**
originNodeId	Строка: 32 символа [0-9a-f]
originSeq	Целое в строке: 1..18446744073709551615
runId	Строка: 32 символа [0-9a-f]
branchId	Строка: 32 символа [0-9a-f]
targetId	Строка: 32 символа [0-9a-f]
sourceNodeId	Строка: 32 символа [0-9a-f]
snapshotId	Строка: 32 символа [0-9a-f]
more	Логическое значение: true или false
cursor?	Строка: 1..32 символа [A-Za-z0-9_.:-]
requestXml	Строка: не более 1024 символов
responseXml	Строка: не более 1024 символов

Таблица 42 - Объект ответа catalog

Поле	**Тип и допустимые значения**
snapshotId	Строка: 32 символа [0-9a-f]
more	Логическое значение: true или false
cursor?	Строка: 1..32 символа [A-Za-z0-9_.:-]
revision	Целое в строке: 1..18446744073709551615
targetCount	Целое в строке: 0..256
groupCount	Целое в строке: 0..128
programCount	Целое в строке: 0..128
targets	Массив 0..32: target
groups	Массив 0..32: group
programs	Массив 0..32: program

Таблица 43 - Объект ответа observation

Поле	**Тип и допустимые значения**
id	Строка: 1..32 символа [A-Za-z0-9_.:-]
type	Строка; одно из: bool, enum, int, counter
value?	Строка: не более 128 байт UTF-8
validity	Строка; одно из: valid, invalid, unavailable
freshness	Строка; одно из: fresh, stale, never
reason?	Строка: 1..32 символа [A-Za-z0-9_.:-]
sourceId	Строка: 1..32 символа [A-Za-z0-9_.:-]
observedAtMs	Целое в строке: 0..18446744073709551615
evaluatedAtMs	Целое в строке: 0..18446744073709551615
maxAgeMs	Целое в строке: 0..18446744073709551615
validUntilUtcMs?	Целое в строке: 0..18446744073709551615

Таблица 44 - Объект ответа observations

Поле	**Тип и допустимые значения**
targetId	Строка: 32 символа [0-9a-f]
sourceNodeId	Строка: 32 символа [0-9a-f]
bootId	Строка: 32 символа [0-9a-f]
capabilityRevision	Целое в строке: 0..18446744073709551615
bindingGeneration	Целое в строке: 1..18446744073709551615
points	Массив 1..8: observation

Таблица 45 - Объект ответа error — code="forbidden"

Поле	**Тип и допустимые значения**
code	Точное значение: "forbidden"
detail	Строка; одно из: roleDenied, scopeDenied, ownerMismatch

Таблица 46 - Объект ответа error — code="busy"

Поле	**Тип и допустимые значения**
code	Точное значение: "busy"
detail	Строка; одно из: ownershipHeld, executionActive, localControl, serviceActive, capacityExhausted

Таблица 47 - Объект ответа error — code="conflict"

Поле	**Тип и допустимые значения**
code	Точное значение: "conflict"
detail	Строка; одно из: requestMismatch, configRevisionMismatch, capabilityRevisionMismatch, bindingChanged, contentRevisionMismatch, instanceMismatch, operationStateMismatch

Таблица 48 - Объект ответа error — вариант 4

Поле	Тип и допустимые значения
code	Строка; одно из: invalidArgument, unsupported, notFound, resultExpired, snapshotExpired, clockUncertain, deadlineExpired, routeLoop, sourceUnavailable, storageFault, startBlockedByCancel

21.6 Оболочка запроса: v="1", profile="node-alert/2", id (32 hex), action и body. Оболочка ответа: v/profile/to/status/kind/body; to равен id. kind определяет запись из таблицы; error имеет status=error, admission - accepted, остальные - ok. node.hello отвечает hello, run.start/run.cancel - admission, run.get - runResult, result.read - rawResult, catalog.get - catalog, observations.get - observations. Каждое чтение использует новый id; повтор изменения с тем же смысловым ключом допускает новый транспортный id. Один обычный запрос без ответа на соединении; run.cancel передается по отдельному TLS-соединению той же пары с собственным hello и не ожидает очереди обычного канала. Предел такого соединения - одно на пару; оно допускает только hello и cancel.

21.7 run.start принимает только неизменную локальную программу planId/planRevision и purpose=real|exercise; поток микрофона и технические тесты не транслируются этим профилем. maxDurationMs:100..86400000; для exercise не более 1080000 с выполнением правил допуска длительных учений. targets фиксируют точный набор и bindingGeneration, без дубликатов, сортируются по targetId; targetDigest - Стрибог-256 от JCS этого массива. Группа раскрывается отправителем в targets до команды; изменение группы не меняет принятую ветвь. Все цели одной ветви относятся к одному конечному ownerNodeId и одному маршруту; несколько конечных узлов получают разные branchId/originSeq одного runId. Посредник пересылает ветвь це-

ликом, не создает скрытых целей и не меняет planRevision/deadline. Конечный АРМ сохраняет ветвь перед session.start УУ и сопоставляет каждому target его собственные bootId/controlEpoch/startSeq/sessionId.

21.8 originNodeId задает уполномоченного автора; значение проверяется по локальному разрешению пары TLS principalId/originNodeId/область целей/действие на каждом переходе. Это доверие к уполномоченным посредникам, не сквозная цифровая подпись origin. path начинается с originNodeId, заканчивается отправляющим узлом и не содержит получателя. При пересылке узел добавляет свой ID, уменьшает hopLimit на 1; прием требует hopLimit:1..16, clockErrorMs:0..500 и UTC с известной погрешностью <=500 мс. deadlineUtcMs ДОЛЖЕН быть позже локального верхнего UTC и не дальше 86400000 мс; уменьшение hopLimit до 0 запрещает следующий переход. Цикл дает routeLoop. Смена пути после accepted не разрешает запуск той же цели новой ветвью: неизвестный предыдущий исход сначала выясняется или адресно отменяется.

21.9 originSeq - положительный uint64, сохраняемый автором до отправки, возрастающий для каждой новой ветви; переполнение запрещает новые ветви, но не их отмену. Каждый принимающий узел хранит постоянный highWater для originNodeId и решения активных ветвей. Новый run.start требует originSeq>highWater; выдача accepted, сохранение тела и highWater атомарны до пересылки. Предметный отказ новой ветви после проверки прав и оболочки также сохраняется и расходует номер; при отказе самого постоянного журнала воздействия запрещены. Точный повтор получает сохраненное решение; другое тело при том же номере - conflict. Вытесненное решение при номере <=highWater дает resultExpired, без повторного воздействия. АРМ сериализует прием новых номеров по каждому маршруту; перестановка новых ветвей может дать отказ старому номеру, но не поздний запуск.

21.10 run.cancel использует originSeq отменяемой ветви, а не новый номер. При известной ветви runId/branchId/originNodeId обязаны

совпасть; отмена адресуется только ее исполнениям через session.stop/ session.cancel УУ. При неизвестном номере >highWater сначала атомарно сохраняются highWater и запрет позднего run.start (startBlockedByCancel проверяется прежде сравнения тела session.start); ответ cancelled с targetCount=0 не означает физического выключения. Для номера <=highWater без истории возвращается resultExpired; это не отменяет уже сохраненного запрета повторного запуска. Активная ветвь и ее отмена не вытесняются. Отмену вправе передать только первоначальный непосредственный делегат той же origin либо сам origin по разрешенному прямому маршруту. Повторы отмены не двигают сроки. Срок отмены определяет ее допуск; принятая остановка продолжается после его истечения.

21.11 Решение ветви и полный итог хранятся не менее 86400000 мс после finished; highWater сохраняется без срока и переживает reboot. Утрата или повреждение этого журнала запрещает новые run.start, но не чтение диагностики и восстановление собственных активных ресурсов. Снимки run.get/catalog.get/result.read живут 30000 мс; continuation передает snapshotId/cursor, ответ more=true требует cursor, more=false запрещает его. Каждая непоследняя страница содержит запись; пустой каталог или отмененная до допуска ветвь имеют единственную страницу more=false; сводные числа/метаданные неизменны, каждая ожидаемая цель передается ровно раз. targetCount/groupCount/programCount относятся ко всему снимку. Неизвестная ссылка, чужая цель, повтор записи или утрата страницы не дают частичного общего успеха.

21.12 runResult фиксирует каждую цель из session.start. Обязательные execution и completion сохраняют раздельные факты: confirmed означает подтверждение всех ожидаемых воздействий, partial — лишь части, notExecuted — доказанное отсутствие всех, unknown — недостаток доказательств, pending — ожидание. Для program пауза не подтверждает звучание. completion переносит полноту исходного результата; по-

средник не выводит complete из succeeded/operatorStop. Полные execution/fault и источник доказательств читаются через result.read. До решения по цели обе сводки pending; при отказе до воздействия execution=notExecuted, completion=interrupted. pending имеет result=none; notStarted требует admission=rejected и result=failed, либо admission=unknown/result=unknown при утрате доказательств. starting/running/stopping требуют admission=accepted и result=none. finished требует admission=accepted, sessionId/bootId и терминальный result/reason раздела 8. В неисполненных состояниях sessionId отсутствует. sourceNodeId и deviceId берутся из зафиксированного реестра. Общий state=finished означает отсутствие незавершенных целей, а не общий успех; успех возможен лишь при succeeded каждой ожидаемой цели. reason=none используется только при result=none; причина отказа допуска - код ошибки профиля. Таймаут канала не создает finished/succeeded.

21.13 result.read возвращает исходную пару XML request/response session.get от конкретного УУ, каждую не более 1024 байт UTF-8, с исходными bootId, ссылками, временем и качеством. Внешний snapshotId/cursor принадлежит сохраненной коллекции, внутренний XML не переписывается. Получатель проверяет полный исходный снимок по правилам АРМ—УУ; одно сообщение не доказывает его полноту. catalog.get передает замороженный разрешенный каталог targets/groups/programs схемы конфигурации; отсутствие записи не отзывается частичной страницей. Объявленные квоты ДОЛЖНЫ позволять кодировать одну полную запись и считывать весь снимок за 27000 мс.

21.14 observations.get возвращает ровно запрошенные 1..8 точек целевого УУ одним ответом; неподдержанная цель дает notFound, недоступная точка - unavailable/never/reason с observedAtMs=0 без value. Тип и enum берутся из полного каталога УУ; valid требует value, invalid/unavailable требуют reason и запрещают value. Ответ сохраняет исходные bootId, evaluatedAtMs и observedAtMs. fresh дополнительно требует

validUntilUtcMs: консервативную нижнюю оценку срока свежести, рассчитанную конечным узлом по исходному maxAgeMs, возрасту, задержке доставки и погрешности UTC. Посредники не увеличивают этот срок. При неизвестной границе задержки или UTC свежесть не заявляется. Получатель считает значение fresh только пока его верхнее UTC не превысило validUntilUtcMs. Получение сообщения само по себе не освежает данные.

21.15 XML управления, JSON конфигурации и строка исходного XML в result.read обслуживают разные границы. result.read сохраняет исходное доказательство УУ: JSON-экранирование снимается ровно один раз, полученная UTF-8 строка проверяется как отдельный кадр XML до 1024 байт с обычными ограничениями глубины, элементов и сущностей. XML не исполняется как команда, внешние сущности и DTD запрещены. Для JSON отдельно действуют пределы внешнего кадра и глубины; они не заменяют внутренние. Хеш исходного XML, если используется вне этого профиля, относится к исходным байтам, а не к их JSON-экранированию. Контроллер УУ не обязан реализовывать межузловой JSON-парсер.

22 Готовность плана и проверка записей

22.1 readiness.get отвечает на вопрос о готовности конкретного режима, набора ресурсов и версии контента. Ответ основан на наблюдениях и не включает выходы, не запускает скрытый функциональный тест, не резервирует ресурс и не дает права управления.

Таблица 49 - Готовность плана и проверка записей

Результат	**Правило оценки**
blocked	Есть обязательное препятствие: например, ресурс занят, выключение неизвестно, контент не совпадает или функция не поддерживается.
unknown	Обязательных доказательств недостаточно либо они устарели; отсутствие данных не считается нормой.
degraded	Обязательные условия выполнены, но есть ухудшение, которое явно разрешено данным профилем и политикой.
ready	Все обязательные условия на момент оценки выполнены; допустимых ухудшений нет. Это не гарантия будущего звучания.

22.2 Приоритет сводки: blocked, затем unknown, затем degraded, иначе ready. По каждому ресурсу сохраняются reason, обязательность проверки, источник, возраст и срок применимости доказательства. Отсутствие необязательного акустического датчика отражает глубину наблюдаемости и само по себе не блокирует все запуски.

22.3 Оценка связывается с mode, выбранными ресурсами, contentId/revision/hash, bootId, bindingGeneration, configRevision, capabilityRevision и stateRevision. session.start повторяет быстрые обязательные проверки актуального контекста. Изменение внутреннего интерфейса, ресурса или записи после ready не позволяет использовать устаревшее заключение как разрешение.

Предпусковая оценка не включает звучащий тест. Обязательные текущие доказательства исправности и off имеют возраст не более 1000 мс; решение readiness формируется не более чем за 500 мс. Отсутствие необязательного акустического датчика допускает degraded только при выполнении заявленного обязательного электрического или цифрового критерия. Права, ревизии, подлинность контента

и неизвестное состояние выхода обходить запрещено. Универсальной команды force нет.

Готовность содержимого

22.4 content.describe возвращает манифест записи: contentId, revision, hashAlgorithm/hash, формат, длительность, размер, место хранения/исполнения, состояние проверки и ее возраст. Ожидаемый хеш манифеста и хеш реально проверенных байтов различаются. Номер слота и имя файла не подтверждают содержание.

22.5 Если интерфейс ОУ не позволяет проверить байты встроенной записи, verified не выдается. Фоновая проверка или отдельная сервисная работа ДОЛЖНЫ иметь конечный бюджет; чтение описания не запускает скрыто полное хеширование. Необходимая проверка перед session.start либо уже достоверна, либо дает понятную причину отказа/неопределенности в пределах бюджета. Текст и видео остаются отдельными возможностями и не следуют из типа ВАУ или Сирена.

22.6 readiness.get: запрос plan с mode=siren|recording|program|stream|test, configRevision, capabilityRevision; resources и ровно один элемент режима с теми же полями, что session.start, но без acceptBeforeMs. Для оценки длительности обязателен maxDurationMs в plan. Ответ readiness - неизменный снимок с snapshotId, evaluatedAtMs, validUntilMs=evaluatedAtMs+1000, stateRevision, more/cursor. Первая страница содержит summary state=ready|degraded|blocked|unknown; записи resource содержат id/bindingGeneration/state/reason. Записи evidence содержат resourceId/pointId/required/observedAtMs/validUntilMs/validity. Продолжение: selection snapshotId/cursor, исходный план фиксируется снимком. Оценка не резервирует ресурс.

22.7 content.describe: selection contentId/revision либо snapshotId/cursor. Ответ contents: snapshotId, capabilityRevision, more/cursor. Для каждой записи content передаются id/revision/bytes/sampleCount/codec/hashAlgorithm/hash/location; location=controller|ou. Отдельная запись

verification с contentId/revision, state=verified|unknown|invalid, observedAtMs, sourceId и actualHash при доступных байтах. verified требует совпадения actualHash с hash той же неизменной ревизии; unknown не содержит actualHash. Для локальной программы УУ допускает только verified и удерживает неизменные байты до завершения исполнения.

23 Каскадная диагностика и групповой отчет

23.1 АРМ может запросить наблюдения через несколько подчиненных узлов, имея отдельное право чтения в указанной области targetId. Запрос имеет operationId и срок. Каждый посредник сохраняет источник, bootId, качество и возраст исходного доказательства; время получения посредником не превращает старое наблюдение в fresh.

23.2 Итог оповещения связан с runId/branchId, planRevision и неизменным набором целей с ревизией/хешем состава. Для каждой цели разделяются принятие команды, исполнение и физическое наблюдение. Подтверждение промежуточного узла не означает срабатывания всех его ОУ.

Таблица 50 - Каскадная диагностика и групповой отчет

Текущее состояние цели	**Значение**
pending / active	Ожидание решения / принятое незавершенное исполнение.
succeeded / failed	Установленный успех по объявленному критерию / установленная ошибка. Критерий и доказательство доступны отдельно.
cancelled / notStarted	Подтвержденная отмена исполнения / установленный факт отсутствия старта в данной ветви.
unknown	Доступные данные не позволяют установить исход. Потеря связи не превращается автоматически в failed или notStarted.

23.3 Состояния взаимоисключающие и в сумме покрывают весь зафиксированный набор получателей. Счетчики переходов хранятся отдельно. Повтор той же ревизии отчета не увеличивает итоги; позднее доказательство создает новую ревизию и сохраняет аудит. Два маршрута к одному физическому ресурсу не дают двух успехов.

23.4 Сводка раскрывается до страниц исходных результатов с учетом прав наблюдателя. Недоступная ветвь сохраняется в expected-наборе с неизвестным результатом. Отменяющий ответ без исторических данных не доказывает notStarted. Старая привязка не сливается с новой по совпадению IP.

Групповой запуск не является атомарной транзакцией. Политика частичной готовности задается до запуска. Остановка уже исполняемых ветвей по этой политике возможна только в своей области полномочий; приоритет или соседняя неисправность не дают права прервать чужое оповещение.

24 Регистрация и согласованная конфигурация

24.1 Обнаружение сообщает кандидата, регистрация устанавливает доверие и назначение. Discovery опционально, ограничено разрешенной областью, временем и квотой; статическая настройка остается возможной. Найденный IP, имя и объявленные возможности не дают устройству права управления.

24.2 Допуск

Проверка идентичности УУ, уникальности ID, роли, права регистрации и привязки конкретного ОУ.

24.3 Импорт

Пакет содержит schemaVersion, configId/revision, целевые ID, профили, группы, сценарии, манифесты контента и происхождение. Рабочие ключи и идентичность не клонируются.

24.4 Валидация

До применения проверяются ссылки, коллизии, права, циклы, неподдержанные функции, отсутствующие записи и лимиты.

24.5 Согласование групп

Уполномоченный автор области, groupRevision и digest состава. Неизвестная ревизия не заменяется одноименной локальной группой.

24.6 Применение

Раздельные desired, saved и applied, причина расхождения и ожидаемый перезапуск. Ответ на SET не заменяет чтения фактического состояния.

24.7 Замена оборудования

Новая физическая идентичность и привязка, отзыв прежнего назначения, проверяемый перенос совместимых параметров. Активные сеансы не переносятся.

24.8 Распределенное применение пакета возвращает applied/rejected/pending/unknown по каждому элементу и позволяет повторное согласование. Оно не является одной атомарной транзакцией всех

устройств. Откат проходит валидацию как новое разрешенное изменение и не отменяет исторически выполненные физические действия.

24.9 Пакет конфигурации - UTF-8 JSON с обязательными schemaVersion="open-alert-config/5", configId, revision, authorId, targets, groups, programs, content, profiles, signalSpecs, controllers и localRules. ID и uint64 представлены строками, повтор ключа запрещен. Для digest применяется Стрибог-256 от JCS по RFC 8785; перед канонизацией множества и вложенные коллекции сортируются по правилам схемы open-alert-config/5 ниже; порядок этапов сохраняется. authorId связывается с аутентифицированным правом изменения области, а не принимается на доверии из файла. Импорт не переносит PSK, deviceId, bootId или текущие контексты владения.

24.10 Применение нового configRevision сохраняет состояние эпизода неизменного localRule.id/revision. Смена постороннего параметра не взводит вход заново и не повторяет запуск. Любое изменение самого правила требует увеличения его revision; новый набор проверяется с правами на указанный purpose до commit. Для получателя-УУ пакет обязан содержать controller с собственным deviceId и его локальные цели; отсутствие собственной политики не заменяется значением по умолчанию.

24.11 open-alert-config/5 содержит обязательные коллекции controllers и localRules. controller.id равен deviceId; каждому deviceId из targets соответствует ровно одна запись controllers. onControlLoss задает сохраненную политику real-сеансов этого УУ. Для exercise/technical и сетевой речи действует stop независимо от этой настройки. Пустая localRules явно отключает локальные запуски; отсутствие любой обязательной коллекции отклоняет пакет. Политики не кодируются произвольными именами profiles.parameters.

24.12 localRule.id и revision идентифицируют правило и его редакцию; deviceId/inputId связывают его с существующим физическим

bool-входом каталога этого УУ. Для пары deviceId/inputId ДОПУСКАЕТСЯ одно правило. activeValue задает активную полярность; trigger=edge|level определяет поведение раздела 27. debounceMs и rearmMs: 1..60000; programId/programRevision ссылаются на неизменную локальную программу пакета. targetIds: 1..4 различные цели того же deviceId; сеть и цели другого УУ для автономного правила запрещены. purpose=real| exercise требует соответствующего разрешения регистрации источника. maxDurationMs: 100..86400000, для exercise не более 1080000 с выполнением правил допуска длительных учений; это полный срок вместе с подготовкой и паузами. Допустимость программы, наличие локального контента и физические границы остановки проверяются до commit. Неизвестный вход, неподтвержденная привязка, отсутствующая редакция или недостаточные полномочия отклоняют весь пакет.

24.13 controllers сортируются по id, localRules по id, localRule.targetIds по строке ID; повтор запрещен. Смена конфигурации не является фронтом входа. Новое или измененное правило edge сначала ожидает устойчивой неактивности; level получает новое наблюдение после атомарного применения конфигурации. Изменение или удаление правила с выполняющимся либо неопределенным сеансом запрещено. Пакет атомарно сохраняет политику потери управления и весь набор локальных правил вместе с configRevision.

Закрытая схема конфигурации open-alert-config/5

24.14 config-schema-0.21.json является нормативной JSON Schema Draft 2020-12. Все объекты закрыты: неизвестное поле, null вместо значения, повтор ключа, неканоническое число или превышение uint64 отклоняет весь пакет до stage. Строки без Unicode-нормализации, предел текстового значения 128 байт UTF-8; ID/token и максимальные количества заданы схемой. Пакет не более 1048576 байт после нормализации и JCS. Пределы maxItems каждой коллекции и общего размера дей-

ствуют одновременно: их произведение не является обещанной емкостью одного пакета. Набор с допустимыми размерами отдельных коллекций, но превышающий общий размер, недопустим. Автор вычисляет точную длину JCS до передачи; config.stage с bytes выше предела получает invalidArgument до выделения staging. Общий пакет не разбивается на независимые частичные commit. Машинный параметр x-maxJcsBytes требует отдельной проверки длины канонических байтов, поскольку стандарт JSON Schema не измеряет сериализованный размер. Отдельная машинная схема и таблицы полей описывают один формат.

24.15 В таблицах каждое поле занимает отдельную строку. Знак ? после имени обозначает необязательное поле; остальные поля обязательны. Варианты объекта приведены отдельно. Целые в строке передаются в канонической десятичной записи; обе границы диапазона включены. Предел в байтах UTF-8 применяется к декодированному значению; непарные суррогаты ЗАПРЕЩЕНЫ. Для массива указаны минимальное и максимальное количество элементов и тип элемента. Дополнительные межполевые условия приведены в правилах соответствующего профиля.

Таблица 51 - Объект root

Поле	**Тип и допустимые значения**
schemaVersion	Точное значение: "open-alert-config/5"
configId	Строка: 32 символа [0-9a-f]
revision	Целое в строке: 1..18446744073709551615
authorId	Строка: 32 символа [0-9a-f]
targets	Массив 0..256: target
groups	Массив 0..128: group
programs	Массив 0..128: program
content	Массив 0..256: content
profiles	Массив 0..256: profile
signalSpecs	Массив 0..256: signalSpec
mediaGroups	Массив 0..128: mediaGroup
controllers	Массив 0..256: controller
localRules	Массив 0..128: localRule

Таблица 52 - Объект mediaGroup — security="tls"

Поле	**Тип и допустимые значения**
id	Строка: 1..32 символа [A-Za-z0-9_.:-]
revision	Целое в строке: 1..18446744073709551615
deviceId	Строка: 32 символа [0-9a-f]
sourcePrincipalId	Строка: 32 символа [0-9a-f]
security	Точное значение: "tls"
sourceAddress	Строка: адрес IPv4
sourcePort	Целое в строке: 1..65535
groupAddress	Строка: адрес IPv4
destinationPort	Целое в строке: 1..65535
ttl	Целое в строке: 1..255
sourceKeyId	Строка: 1..32 символа [A-Za-z0-9_.:-]
publicX	Строка: 64 символа [0-9a-f]
publicY	Строка: 64 символа [0-9a-f]
resources	Массив 1..4: Объект mediaGroup.resources[]

Таблица 53 - Объект mediaGroup — security="plain"

Поле	Тип и допустимые значения
id	Строка: 1..32 символа [A-Za-z0-9_.:-]
revision	Целое в строке: 1..18446744073709551615
deviceId	Строка: 32 символа [0-9a-f]
sourcePrincipalId	Строка: 32 символа [0-9a-f]
security	Точное значение: "plain"
sourceAddress	Строка: адрес IPv4
sourcePort	Целое в строке: 1..65535
groupAddress	Строка: адрес IPv4
destinationPort	Целое в строке: 1..65535
ttl	Целое в строке: 1..255
resources	Массив 1..4: Объект mediaGroup.resources[]

Таблица 54 - Объект mediaGroup.resources[]

Поле	Тип и допустимые значения
resourceId	Строка: 1..32 символа [A-Za-z0-9_.:-]
bindingGeneration	Целое в строке: 1..18446744073709551615

Таблица 55 - Объект signalSpec — kind="builtIn"

Поле	Тип и допустимые значения
id	Строка: 1..32 символа [A-Za-z0-9_.:-]
revision	Целое в строке: 1..18446744073709551615
kind	Точное значение: "builtIn"

Таблица 56 - Объект signalSpec — вариант 2

Поле	Тип и допустимые значения
id	Строка: 1..32 символа [A-Za-z0-9_.:-]
revision	Целое в строке: 1..18446744073709551615
kind	Строка; одно из: tone, linearSweep, mechanical
waveform	Строка; одно из: sine, square, mechanical
minHz	Целое в строке: 1..20000
maxHz	Целое в строке: 1..20000
toleranceHz	Целое в строке: 0..19999
periodMs	Точное значение: "0" ИЛИ Целое в строке: 100..60000

Таблица 57 - Объект target

Поле	Тип и допустимые значения
id	Строка: 32 символа [0-9a-f]
ownerNodeId	Строка: 32 символа [0-9a-f]
deviceId	Строка: 32 символа [0-9a-f]
resourceId	Строка: 1..32 символа [A-Za-z0-9_.:-]
bindingGeneration	Целое в строке: 1..18446744073709551615
networkScope	Строка: 1..32 символа [A-Za-z0-9_.:-]
address	Строка: адрес IPv4
port	Целое в строке: 1..65535

Таблица 58 - Объект group

Поле	Тип и допустимые значения
id	Строка: 1..32 символа [A-Za-z0-9_.:-]
revision	Целое в строке: 1..18446744073709551615
members	Массив 1..256: Строка: 32 символа [0-9a-f]; без повторов

Таблица 59 - Объект siren

Поле	Тип и допустимые значения
pattern	Строка; одно из: continuous, intermittent
durationMs	Целое в строке: 100..3600000
signalProfileId	Строка: 1..32 символа [A-Za-z0-9_.:-]
signalProfileRevision	Целое в строке: 1..18446744073709551615

Таблица 60 - Объект recording

Поле	Тип и допустимые значения
contentId	Строка: 1..32 символа [A-Za-z0-9_.:-]
revision	Целое в строке: 1..18446744073709551615
hashAlgorithm	Точное значение: "streebog256"
hash	Строка: 64 символа [0-9a-f]

Таблица 61 - Объект pause

Поле	Тип и допустимые значения
durationMs	Целое в строке: 0..60000

Таблица 62 - Объект stage — kind="siren"

Поле	Тип и допустимые значения
index	Целое в строке: 1..8
kind	Точное значение: "siren"
repeat	Целое в строке: 1..16
gapMs	Целое в строке: 0..60000
siren	siren

Таблица 63 - Объект stage — kind="recording"

Поле	**Тип и допустимые значения**
index	Целое в строке: 1..8
kind	Точное значение: "recording"
repeat	Целое в строке: 1..16
gapMs	Целое в строке: 0..60000
recording	recording

Таблица 64 - Объект stage — kind="pause"

Поле	**Тип и допустимые значения**
index	Целое в строке: 1..8
kind	Точное значение: "pause"
repeat	Точное значение: "1"
gapMs	Точное значение: "0"
pause	pause

Таблица 65 - Объект program

Поле	**Тип и допустимые значения**
id	Строка: 1..32 символа [A-Za-z0-9_.:-]
revision	Целое в строке: 1..18446744073709551615
stages	Массив 1..8: stage

Таблица 66 - Объект placement

Поле	**Тип и допустимые значения**
targetId	Строка: 32 символа [0-9a-f]
location	Строка; одно из: controller, ou

Таблица 67 - Объект content

Поле	Тип и допустимые значения
id	Строка: 1..32 символа [A-Za-z0-9_.:-]
revision	Целое в строке: 1..18446744073709551615
codec	Строка; одно из: PCMA, GSM
bytes	Целое в строке: 1..33554432
sampleCount	Целое в строке: 1..18446744073709551615
hashAlgorithm	Точное значение: "streebog256"
hash	Строка: 64 символа [0-9a-f]
placements	Массив 1..256: placement

Таблица 68 - Объект parameter

Поле	Тип и допустимые значения
name	Строка: 1..32 символа [A-Za-z0-9_.:-]
type	Строка; одно из: bool, enum, int, counter, string
value	Строка: не более 128 байт UTF-8

Таблица 69 - Объект profile

Поле	Тип и допустимые значения
id	Строка: 1..32 символа [A-Za-z0-9_.:-]
revision	Целое в строке: 1..18446744073709551615
kind	Строка; одно из: vau, siren
targetIds	Массив 1..256: Строка: 32 символа [0-9a-f]; без повторов
parameters	Массив 1..64: parameter

Таблица 70 - Объект controller

Поле	Тип и допустимые значения
id	Строка: 32 символа [0-9a-f]
onControlLoss	Строка; одно из: stop, finish

Таблица 71 - Объект localRule

Поле	Тип и допустимые значения
id	Строка: 1..32 символа [A-Za-z0-9_.:-]
revision	Целое в строке: 1..18446744073709551615
deviceId	Строка: 32 символа [0-9a-f]
inputId	Строка: 1..32 символа [A-Za-z0-9_.:-]
trigger	Строка; одно из: edge, level
activeValue	Строка; одно из: true, false
debounceMs	Целое в строке: 1..60000
rearmMs	Целое в строке: 1..60000
programId	Строка: 1..32 символа [A-Za-z0-9_.:-]
programRevision	Целое в строке: 1..18446744073709551615
purpose	Строка; одно из: real, exercise
maxDurationMs	Целое в строке: 100..86400000
targetIds	Массив 1..4: Строка: 32 символа [0-9a-f]; без повторов

24.16 В каждой коллекции id уникален; каждая цель связана не более чем с одним физическим профилем. Targets описывают адресацию существующих ресурсов, а не меняют deviceId прибора. port:1..65535; bindingGeneration/revision положительны. Тройка ownerNodeId/deviceId/resourceId не повторяется. IPv4 используется только в указанной networkScope; авторизованный реестр связывает ownerNodeId с доверенным узлом. group.members ссылаются только на targets этого пакета, вложенных групп и циклов нет. Все profile.targetIds и content.placements.targetId также существуют в targets; placement не повторяет targetId. Программа ссылается на content.id/revision/hash и signalProfileId/revision из того же пакета; несогласованность отклоняет

весь пакет. Профили, требуемые программой, ДОЛЖНЫ быть применимы ко всем выбранным целям session.start.

24.17 stage.index последовательно 1..число этапов; repeat:1..16, gapMs:0..60000. pause требует repeat=1/gapMs=0 и durationMs:0..60000; siren.durationMs:100..3600000. В config/5 длительность этапа всегда конечная; stopMode в JSON не передается, при экспорте XML programs.list указывается stopMode=duration. Размер content.bytes:1..33554432, кодек и sampleCount удовлетворяют правилам локального файла раздела 15. parameter.name уникален в профиле; bool имеет true/false, int - int64, counter - uint64, enum - token, string - строку до 128 байт. Обязательные параметры физического паспорта и межполевые ограничения раздела 10 проверяются сверх структуры JSON. Импорт манифеста не присваивает verified: фактические байты и наблюдения проверяются независимо.

24.18 Нормализация множеств перед JCS: targets/groups/programs/content/profiles/signalSpecs/controllers/localRules сортируются по id, members и targetIds - по строке ID, placements - по targetId, parameters - по name. Порядок stages сохраняется; индекс сверяется с позицией, перестановка не исправляется автоматически. УУ принимает по config.stage только полученные канонические байты; их хеш совпадает с хешем сохраненного пакета. configId идентифицирует линию конфигурации, revision возрастает; одинаковая пара с другими байтами дает conflict. authorId ДОЛЖЕН совпадать с разрешенным автором аутентифицированного principalId. Общий пакет содержит полное локальное состояние, а не неявный patch; удаление занятого объекта запрещено.

25 Паспорт масштаба и бюджета времени

25.1 Системная емкость не вычисляется из размера адресного поля или числа резервных копий АРМ. ТРЕБУЕТСЯ паспорт конкретного

развертывания и испытание на объявленной нагрузке. Резервная копия того же узла не умножает число независимых подчиненных.

Таблица 72 - Паспорт масштаба и бюджета времени

Измерение	**Что фиксируется**
Топология	Число узлов, приборов и ресурсов, глубина маршрута, число целей группы и пересечения путей.
Канал	Полоса, RTT, потери, восстановление соединений и допустимые перерывы.
Конкурентность	Число одновременно запускаемых приборов, наблюдателей, опросов и сервисных работ; ограниченный пул на АРМ.
Ограничение нагрузки	Квоты очередей, событий, отчетов, discovery и reconnect; разброс задержек повторного подключения.
Время	Срок приема, подготовка, время до наблюдаемого действия, session.stop, таймауты источников; критический путь вместо безусловного умножения таймеров.
Отказы	Шторм событий, недоступные ветви, разделение сети, потеря питания и насыщение журнала; отдельно каждый ожидаемый результат.

25.2 АРМ распараллеливает работу разных приборов в пределах квот, а УУ сохраняет установленный порядок изменяющих команд и воздействий на общие ресурсы ОУ. Оповещение не ожидает полного последовательного обхода всей диагностики. Численные целевые по-

казатели задаются требованиями развертывания и подтверждаются стендовыми измерениями.

25.3 Предел задержки session.stop измеряется с учетом уже выполняемой операции внутреннего интерфейса, защищенной записи TLS, медленного читателя TCP, Flash и криптографической обработки. Приоритет внутри очереди не прерывает уже заблокированный обмен. Нужны конечные сроки каждого шага, ограничение исходящих данных и сохраняемая возможность принять отмену. Конкретные пределы подтверждаются на целевой плате.

26 Этапы программы и повторы

26.1 Сценарий АРМ, запуск runId, ветвь branchId, сеанс УУ и этап программы различаются. Сирена, пауза, встроенная запись и живая речь не объединяются одним неопределенным признаком "идет оповещение". Изменение этапа сохраняет происхождение запуска и права остановки.

Таблица 73 - Этапы программы и повторы

Правило	**Контракт**
Контекст этапа	revision программы, stageIndex и passIndex; результаты и отложенные действия относятся к точному этапу. Поздний ответ прежнего этапа не завершает новый.
Переход	Профиль определяет порядок остановки, подтверждения состояния, переключения входа и подготовки следующего действия. Нельзя включить речь и затем выполнить задержанный session.stop сирены.
Повтор записи	Различаются повтор доставки команды и новый проход программы. Для нового прохода нужны подтвержденный конец предыдущего, предел числа проходов и длительности, интервал паузы и собственный контекст.
Отмена	Запрещает еще не исполненные этапы, повторы, таймеры и выдачу новых медиаразрешений. Подтверждение отмены не теряется после очистки исполнительной очереди.
Завершение	Конец шага, выключение ресурса и завершение всей ветви являются отдельными результатами. Таймер АРМ не заменяет наблюдение результата ОУ.

Границы автономности

26.2 АРМ связывает отдельные конечные сеансы в управляемый им план либо запускает единый автономный mode=program. Для первого варианта следующий session.start является отдельным решением АРМ и

требует живой связи; для второго весь план принят УУ заранее и исполняется по правилам настоящего раздела. Выбор варианта входит в задание оповещения.

26.3 Завершение сирены в плане "сирена, затем речь" закрывает этап сирены, а не всю ветвь. Для подготовки следующего этапа АРМ различает ожидание, готовность, неподдержанную функцию, отказ и неизвестный результат; показывает причину и число ожидаемых получателей. Недоступность медиарежима не маскируется под продолжающуюся подготовку или сетевой таймаут. Сводная готовность группы не доказывает готовность каждого прибора: учитываются возможности конкретных получателей и заранее выбранная политика частичного исполнения.

26.4 Отображение ожидания и доступность кнопок относятся к интерфейсу АРМ. Они не добавляют XML-команд и не заменяют проверку готовности и прав исполнителем при фактическом запуске. Полученное прежде ready не дает права обойти изменившееся состояние УУ.

Для сценария, требующего завершения последовательности "сирена, затем речь" после потери АРМ, НЕОБХОДИМО использовать purpose=real, mode=program с локальной записью и onControlLoss=finish. Последовательность самостоятельных сетевых session.start не удовлетворяет этому требованию.

26.5 Ядро определяет один конечный сеанс siren, recording, program, stream или test. Объявление режима требует исполнения всего его контракта. Владение обычного оповещения сохраняется между самостоятельными сеансами при действующем lease; технический тест освобождает эпоху автоматически после off и закрытия результата. Автономный plan фиксируется единым mode=program и выполняется без новых сетевых session.start.

Автономное исполнение составной программы

26.6 Составное оповещение с автономным завершением запускается одним session.start с mode=program и дочерним program, со-

держащим id и revision. Программа хранится на стороне УУ как неизменяемый манифест из 1..8 последовательных этапов siren, recording или pause. Ревизия идентифицирует весь упорядоченный план, параметры и манифесты записей. session.start фиксирует полный план в журнале намерения до первого воздействия; будущие этапы не требуют новых сетевых session.start.

26.7 id / revision

id: 1..32 ASCII-символа; revision: uint64. Изменение любого этапа или содержимого создает новую ревизию. Программа и ее содержимое недоступны для изменения или удаления до завершения всех ссылающихся исполнений.

26.8 stage

index: 1..8 без пропусков; kind=siren|recording|pause; repeat: 1..16; gapMs: 0..60000. Ресурсы наследуются из session.start; скрытые дополнительные ресурсы запрещены.

26.9 siren

pattern=continuous|intermittent, stopMode=duration в XML programs.list, durationMs=100..3600000, signalProfileId и signalProfileRevision. Параметры каждого прохода неизменны.

26.10 recording

contentId, revision, hashAlgorithm=streebog256, hash из 64 строчных hex; целиком локально доступная запись, один проход за повтор.

26.11 pause

durationMs=0..60000, repeat=1, gapMs=0. Выходы остаются off; отмена в паузе запрещает следующий этап.

26.12 Предел

Сумма всех подготовок, проходов, остановок и пауз ДОЛЖНА укладываться в maxDurationMs session.start <= 86400000. УУ проверяет верхнюю оценку до accepted. При неизвестной верхней длительности записи session.start отклоняется.

26.13 Все ресурсы и записи программы проверяются до accepted. Для перехода к следующему этапу УУ подтверждает off предыдущего, прекращает его таймеры и медиаразрешения и присваивает локальные stageIndex/passIndex. Позднее сообщение предыдущего этапа не действует на новый. Любой отказ или unknown прерывает оставшуюся программу и инициирует остановку; пропуск этапа при ошибке запрещен. succeeded относится ко всей программе и требует выполнения всех этапов.

26.14 Результат program дополнительно содержит stageResult для каждого предусмотренного прохода: stageIndex, passIndex, state=pending|running|stopping|finished|notStarted и, при finished, result/reason. Поля runningAtMs, stoppingAtMs, finishedAtMs присутствуют только для состоявшихся переходов. acceptedAtMs <= runningAtMs <= stoppingAtMs <= finishedAtMs <= evaluatedAtMs; время следующего runningAtMs не меньше предыдущего finishedAtMs + gapMs между повторами. Времена каждого прохода лежат между началом плана и его finishedAtMs, если он завершен. Для pause runningAtMs обозначает начало паузы и не является доказательством включения. pending относится только к будущему проходу активного плана; после прекращения плана он становится notStarted с reason=programAborted из закрытого словаря, без result и времен воздействия; programAborted запрещен у session и у finished-этапа. Последний исполненный проход задает начало общей terminal stopping; промежуточные остановки этапов не меняют общую timeline. У каждого прохода применяются собственные prepareTimeoutMs, stopBeginMs и stopConfirmMs аппаратного профиля. Пара stageIndex/passIndex уникальна и соответствует зафиксированному манифесту. Снимок содержит все проходы, включая notStarted; до session.start УУ проверяет, что их максимальное число укладывается в объявленные память, snapshotMaxPages и snapshotReadMaxMs. Успех требует finished/succeeded всех проходов.

26.15 При purpose=real onControlLoss=finish сохраняет исполнение всего принятого плана, включая подготовку, паузы и последующие локальные речевые этапы, в исходном maxDurationMs. stop отменяет все еще не исполненные этапы. После reboot автоматическое возобновление программы запрещено: выполняется восстановление ресурсов. Живой микрофон не включается в автономный план; для заранее предусмотренного сообщения используется локальная запись.

26.16 programs.list является обязательным чтением при объявлении mode=program. Запрос содержит selection с необязательным programId либо парой snapshotId/cursor. Ответ programs содержит snapshotId, capabilityRevision, more и cursor при more=true. Записи program несут id/revision/stageCount; записи stage несут programId/index/kind/repeat/gapMs и один дочерний siren, recording или pause с полями таблицы. Полный снимок содержит ровно stageCount различных индексов каждого выбранного program. gapMs применяется между повторами одного этапа, после последнего повтора дополнительная пауза отсутствует. Манифесты передаются страницами в пределах общих лимитов XML; снимок не запускает исполнение.

27 Внешние входы и автономное управление

27.1 Внешний контакт, местная кнопка, тангента и запуск из другой системы являются отдельными источниками. Наличие входа не дает ему права прерывать сетевой сеанс. Каталог указывает реально поддержанный источник, а конфигурация - разрешенное действие, область ресурсов и правила взаимодействия с действующим управлением.

27.2 Новое срабатывание

Источник и эпизод однозначно идентифицированы; правило запуска имеет ID и ревизию. Решение, запуск и его происхождение связаны в журнале.

27.3 Дребезг и удержание

Профиль задает обработку фронта/уровня, подавление дребезга, повторное взведение и максимальную длительность. Частота событий не становится частотой повторного запуска.

27.4 Снимок и повтор события

Начальная синхронизация, восстановление истории и повтор доставки не запускают сценарий заново. Один эпизод не создает несколько независимых запусков.

27.5 Уже активный вход после питания

Отличается от нового фронта. Запрет автоматического возобновления прежнего сетевого сеанса сохраняется; trigger=edge требует нового фронта после повторного взведения; trigger=level повторно оценивает текущее физическое условие после восстановления готовности.

27.6 ОУ запущено вне УУ

Передается наблюдаемое внешнее управление и его доступный источник. Чужое исполнение не присваивается текущему sessionId, и ложный успех нашей команды не формируется.

27.7 Отмена и аппаратная защита

Сетевое прерывание проверяет происхождение. Для местного запуска и защитного отключения профиль отдельно задает полномочия и реакцию; универсальный удаленный reset не вводится.

27.8 АРМ исполняет централизованные правила, а УУ - локальные правила из зафиксированной конфигурации. Закрытый формат localRules задает идентичность, редакцию, привязку входа, активную полярность, trigger, фильтры, программу, ресурсы и полный срок. Правила edge и level различаются по условиям создания нового эпизода, описанным ниже. Удержанный контакт и восстановление снимка не являются фронтом. Местный запуск получает отдельные principalId/controlEpoch/sessionId; сетевой клиент не присваивает это владение. Конфликт не прерывает действующий сеанс. Для edge он дает отказ без отложенного

включения; для level условие повторно проверяется после освобождения ресурса.

27.9 Локальный источник имеет зарегистрированную идентичность и собственный сеанс. Свободный ресурс может запускаться только разрешенным правилом входа. При сетевом владении несовместимое местное включение отклоняется и регистрируется; при местном исполнении сетевой session.start получает busy. Аппаратная аварийная защита действует независимо по паспорту ОУ. Если изделие допускает физическое ручное включение в обход УУ, оно ДОЛЖНО сообщать externalControl и не подтверждать его как исполнение сетевого session.start.

Импульс и длительно действующее условие тревоги

27.10 Локальные правила задаются коллекцией localRules схемы open-alert-config/5. Для trigger=edge при загрузке, применении нового правила и после каждого эпизода вход взводится только после неактивного значения в течение rearmMs. Затем активное значение в течение debounceMs создает один новый эпизод. Активный уровень при загрузке, удержание контакта и повтор сетевого события не являются новым фронтом. Если ресурс занят при фронте, отказ регистрируется без отложенного исполнения; повторное взведение требует нового неактивного интервала.

27.11 trigger=level означает действующее физическое условие. После загрузки или атомарного применения правила УУ сначала завершает восстановление ресурсов, подтверждает off и готовность, затем получает новое valid/fresh наблюдение своего входа. Устойчивое активное значение в течение debounceMs разрешает новый локальный запуск даже без нового фронта. Это решение по текущему входу с новым bootId/sessionId, а не повтор прежнего сетевого session.start или восстановление старой программы. УУ фиксирует источник, правило, configRevision

и наблюдение, обусловившее решение. Старый снимок, неизвестное значение и неисправный вход запуск не разрешают.

27.12 Для level при занятости ресурса активное условие сохраняется как ожидающее наблюдение, без заранее принятого session.start. После освобождения выполняются заново проверка прав, готовности, свежести и debounceMs. Устойчивое неактивное значение rearmMs снимает ожидание. Для одной редакции правила в одной непрерывной загрузке один активный эпизод создает не более одного принятого исполнения, включая случай его остановки или ошибки; повтор разрешен после неактивности rearmMs. После reboot применяется новое наблюдение по предыдущему правилу. Само снятие уровня не отменяет уже принятую конечную программу; ее остановка подчиняется полномочиям, maxDurationMs и аппаратной защите. Любой локальный запуск получает собственное владение и не прерывает чужой сеанс.

Приложение А

(обязательное для ядра)

Каталог команд

Обязательность команд определяется указанным профилем. Объявленная поддержка функции требует всех связанных с ней команд и правил.

A.1 events.subscribe - Включить инициативную доставку событий

Обязательность. Обязательно при eventsPush=1

Профиль. open-alert-draft/0.21

Полномочия. Право полного журнала данного УУ; eventsPush=1 и обе команды подписки объявлены. Владение НЕ ТРЕБУЕТСЯ.

Запрос. bootId в request; пустой subscription с subscriptionSeq (uint64 1..18446744073709551615) и afterSeq (uint64 0..18446744073709551615).

Ответ. ok; пустой subscription с id (32 строчных hex), subscriptionSeq и afterSeq, равными запросу. Уведомления начинаются только после ответа.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, notFound

Повторы и поведение. Одна подписка на соединение. Новый больший subscriptionSeq заменяет прежнюю подписку только после успешного принятия. Повтор последнего принятого номера с теми же id/afterSeq возвращает прежний ответ для действующей подписки. Меньший номер или повтор прекращенной подписки дает notFound; изменение id/afterSeq при том же номере дает invalidArgument. Не изменяет ОУ или lease. Формат notification/notificationAck, повтор и gap заданы разделом инициативных уведомлений.

A.2 events.unsubscribe - Отключить инициативную доставку событий

Обязательность. Обязательно при eventsPush=1

Профиль. open-alert-draft/0.21

Полномочия. Участник данного соединения; подписка другого соединения недоступна.

Запрос. bootId в request; пустой subscription с единственным id (32 строчных hex).

Ответ. ok; пустой subscription с тем же id. После ответа уведомления этой подписки не передаются.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported

Повторы и поведение. Повтор безопасен; неизвестный id также дает ok и не отменяет другую подписку. Уже начатый notification заканчивается перед ответом. Владение и исполнение ОУ сохраняются.

A.3 transfer.capabilities - Прочитать параметры файлового канала

Обязательность. Обязательно при любой сетевой файловой функции

Профиль. open-alert-draft/0.21

Полномочия. Аутентифицированный участник с правом чтения возможностей; владение НЕ ТРЕБУЕТСЯ.

Запрос. Без дочерних элементов; управляющее соединение.

Ответ. transfer profile/port/maxBlockBytes/maxWindowBlocks/maxInFlightBytes.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported

Повторы и поведение. Не изменяет данные и не продлевает lease. Параметры и допустимые сочетания определены в разделе 16.

A.4 transfer.hello - Привязать файловое соединение

Обязательность. Обязательно при любой сетевой файловой функции

Профиль. open-alert-draft/0.21

Полномочия. Тот же principalId, живое управляющее соединение и maintenance-эпоха владельца транзакции.

Запрос. Первый XML отдельного соединения того же режима tls либо plain: bootId/controlEpoch, transfer id/kind; cmdSeq отсутствует.

Ответ. ready id/blockBytes/windowBlocks/receivedBytes/remainingBudgetMs, затем двоичные кадры bulk/1.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, staleEpoch, resultExpired, unsafeState, notFound, staleAttachment

Повторы и поведение. Атомарная привязка к экземплярам двух соединений. Неуспешное подключение не изменяет существующую передачу; полные правила раздела 16.

A.5 audio.pause - Остановить источник без отбоя ОУ

Обязательность. Обязательно при sourceSwitch=true

Профиль. open-alert-draft/0.21

Полномочия. Действующий владелец с правом выбранного воздействия.

Запрос. stream mediaId/sessionId; обычный cmdSeq владельца согласованного holdOutput.

Ответ. paused mediaId/sessionId/startSeq; состояние источника через audio.state.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, staleEpoch, staleAttachment, conflict, resultExpired, notFound, unsafeState

Повторы и поведение. Изменение использует общие проверки контекста и дедупликацию cmdSeq. Точный повтор возвращает сохраненное решение без повторного действия и продления предметных сроков. Переходы операции и межполевые условия определены ее предметным контрактом.

A.6 audio.start - Запустить следующий источник

Обязательность. Обязательно при sourceSwitch=true

Профиль. open-alert-draft/0.21

Полномочия. Действующий владелец с правом выбранного воздействия.

Запрос. stream mediaId/sessionId после новой подготовки и BIND/READY; обычный cmdSeq владельца.

Ответ. started mediaId/sessionId/startSeq без повторного включения ОУ.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, staleEpoch, staleAttachment, conflict, resultExpired, notFound, unsafeState

Повторы и поведение. Изменение использует общие проверки контекста и дедупликацию cmdSeq. Точный повтор возвращает сохраненное решение без повторного действия и продления предметных сроков. Переходы операции и межполевые условия определены ее предметным контрактом.

A.7 firmware.cancel - Отменить прием или проверку образа

Обязательность. Обязательно при обновлении прошивки

Профиль. open-alert-draft/0.21

Полномочия. Владелец maintenance с административным правом соответствующей операции.

Запрос. upload id; cmdSeq владельца maintenance.

Ответ. upload id/state=cancelled; после activating - conflict.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, staleEpoch, staleAttachment, conflict, resultExpired, notFound, storageFault, unsafeState

Повторы и поведение. Отмена использует общие проверки контекста и дедупликацию cmdSeq, отзывает привязку передачи и останавливает незавершенную проверку. После атомарного commit возвращает conflict/ detail=operationStateMismatch. Точный повтор возвращает сохраненный ответ без повторного действия.

A.8 profiles.get - Паспорт физического исполнения

Обязательность. Обязательно: ядро

Профиль. open-alert-draft/0.21

Полномочия. Право чтения выбранной области.

Запрос. selection resourceId либо resourceId/snapshotId/cursor.

Ответ. profiles с profile и parameter по разделу 10.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, notFound, unsupported, snapshotExpired

Повторы и поведение. Чтение не изменяет выходы, операцию или lease. Новый запрос читает актуальное состояние; продолжение снимка сохраняет его контекст.

A.9 programs.list - Каталог автономных программ

Обязательность. Обязательно при mode=program

Профиль. open-alert-draft/0.21

Полномочия. Право чтения выбранной области.

Запрос. selection programId либо snapshotId/cursor.

Ответ. Страницы programs с program и stage; формат раздела 26.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, notFound, snapshotExpired

Повторы и поведение. Чтение не изменяет выходы, операцию или lease. Новый запрос читает актуальное состояние; продолжение снимка сохраняет его контекст.

A.10 audio.prepare - Подготовить прием потока

Обязательность. Обязательно при mode=stream

Профиль. open-alert-draft/0.21

Полномочия. Действующий владелец с правом выбранного воздействия.

Запрос. stream с параметрами RTP, resources, route и buffer по разделу 15; обычный cmdSeq владельца.

Ответ. prepared mediaId/expiresAtMs/configRevision/capabilityRevision.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, staleEpoch, staleAttachment, conflict, resultExpired, notFound, unsafeState

Повторы и поведение. Изменение использует общие проверки контекста и дедупликацию cmdSeq. Точный повтор возвращает сохраненное решение без повторного действия и продления предметных сроков. Переходы операции и межполевые условия определены ее предметным контрактом.

A.11 audio.state - Прочитать состояние потока

Обязательность. Обязательно при mode=stream

Профиль. open-alert-draft/0.21

Полномочия. Право чтения выбранной области.

Запрос. selection mediaId.

Ответ. audio со состоянием и счетчиками по разделу 15.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, notFound

Повторы и поведение. Чтение не изменяет выходы, операцию или lease. Новый запрос читает актуальное состояние; продолжение снимка сохраняет его контекст.

A.12 audio.finish - Зафиксировать конец источника

Обязательность. Обязательно при mode=stream

Профиль. open-alert-draft/0.21

Полномочия. Действующий владелец с правом выбранного воздействия.

Запрос. stream mediaId/lastPacketIndex/sampleCount; обычный cmdSeq владельца.

Ответ. eof с теми же полями; полный физический итог через session.get.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, staleEpoch, staleAttachment, conflict, resultExpired, notFound, unsafeState

Повторы и поведение. Изменение использует общие проверки контекста и дедупликацию cmdSeq. Точный повтор возвращает сохраненное решение без повторного действия и продления предметных сроков. Переходы операции и межполевые условия определены ее предметным контрактом.

A.13 audio.discard - Освободить неиспользованную подготовку

Обязательность. Обязательно при mode=stream

Профиль. open-alert-draft/0.21

Полномочия. Действующий владелец с правом выбранного воздействия.

Запрос. prepared mediaId; обычный cmdSeq владельца.

Ответ. discarded mediaId; для активного сеанса unsafeState.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, staleEpoch, staleAttachment, conflict, resultExpired, notFound, unsafeState

Повторы и поведение. Изменение использует общие проверки контекста и дедупликацию cmdSeq. Точный повтор возвращает сохраненное решение без повторного действия и продления предметных сроков. Переходы операции и межполевые условия определены ее предметным контрактом.

A.14 content.begin - Начать размещение записи

Обязательность. Обязательно при сетевой загрузке контента

Профиль. open-alert-draft/0.21

Полномочия. Владелец maintenance с административным правом соответствующей операции.

Запрос. content с неизменным манифестом и transfer с параметрами bulk/1; обычный cmdSeq.

Ответ. upload id/receivedBytes/remainingBudgetMs.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, staleEpoch, staleAttachment, conflict, resultExpired, unsafeState, noSpace, storageFault

Повторы и поведение. Изменение использует общие проверки контекста и дедупликацию cmdSeq. Точный повтор возвращает сохраненное решение без повторного действия и продления предметных сроков. Переходы операции и межполевые условия определены ее предметным контрактом.

A.15 content.commit - Проверить и опубликовать запись

Обязательность. Обязательно при сетевой загрузке контента

Профиль. open-alert-draft/0.21

Полномочия. Владелец maintenance с административным правом соответствующей операции.

Запрос. upload id; обычный cmdSeq; только после полного приема.

Ответ. content id/revision/state=committed либо accepted с operation.id=upload.id; итог в content.status.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, staleEpoch, staleAttachment, conflict, resultExpired, notFound, unsafeState, storageFault

Повторы и поведение. Изменение использует общие проверки контекста и дедупликацию cmdSeq. Точный повтор возвращает сохраненное решение без повторного действия и продления предметных сроков. Переходы операции и межполевые условия определены ее предметным контрактом.

A.16 content.status - Состояние размещения записи

Обязательность. Обязательно при сетевой загрузке контента

Профиль. open-alert-draft/0.21

Полномочия. Право чтения выбранной области.

Запрос. selection uploadId.

Ответ. upload id/state/receivedBytes/totalBytes/remainingBudgetMs; progressBytes в verifying, reason в failed; раздел 16.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, notFound, storageFault

Повторы и поведение. Чтение не изменяет выходы, операцию или lease. Новый запрос читает актуальное состояние; продолжение снимка сохраняет его контекст.

A.17 content.cancel - Отменить размещение записи

Обязательность. Обязательно при сетевой загрузке контента

Профиль. open-alert-draft/0.21

Полномочия. Владелец maintenance с административным правом соответствующей операции.

Запрос. upload id; обычный cmdSeq.

Ответ. upload id/state=cancelled.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, staleEpoch, staleAttachment, conflict, resultExpired, notFound, storageFault

Повторы и поведение. Отмена использует общие проверки контекста и дедупликацию cmdSeq, отзывает привязку передачи и останавливает незавершенную проверку. После атомарного commit возвращает conflict/ detail=operationStateMismatch. Точный повтор возвращает сохраненный ответ без повторного действия.

A.18 audio.group.prepare - Подготовить групповую речь

Обязательность. Обязательно при объявлении group-audio/2 либо group-audio-open/1

Профиль. open-alert-draft/0.21

Полномочия. Владелец с правом на все ресурсы шаблона mediaGroups; sourcePrincipalId совпадает с участником.

Запрос. group id/revision/streamId/source; totalSamples только для recording; key (64 hex) только при tls. Обычный cmdSeq. Полный контракт group-audio/2 и group-audio-open/1 — нормативное приложение Ж; в онлайн-версии — раздел «Групповая передача речи».

Ответ. ok, prepared mediaId/expiresAtMs/configRevision/capabilityRevision. Выходы не включаются; session.start выполняется отдельно.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, staleEpoch, staleAttachment, conflict, resultExpired, unsafeState, notFound, sourceUnavailable

Повторы и поведение. Точный повтор возвращает прежний mediaId без продления 60000 мс. Измененное содержимое повтора — conflict. Отсутствие шаблона — notFound, его другая редакция — conflict, неподтвержденные часы — sourceUnavailable. Чужой sourcePrincipalId — forbidden/detail=ownerMismatch. После session.stop/reboot разрешение и оперативный ключ отозваны. Подготовка не обходит неподтвержденный off и не вытесняет активный поток.

A.19 protocol.hello - Выбрать профиль обмена

Обязательность. Обязательно: ядро

Профиль. open-alert-draft/0.21

Полномочия. Участник завершил TLS-аутентификацию либо явно выбрал разрешенный plain; предметный профиль еще не выбран.

Запрос. v=1, profile=bootstrap/1, id, action; hello содержит 1-8 profile id; при plain обязателен атрибут clientId (32 hex), при tls запрещен. Порядок списка не задает приоритет УУ.

Ответ. ok, bootId и hello selectedProfile/deviceId/security=tls|plain. Выбран разрешенный общий профиль по приоритету УУ, независимо от порядка предложения клиента.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden

Повторы и поведение. Точный повтор до первой предметной команды возвращает тот же выбор. После фиксации профиля другой hello запрещен; изменение предложения требует нового соединения.

A.20 readiness.get - Оценить готовность конкретного плана

Обязательность. Дополнительная возможность

Профиль. open-alert-draft/0.21

Полномочия. Право наблюдения за выбранными ресурсами; владение оповещением НЕ ТРЕБУЕТСЯ.

Запрос. plan mode/configRevision/capabilityRevision/maxDurationMs, resources и элемент режима по разделу 22; продолжение selection snapshotId/cursor.

Ответ. readiness snapshotId/evaluatedAtMs/validUntilMs/stateRevision/more/cursor, summary, resource и evidence по разделу 22.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, notFound, snapshotExpired

Повторы и поведение. Повтор разрешен; это новое чтение, поэтому данные могут измениться. Если команда использует снимок, все его страницы читаются с одним snapshotId. Чтение не включает ОУ, не запускает тест или полное хеширование, не резервирует ресурс и не поддерживает связь владельца. session.start повторно проверяет актуальные условия; старое ready не разрешает обход препятствий.

A.21 content.describe - Прочитать манифест и доказательства проверки записи

Обязательность. Обязательно при mode=recording или program

Профиль. open-alert-draft/0.21

Полномочия. Право чтения описаний контента в заданной области.

Запрос. selection contentId/revision либо snapshotId/cursor.

Ответ. contents с записями content и verification по разделу 22.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, notFound, snapshotExpired, storageFault

Повторы и поведение. Чтение не изменяет контент. verified ДОПУСКАЕТСЯ только при совпадении хеша доступных байтов с неизменным манифестом; unknown не содержит actualHash. session.start требует verified.

A.22 device.describe - Идентификация изделия

Обязательность. Обязательно: ядро

Профиль. open-alert-draft/0.21

Полномочия. Аутентифицированный участник с правом чтения; владение НЕ ТРЕБУЕТСЯ.

Запрос. Без предметных параметров. Общие v, profile и id обязательны; запрос выполняется после аутентификации.

Ответ. bootId в оболочке; device deviceId, kind=vau|siren, version, uptimeMs, securityPolicy=optional|secureOnly; отдельный info protocolName=Открытый протокол систем оповещения; manufacturer/softwareDeveloper необязательны и указывают фактических изготовителей. Профиль уже выбран через bootstrap.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy

Повторы и поведение. Повтор разрешен; это новое чтение, поэтому данные могут измениться. Если команда использует снимок, все его страницы читаются с одним snapshotId. Не возвращает ключи, пароли и заводские секреты.

A.23 device.capabilities - Возможности и пределы

Обязательность. Обязательно: ядро

Профиль. open-alert-draft/0.21

Полномочия. Читатель.

Запрос. Без дочернего элемента для первой страницы; selection snapshotId/cursor для продолжения каталога возможностей УУ.

Ответ. capabilityRevision, драйверы и привязки ОУ, поддержанные команды, режимы, ресурсные группы конфликтов, глубина доказательств, лимиты сообщений и опросов, виды собственных запросов УУ.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, snapshotExpired, notFound

Повторы и поведение. Повтор разрешен; это новое чтение, поэтому данные могут измениться. Если команда использует снимок, все его страницы читаются с одним snapshotId. Функция считается доступной только при явном объявлении для нужного ресурса.

A.24 device.state - Согласованное текущее состояние

Обязательность. Обязательно: ядро

Профиль. open-alert-draft/0.21

Полномочия. Читатель.

Запрос. Новый снимок: без дочернего элемента; при snapshotReplace=1 допустим selection replaces по разделу 21. Продолжение: selection snapshotId/cursor.

Ответ. bootId, stateRevision, eventSeq; обязательные configuration с configRevision/appliedRevision/onControlLoss, ownership/recovery; resourceCount/alarmCount/sessionCount и страницы ресурсов, текущего сеанса, активных и неизвестных аварий.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, snapshotExpired

Повторы и поведение. Повтор разрешен; это новое чтение, поэтому данные могут измениться. Если команда использует снимок, все его страницы читаются с одним snapshotId. Отсутствие активного сеанса не доказывает успешного окончания предыдущего.

A.25 inventory.get - Каталог оборудования и точек

Обязательность. Обязательно: ядро

Профиль. open-alert-draft/0.21

Полномочия. Читатель.

Запрос. Выборка по pointId, resourceId или ouId; cursor/snapshotId для продолжения.

Ответ. Страницы самостоятельных записей point с id и facet=identity|meaning|source|acquisition; enum pointId/code/label и threshold pointId/conditionId/operator/value/hysteresis/severity отдельными записями. Иерархия, типы, единицы, presence, support, readMethod, источник; capabilityRevision и bindingGeneration в контексте снимка.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, notFound, snapshotExpired

Повторы и поведение. Повтор разрешен; это новое чтение, поэтому данные могут измениться. Если команда использует снимок, все его страницы читаются с одним snapshotId. Отсутствующая точка и неподдерживаемая точка различаются.

A.26 sensors.read - Показания без воздействия

Обязательность. Обязательно: ядро

Профиль. open-alert-draft/0.21

Полномочия. Читатель.

Запрос. selection с 1-8 point id, без повторов; при snapshotReplace=1 допустим replaces по разделу 21. Продолжение содержит тот же набор и пару snapshotId/cursor, без replaces.

Ответ. Значения вместе с validity, freshness, reason, sourceId, sampleSeq и временем наблюдения; метаданные снимка.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, notFound, snapshotExpired

Повторы и поведение. Повтор разрешен; это новое чтение, поэтому данные могут измениться. Если команда использует снимок, все его страницы читаются с одним snapshotId. Кеш не освежается фактом чтения; физический тест не запускается.

A.27 diagnostics.read - Инженерная диагностика

Обязательность. Дополнительная возможность

Профиль. open-alert-draft/0.21

Полномочия. Право диагностики; обычное чтение может иметь меньшую детализацию.

Запрос. selection scope=controller|ethernet|control|ouInterface|media либо snapshotId/cursor.

Ответ. diagnostics snapshotId/uptimeMs/more/cursor и metric name/type/value/unit по разделу 13.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, snapshotExpired

Повторы и поведение. Повтор разрешен; это новое чтение, поэтому данные могут измениться. Если команда использует снимок, все его страницы читаются с одним snapshotId. Не сбрасывает счетчики, не перезагружает интерфейс и не меняет выходы.

A.28 tests.list - Профили опросов и функциональных проверок

Обязательность. Обязательно при mode=test

Профиль. open-alert-draft/0.21

Полномочия. Читатель.

Запрос. selection с необязательным testId либо snapshotId/cursor.

Ответ. tests snapshotId/capabilityRevision/more/cursor; test, check и observation по разделу 12.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, notFound, snapshotExpired

Повторы и поведение. Повтор разрешен; это новое чтение, поэтому данные могут измениться. Если команда использует снимок, все его страницы читаются с одним snapshotId. Перечень не запускает проверку. Выполнение: session.start с mode=test. Формат tests.list определен в разделе 12; каталог содержит все критерии и наблюдения принятого плана.

A.29 log.read - Журнал устройства

Обязательность. Дополнительная возможность

Профиль. open-alert-draft/0.21

Полномочия. Право чтения журнала.

Запрос. selection afterSeq/limit, limit=1..8.

Ответ. log oldestSeq/nextSeq/gap/more и entry seq/atMs/level/code/sessionId по разделу 13.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, cursorExpired

Повторы и поведение. Повтор разрешен; это новое чтение, поэтому данные могут измениться. Если команда использует снимок, все его страницы читаются с одним snapshotId. Ключи и секреты отсутствуют; утрата истории не заменяется пустым успешным результатом.

A.30 poll.start - Опрос без запуска ОУ

Обязательность. Дополнительная возможность

Профиль. open-alert-draft/0.21

Полномочия. Право опроса; владение оповещением НЕ ТРЕБУЕТСЯ.

Запрос. poll clientOperationId/deadlineMs/refresh=cache|readBus, 1..8 point id.

Ответ. accepted, operation id/clientOperationId/state=queued; результат через poll.get.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, conflict

Повторы и поведение. clientOperationId дедуплицируется в principalId/bootId до deadlineMs+resultRetentionMs. Другой набор полей с тем же ID дает conflict. Квоты: одно задание на участника, два на УУ. Чтение не включает выходы и не продлевает lease.

A.31 poll.get - Результаты опроса

Обязательность. Дополнительная возможность

Профиль. open-alert-draft/0.21

Полномочия. Создатель задания или роль с разрешенным доступом к нему.

Запрос. selection operationId либо operationId/snapshotId/cursor.

Ответ. poll snapshotId/operationId/state=queued|running|finished|cancelled/expectedCount/complete/evaluatedAtMs/more/cursor; point по формату sensors.read.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, notFound, snapshotExpired

Повторы и поведение. Повтор разрешен; это новое чтение, поэтому данные могут измениться. Если команда использует снимок, все его страницы читаются с одним snapshotId. complete означает собранный набор результатов, а не исправность всех точек.

A.32 poll.cancel - Отменить свой опрос

Обязательность. Дополнительная возможность

Профиль. open-alert-draft/0.21

Полномочия. Создатель задания с правом опроса.

Запрос. poll operationId/clientOperationId.

Ответ. cancelled operationId; уже полученные результаты сохраняются.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, notFound

Повторы и поведение. Не отменяет чужие опросы и не останавливает оповещение. Повтор для доступного завершенного задания возвращает его состояние; уже начатое получение наблюдений от ОУ может закончиться до отмены.

A.33 control.acquire - Получить управление

Обязательность. Обязательно: ядро

Профиль. open-alert-draft/0.21

Полномочия. Участник с правом управления; активного несовместимого владельца нет.

Запрос. control acquireSeq/instanceId/purpose=real|exercise|technical|maintenance; контекст bootId. Назначение проверяется по правам и сохраняется в эпохе.

Ответ. control с acquireSeq, instanceId, purpose, controlEpoch, highWater, attachSeq, leaseRemainingMs, current и idleDeadlineMs при отсутствии исполнения. Повторы и свободное владение - раздел 5.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, conflict, resultExpired, unsafeState, storageFault

Повторы и поведение. Последний acquireSeq и решение хранятся для допущенного principalId до смены bootId. Точный повтор возвращает ту же эпоху; current=false означает завершенное владение. На новом канале повтор только возвращает решение, а прикрепление выполняет resume. Новый номер не захватывает чужой сеанс.

A.34 control.resume - Восстановить свое управление

Обязательность. Обязательно: ядро

Профиль. open-alert-draft/0.21

Полномочия. Тот же логический владелец с подтвержденным правом восстановления.

Запрос. bootId/controlEpoch в оболочке; control acquireSeq, instanceId и возрастающий attachSeq. Право резервной копии требует ограждения прежней по разделу 5.

Ответ. control с прежними purpose/controlEpoch, новым attachSeq и текущими highWater/leaseRemainingMs/current; idleDeadlineMs не продлевается.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, staleEpoch, staleAttachment, conflict, storageFault

Повторы и поведение. Живая прежняя привязка не вытесняется: busy/detail=ownershipHeld. После ее отсоединения старый канал лишен изменяющих полномочий. Смена IP допустима после проверки идентичности и прежнего контекста. Повторное подключение не сбрасывает номера, не перезапускает сеанс и не решает НА само по себе. При утрате контекста после перезапуска АРМ читает актуальный снимок и сохраненный результат; присвоение чужого исполнения запрещено.

A.35 control.release - Освободить управление

Обязательность. Обязательно: ядро

Профиль. open-alert-draft/0.21

Полномочия. Текущий владелец, нет активного сеанса или занятых неизвестным исходом ресурсов.

Запрос. Текущие bootId/controlEpoch и новый cmdSeq.

Ответ. ok и released controlEpoch/released=true. Точный повтор на исходном либо новом допущенном канале того же участника возвращает сохраненный ответ из кеша завершенной эпохи.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, staleEpoch, staleAttachment, conflict, resultExpired, unsafeState, storageFault

Повторы и поведение. Точный повтор cmdSeq в том же bootId/controlEpoch возвращает сохраненное решение без повторного действия. Если решение и сведения об адресной отмене вытеснены, старый номер получает resultExpired. Не используется как замена явной остановке; при активном сеансе возвращает busy.

A.36 channel.ping - Проверить жизнь канала

Обязательность. Обязательно: ядро

Профиль. open-alert-draft/0.21

Полномочия. Аутентифицированный участник в своем канале.

Запрос. id и bootId в оболочке; владелец передает также controlEpoch. Дочернего элемента и cmdSeq нет.

Ответ. ok, bootId и channel uptimeMs/leaseRemainingMs. Для наблюдателя leaseRemainingMs=0.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, staleEpoch, staleAttachment

Повторы и поведение. Отдельный слот heartbeat, период 5000 мс, lease 30000 мс. Только запрос прикрепленного владельца с текущей эпохой продлевает связь. Наблюдатели не продлевают владение.

A.37 session.start - Начать оповещение, опрос с запуском или проверку

Обязательность. Обязательно: ядро

Профиль. open-alert-draft/0.21

Полномочия. Текущий владелец; поддержанные ресурсы свободны и имеют допустимое физическое состояние.

Запрос. bootId/controlEpoch/cmdSeq; session mode=siren|recording|test|program|stream, purpose, acceptBeforeMs, maxDurationMs, configRevision, capabilityRevision; resources с 1..4 resource id/bindingGeneration и ровно один элемент соответствующего режима. Форматы приведены в разделах 12, 15, 19 и 26.

Ответ. accepted с session id/startSeq/state/onControlLoss/configRevision/capabilityRevision/acceptedAtMs/maxDurationMs. Принятие и физический результат раздельны; наблюдения и проверки читаются страницами session.get.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, staleEpoch, staleAttachment, conflict, resultExpired, unsupported, startWindowExpired, startBlockedByCancel, sourceUnavailable, unsafeState, notFound, storageFault

Повторы и поведение. Точный повтор cmdSeq в том же bootId/controlEpoch возвращает сохраненное решение без повторного действия. Если решение и сведения об адресной отмене вытеснены, старый номер получает resultExpired. onControlLoss не переопределяется в session.start. Отмена без ответа: session.cancel по startSeq; с известным ID: session.stop. Все поля конкретного режима и быстрые обязательные предусловия проверяются до воздействия по текущим наблюдениям.

Старое заключение readiness.get не резервирует ресурс и не заменяет повторной проверки.

A.38 session.stop - Остановить известный сеанс

Обязательность. Обязательно: ядро

Профиль. open-alert-draft/0.21

Полномочия. Текущий владелец именно этого исполнения.

Запрос. bootId/controlEpoch, новый cmdSeq и точный sessionId.

Ответ. Принятие остановки/state=stopping; затем результат исполнения и отдельно подтвержденное состояние выходов.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, staleEpoch, staleAttachment, conflict, resultExpired, notFound, storageFault

Повторы и поведение. Точный повтор cmdSeq в том же bootId/controlEpoch возвращает сохраненное решение без повторного действия. Если решение и сведения об адресной отмене вытеснены, старый номер получает resultExpired. Разделяет отдельный приоритетный слот с session.cancel и не ожидает ответа обычного чтения или сервисной команды. Старый sessionId никогда не означает "остановить текущий". Возврат accepted еще не доказывает физического выключения. Отмена удаляет будущие включения и повторы этого исполнения, отзывает его медиаразрешения; обработка session.stop НЕ ДОЛЖНА оставлять задержанный session.start в драйвере.

A.39 session.cancel - Отменить неподтвержденный session.start

Обязательность. Обязательно: ядро

Профиль. open-alert-draft/0.21

Полномочия. Текущий владелец соответствующего bootId/controlEpoch.

Запрос. Новый cmdSeq и startSeq исходного session.start; 0 < startSeq < cmdSeq. sessionId знать НЕ ТРЕБУЕТСЯ.

Ответ. Совпадающий активный сеанс переходит в stopping; иначе activeMatch=false, lateStartBlocked=true и честное состояние доступности истории.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, staleEpoch, staleAttachment, conflict, resultExpired, storageFault, startBlockedByCancel

Повторы и поведение. Точный повтор cmdSeq в том же bootId/controlEpoch возвращает сохраненное решение без повторного действия. Если решение и сведения об адресной отмене вытеснены, старый номер получает resultExpired. Поднимает highWater и блокирует поздний запуск с указанным старым номером. Сохраненная отмена позволяет ответить startBlockedByCancel с cancelSeq, когда решение самого session.start отсутствует; после вытеснения истории остается resultExpired. Чужой или более новый сеанс не затрагивает. Решение по резервному номеру завершения не вытесняется до освобождения эпохи; условия допуска и нерасходования номера определены разделом 6.

A.40 session.get - Узнать состояние и итог сеанса

Обязательность. Обязательно: ядро

Профиль. open-alert-draft/0.21

Полномочия. Участник с правом чтения соответствующего исполнения.

Запрос. bootId в оболочке; selection sessionId либо startSeq (для startSeq также controlEpoch в оболочке); при snapshotReplace=1 допустим replaces по разделу 21. Продолжение: те же поля выбора и пара snapshotId/cursor без replaces; одно поле без другого запрещено.

Ответ. Неизменный result; одна session на первой странице, одна timeline и одна completion на весь снимок, resource/check/observation/execution/fault и для program все stageResult. Времена, полнота и result/reason определены разделами 8, 19 и 26.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, notFound, resultExpired, snapshotExpired

Повторы и поведение. Повтор разрешен; это новое чтение, поэтому данные могут измениться. Если команда использует снимок, все его страницы читаются с одним snapshotId. result содержит snapshotId, stateRevision, evaluatedAtMs, more и cursor только при more=true; сводка session есть только на первой странице; mode и счетчики resourceCount/checkCount/observationCount/executionCount/faultCount неизменны на всех страницах, далее целые timeline/resource/check/observation/execution/completion/fault/stageResult. Размер с оболочкой до 1024 байт. Конец страницы не является концом сеанса. Неизвестный исторический результат не выводится из текущего состояния устройства.

A.41 command.get - Уточнить решение по команде

Обязательность. Дополнительная возможность

Профиль. open-alert-draft/0.21

Полномочия. Владелец контекста либо явно разрешенная диагностическая роль.

Запрос. selection controlEpoch/cmdSeq; bootId в оболочке.

Ответ. command cmdSeq/state=accepted|rejected|expired; action и исходное тело решения при сохраненной истории по разделу 13.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, resultExpired, staleEpoch

Повторы и поведение. Доступен только участнику исходной команды. Не исполняет ее повторно; expired не означает отсутствия воздействия. Сохраненное тело возвращается без исходных транспортных to/id.

A.42 events.read - Получить события после курсора

Обязательность. Обязательно: ядро

Профиль. open-alert-draft/0.21

Полномочия. Читатель с правом полного журнала данного УУ; частичное право датчиков не дает доступа к базовому журналу.

Запрос. Элемент events с bootId и afterSeq. Полей snapshotId/cursor и запрашиваемого размера страницы в ядре нет; размер ответа ограничивает УУ.

Ответ. Контейнер events с nextSeq, more, gap и oldestSeq; внутри записи event с seq и stateRevision. more относится к текущей выборке. Возобновление связывается с границей eventSeq согласованного снимка.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy

Повторы и поведение. Повтор с тем же afterSeq читает текущий доступный диапазон журнала; неизменность страницы не гарантируется. Эта команда читает журнал по запросу и сама не включает подписку. Инициативная доставка отдельно согласуется через events.subscribe при eventsPush=1. Продолжение задает afterSeq=nextSeq предыдущего обработанного ответа, без snapshotId/cursor. При gap=true записей нет, more=false, nextSeq=afterSeq; нужен новый согласованный снимок. Курсор сохраняется после проверки и обработки страницы; повторы seq не создают повторных событий, исторические события снимком не восстанавливаются.

A.43 config.get - Прочитать действующую конфигурацию

Обязательность. Дополнительная возможность

Профиль. open-alert-draft/0.21

Полномочия. Право чтения конфигурации.

Запрос. selection configId либо transactionId или snapshotId/cursor.

Ответ. configuration snapshotId/configRevision/appliedRevision/more/cursor; entry либо transaction по разделу 16.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, snapshotExpired, storageFault

Повторы и поведение. Повтор разрешен; это новое чтение, поэтому данные могут измениться. Если команда использует снимок, все его страницы читаются с одним snapshotId.

A.44 config.stage - Подготовить конфигурацию

Обязательность. Дополнительная возможность

Профиль. open-alert-draft/0.21

Полномочия. Действующий владелец с административным правом, подтвержденный off затрагиваемых ресурсов; раздел 16.

Запрос. transaction id/baseRevision/totalBytes/hashAlgorithm/hash и transfer profile/blockBytes/windowBlocks/windowRoundTripMaxMs; cmdSeq maintenance; open-alert-config/5.

Ответ. transaction id/state/receivedBytes/remainingBudgetMs; новый прием receiving, повтор возвращает текущее состояние.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, staleEpoch, staleAttachment, conflict, resultExpired, noSpace, storageFault, unsafeState

Повторы и поведение. Байты передаются отдельным каналом bulk/1. Полный прием запускает validating; config.get сообщает прогресс и staged/failed. Проверка ограничена 60000 мс, порции не более 100 мс; действующая конфигурация не меняется до commit.

A.45 config.commit - Применить подготовленные настройки

Обязательность. Дополнительная возможность

Профиль. open-alert-draft/0.21

Полномочия. Действующий владелец с административным правом, подтвержденный off затрагиваемых ресурсов; раздел 16.

Запрос. transaction id; обычный cmdSeq.

Ответ. committed id/configRevision/appliedRevision; атомарное применение полного пакета.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, staleEpoch, staleAttachment, conflict, resultExpired, notFound, storageFault, unsafeState

Повторы и поведение. Атомарно сохраняет и применяет только локальные правила получателя; configRevision=appliedRevision. Пакет не выполняет физическую перенастройку ОУ. Несовпадение baseRevision отклоняется до фиксации; повтор возвращает сохраненное решение.

A.46 config.cancel - Отбросить подготовленные настройки

Обязательность. Дополнительная возможность

Профиль. open-alert-draft/0.21

Полномочия. Действующий владелец с административным правом, подтвержденный off затрагиваемых ресурсов; раздел 16.

Запрос. transaction id; обычный cmdSeq.

Ответ. cancelled id; зафиксированная конфигурация сохраняется.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, staleEpoch, staleAttachment, conflict, resultExpired, notFound, storageFault

Повторы и поведение. Повтор возвращает сохраненное решение; cancel после commit дает conflict.

A.47 network.stage - Подготовить смену сети

Обязательность. Дополнительная возможность

Профиль. open-alert-draft/0.21

Полномочия. Действующий владелец с административным правом, подтвержденный off затрагиваемых ресурсов; раздел 16.

Запрос. network id/address/prefix/gateway/confirmMs; обычный cmdSeq.

Ответ. staged id/expiresAtMs; пробное применение с автоматическим возвратом по разделу 16.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, staleEpoch, staleAttachment, conflict, resultExpired, unsafeState, storageFault

Повторы и поведение. Пробные параметры применяются после сохранения прежних; возврат по expiresAtMs или cancel. Повтор не продлевает срок. Невозможность надежного возврата отклоняет stage до изменения адреса.

A.48 network.confirm - Подтвердить новый сетевой путь

Обязательность. Дополнительная возможность

Профиль. open-alert-draft/0.21

Полномочия. Действующий владелец с административным правом, подтвержденный off затрагиваемых ресурсов; раздел 16.

Запрос. network id; обычный cmdSeq; действующий канал того же principalId и режима tls либо plain через новый адрес до expiresAtMs.

Ответ. network id/state=committed.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, staleEpoch, staleAttachment, conflict, resultExpired, notFound, confirmationWindowExpired, storageFault

Повторы и поведение. Новый канал проходит protocol.hello и control.resume; после истечения эпохи требуется новое maintenance-владение того же участника. Режим фиксируется network.stage; смена канала или эпохи не продлевает expiresAtMs. Повтор в доступной истории возвращает решение. Опоздавший confirm не применяет параметры повторно.

A.49 network.cancel - Отменить сетевой переход

Обязательность. Дополнительная возможность

Профиль. open-alert-draft/0.21

Полномочия. Действующий владелец с административным правом, подтвержденный off затрагиваемых ресурсов; раздел 16.

Запрос. network id; обычный cmdSeq.

Ответ. network id/state=cancelled; восстановлены прежние параметры.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, staleEpoch, staleAttachment, conflict, resultExpired, notFound, storageFault

Повторы и поведение. Точный повтор cmdSeq в том же bootId/controlEpoch возвращает сохраненное решение без повторного действия. Если решение и сведения об адресной отмене вытеснены, старый номер получает resultExpired. Если сеть потеряна, возврат обеспечивается таймером устройства, а не доставкой этой команды.

A.50 time.set - Установить время

Обязательность. Дополнительная возможность

Профиль. open-alert-draft/0.21

Полномочия. Действующий владелец с административным правом, подтвержденный off затрагиваемых ресурсов; раздел 16.

Запрос. clock utcMs/errorMs; errorMs=0..500; обычный cmdSeq.

Ответ. clock utcMs/errorMs; монотонные таймеры неизменны.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, staleEpoch, staleAttachment, conflict, resultExpired

Повторы и поведение. Точный повтор cmdSeq в том же bootId/controlEpoch возвращает сохраненное решение без повторного действия. Если решение и сведения об адресной отмене вытеснены, старый номер получает resultExpired. Коррекция настенных часов не меняет монотонные таймеры сеанса и срок сборки сообщения.

A.51 firmware.begin - Начать прием образа

Обязательность. Дополнительная возможность

Профиль. open-alert-draft/0.21

Полномочия. Действующий владелец с административным правом, подтвержденный off затрагиваемых ресурсов; раздел 16.

Запрос. image id/deviceKind/hardwareRevision/version/bytes/hashAlgorithm/hash/signature/keyId и transfer profile/blockBytes/windowBlocks/windowRoundTripMaxMs; cmdSeq maintenance; upload.id=image.id.

Ответ. upload id=image.id/state=receiving/receivedBytes/remainingBudgetMs. Повтор begin возвращает тот же ID и текущий префикс.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, staleEpoch, staleAttachment, conflict, resultExpired, incompatible, noSpace, unsafeState, invalidSignature, rollbackDenied, storageFault

Повторы и поведение. Не активирует образ. Байты передаются отдельным каналом bulk/1 того же явно выбранного режима tls либо plain с согласованными

блоком и окном. Общий срок рассчитывается по числу окон; idleTimeoutMs=600000. Подтвержденный префикс и ID сохраняются при смене владения и reboot по разделу 16.

A.52 firmware.verify - Проверить принятый образ

Обязательность. Дополнительная возможность

Профиль. open-alert-draft/0.21

Полномочия. Действующий владелец с административным правом, подтвержденный off затрагиваемых ресурсов; раздел 16.

Запрос. upload id; обычный cmdSeq.

Ответ. upload id/state=verified либо accepted с operation id=upload.id; firmware.status возвращает verifying/verified/failed; progressBytes присутствует только в verifying.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, staleEpoch, staleAttachment, conflict, resultExpired, incomplete, invalidSignature, incompatible, rollbackDenied, unsafeState, notFound, storageFault

Повторы и поведение. Точный повтор cmdSeq в том же bootId/controlEpoch возвращает сохраненное решение без повторного действия. Если решение и сведения об адресной отмене вытеснены, старый номер получает resultExpired. Успешный хеш без проверки подлинности не разрешает установку.

A.53 firmware.commit - Активировать проверенный образ

Обязательность. Дополнительная возможность

Профиль. open-alert-draft/0.21

Полномочия. Действующий владелец с административным правом, подтвержденный off затрагиваемых ресурсов; раздел 16.

Запрос. upload id; обычный cmdSeq; состояние verified и подтвержденный off.

Ответ. accepted, operation id=upload.id. Допуск требует местного разрешения обслуживания и готового независимого тракта тревоги; итог через firmware.status.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, staleEpoch, staleAttachment, conflict, resultExpired, unsafeState, incompatible, incomplete, invalidSignature, rollbackDenied, notFound, storageFault

Повторы и поведение. Точный повтор cmdSeq в том же bootId/controlEpoch возвращает сохраненное решение без повторного действия. Если решение и сведения об адресной отмене вытеснены, старый номер получает resultExpired. Ответ до перезагрузки не доказывает успешную загрузку. Активация сохраняет независимый путь восстановления при потере питания по разделу 16.

A.54 firmware.status - Уточнить состояние обновления

Обязательность. Дополнительная возможность

Профиль. open-alert-draft/0.21

Полномочия. Разрешенный участник операции обновления.

Запрос. selection operationId; это тот же ID, что upload.id=image.id, с самого firmware.begin.

Ответ. firmware operationId/state/receivedBytes/version/reason/ remainingBudgetMs; состояния с receiving до booted/cancelled по разделу 16.

Ошибки. invalidArgument, unsupportedVersion, incompatibleProfile, forbidden, staleBoot, busy, unsupported, notFound, resultExpired, storageFault

Повторы и поведение. Повтор разрешен; это новое чтение, поэтому данные могут измениться. Если команда использует снимок, все его страницы читаются с одним snapshotId. Результат операции сохраняется в постоянном журнале загрузчика и доступен по operationId после смены bootId.

Приложение Б

(обязательное для ядра)

Машинные ошибки

Таблица 74 - Машинные ошибки

Код	**Смысл**
incompatibleProfile / staleAttachment	Нет общего профиля / номер прикрепления устарел либо уже использован другим каналом.
unsupportedVersion / unsupported	Не поддержана версия или функция. Нельзя подменять запрошенный режим другим.
invalidArgument	Поле, сочетание полей или диапазон неверны; действий нет.
forbidden	Нет права действия, области либо управления чужим исполнением; detail обязателен: roleDenied, scopeDenied, ownerMismatch.
staleBoot / staleEpoch	Контекст загрузки или владения устарел. Старое поручение не переносится в новый контекст.
busy / unsafeState	busy: разрешенная операция конфликтует с владением, исполнением, местным управлением, сервисом или квотой; detail обязателен. unsafeState: состояние физически небезопасно, в том числе недоказанный off.
startWindowExpired	Отказ приема START: истек срок приема нового запуска.
confirmationWindowExpired	Отказ подтверждения сетевой настройки: истек срок подтверждения.

Окончание таблицы 74

Код	**Смысл**
maxDurationExceeded / stopUnconfirmed	Причины результата уже принятого исполнения: исчерпан общий предел / выключение не подтверждено. Это не отказ приема START и не доказательство успешного окончания.
startBlockedByCancel	Поздний START блокирован сохраненной отменой cancelSeq. Код не доказывает, что до отмены исполнения не было.
conflict	Несовпадение содержимого повтора, ревизии или состояния операции. Причина уточняется машинным detail.
resultExpired / notFound	Решение уже недоступно / ссылка не найдена в доступной области. Ни одно не доказывает, что действие исторически не происходило.
snapshotExpired / cursorExpired	Снимок или курсор недействителен; нужна новая последовательность чтения.
sourceUnavailable / storageFault	Источник или хранилище недоступны; последующие попытки требуют оценки состояния.
incompatible / incomplete / noSpace	Неподходящий образ, незавершенная передача или недостаточный ресурс.
invalidSignature / rollbackDenied	Не подтверждена подлинность либо версия не разрешена политикой отката.

Приложение B

(справочное)

Примеры XML

B.1 Подготовка файла PCMA/16000

Выбранный mediaProfile объявил sampleRates с частотой 16000. Пример использует прежнюю пакетизацию 20 мс и поток записи длительностью одну секунду.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000000ab01" action="audio.prepare"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="11">
  <stream startup="outputReady"
key="111111111111111111111111111111111111111111111111111111111111111"
    source="recording" sampleRate="16000" totalSamples="16000" codec="PCMA"
    packetMs="20" ssrc="01234567" firstSeq="0" firstTimestamp="0">
    <resources>
      <resource id="ou1" bindingGeneration="2"/>
    </resources>
    <route sourceAddress="192.0.2.1" destinationAddress="192.0.2.2"
      sourcePort="5004" destinationPort="5004" ttl="1"/>
    <buffer initialMs="400" maxMs="1000"/>
  </stream>
</request>
```

B.2 Подготовка файла PCMA/32000

Выбранный mediaProfile объявил sampleRates с частотой 32000. Пример использует прежнюю пакетизацию 20 мс и поток записи длительностью одну секунду.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000000ab02" action="audio.prepare"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="11">
  <stream startup="outputReady"
key="1111111111111111111111111111111111111111111111111111111111111111"
    source="recording" sampleRate="32000" totalSamples="32000" codec="PCMA"
    packetMs="20" ssrc="01234567" firstSeq="0" firstTimestamp="0">
    <resources>
      <resource id="ou1" bindingGeneration="2"/>
    </resources>
    <route sourceAddress="192.0.2.1" destinationAddress="192.0.2.2"
      sourcePort="5004" destinationPort="5004" ttl="1"/>
    <buffer initialMs="400" maxMs="1000"/>
  </stream>
</request>
```

B.3 Подготовка речи с ожиданием ОУ

Выбранный mediaProfile rtp-alert/5, tls, PCMA/20 объявил startup=outputReady. После UDP READY и принятого session.start АРМ ожидает audio.state outputReady=true перед захватом микрофона.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000000aa01" action="audio.prepare"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="11">
  <stream startup="outputReady"
key="1111111111111111111111111111111111111111111111111111111111111111"
    source="microphone" codec="PCMA" packetMs="20" ssrc="01234567" firstSeq="0"
    firstTimestamp="0">
    <resources>
      <resource id="ou1" bindingGeneration="2"/>
    </resources>
    <route sourceAddress="192.0.2.1" destinationAddress="192.0.2.2"
      sourcePort="5004" destinationPort="5004" ttl="1"/>
    <buffer initialMs="400" maxMs="1000"/>
  </stream>
</request>
```

B.4 Речевой выход готов к приему источника

Ответ audio.state после свежего подтверждения речевого режима ОУ. Источник еще не запущен, running и исполнение отдельно подтверждаются через session.get.

```
<response v="1" profile="open-alert-draft/0.21"
  to="0000000000000000000000000000aa02"
  bootId="00000000000000000000000000000073" status="ok">
  <audio mediaId="00000000000000000000000000000071"
    sessionId="00000000000000000000000000000072" state="buffering"
    outputReady="true" receivedPackets="0" lostPackets="0" latePackets="0"
    duplicatePackets="0" emittedSamples="0" bufferedMs="0" lastPacketAtMs="0">
    <clock sourceSamples="0" concealedSamples="0" rateCorrectionPpm="0"
      state="locking"/>
  </audio>
</response>
```

B.5 Начальный обмен: предложение профиля

Сценарий fragment-000; начальное состояние определяется только этим фрагментом и его пояснением. Первый прикладной запрос после взаимной TLS-аутентификации.

```
<request v="1" profile="bootstrap/1" id="00000000000000000000000000003000"
  action="protocol.hello">
  <hello>
    <profile id="open-alert-draft/0.21"/>
  </hello>
</request>
```

B.6 Начальный обмен: выбор профиля

Сценарий fragment-000; начальное состояние определяется только этим фрагментом и его пояснением. Дальнейшие запросы используют выбранный профиль.

```
<response v="1" profile="bootstrap/1" to="00000000000000000000000000003000"
  bootId="00000000000000000000000000000073" status="ok">
  <hello selectedProfile="open-alert-draft/0.21"
    deviceId="00000000000000000000000000000072" security="tls"/>
</response>
```

B.7 Получение управления: запрос

Сценарий fragment-002; начальное состояние определяется только этим фрагментом и его пояснением. acquireSeq и instanceId сохранены APM до отправки; principalId установлен TLS-контекстом.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000003001" action="control.acquire"
  bootId="00000000000000000000000000000073">
  <control acquireSeq="1" instanceId="00000000000000000000000000000081"
    purpose="exercise"/>
</request>
```

B.8 Получение управления: решение для повторной доставки

Сценарий fragment-002; начальное состояние определяется только этим фрагментом и его пояснением. Потерянный ответ восстанавливается повтором acquireSeq=1; новый канал прикрепляется через resume.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000003001"
  bootId="00000000000000000000000000000073" status="ok">
  <control controlEpoch="00000000000000000000000000000091" acquireSeq="1"
    instanceId="00000000000000000000000000000081" attachSeq="0"
    highWater="0" current="true" leaseRemainingMs="30000" purpose="exercise"
    idleDeadlineMs="60000"/>
</response>
```

B.9 Восстановление управления: привязка нового соединения

Сценарий fragment-004; начальное состояние определяется только этим фрагментом и его пояснением. Та же эпоха, следующий сохраненный attachSeq; прежний канал теряет право изменения.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000003002" action="control.resume"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091">
  <control acquireSeq="1" instanceId="00000000000000000000000000000081"
    attachSeq="1"/>
</request>
```

B.10 Восстановление управления: ответ

Сценарий fragment-004; начальное состояние определяется только этим фрагментом и его пояснением. highWater относится к сохраненной истории эпохи.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000003002"
  bootId="00000000000000000000000000000073" status="ok">
  <control controlEpoch="00000000000000000000000000000091" acquireSeq="1"
    instanceId="00000000000000000000000000000081" attachSeq="1"
    highWater="1" current="true" leaseRemainingMs="30000" purpose="exercise"
    idleDeadlineMs="60000"/>
</response>
```

B.11 Освобождение управления: запрос

Сценарий fragment-006; начальное состояние определяется только этим фрагментом и его пояснением. Допустим после завершения исполнения и подтвержденного off. Пример предполагает, что команда 2 уже обработана.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000003003" action="control.release"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="3"/>
```

B.12 Освобождение управления: ответ

Сценарий fragment-006; начальное состояние определяется только этим фрагментом и его пояснением. Повтор решения не восстанавливает завершенную эпоху.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000003003"
  bootId="00000000000000000000000000000073" status="ok">
  <released controlEpoch="00000000000000000000000000000091" released="true"/>
</response>
```

B.13 Проверка связи владельца

Сценарий fragment-008; начальное состояние определяется только этим фрагментом и его пояснением. Самостоятельный пример действующей эпохи. Отдельный слот и heartbeatIntervalMs=5000 мс.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000003004" action="channel.ping"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091"/>
```

B.14 Проверка связи: ответ

Сценарий fragment-008; начальное состояние определяется только этим фрагментом и его пояснением. Монотонное время УУ и остаток lease на момент ответа.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000003004"
  bootId="00000000000000000000000000000073" status="ok">
  <channel uptimeMs="50000" leaseRemainingMs="30000"/>
</response>
```

В.15 Запуск: встроенная запись

Сценарий fragment-010; начальное состояние определяется только этим фрагментом и его пояснением. Отдельная трасса. Версия и Стрибог-256 содержимого обязательны; один проход.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000003005" action="session.start"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="3">
  <session mode="recording" purpose="exercise" acceptBeforeMs="60000"
    maxDurationMs="30000" configRevision="7" capabilityRevision="4">
    <resources>
      <resource id="ou1" bindingGeneration="2"/>
    </resources>
    <recording contentId="message1" revision="1" hashAlgorithm="streebog256"
hash="aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"
    />
  </session>
</request>
```

B.16 Запуск: опрос с запуском

Сценарий fragment-011; начальное состояние определяется только этим фрагментом и его пояснением. Самостоятельная ветвь: controlEpoch заранее получен с purpose=technical. Отдельная трасса. testId и testRevision ссылаются на tests.list; purpose=technical.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000003006" action="session.start"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="3">
  <session mode="test" purpose="technical" acceptBeforeMs="60000"
    maxDurationMs="10000" configRevision="7" capabilityRevision="4">
    <resources>
      <resource id="ou1" bindingGeneration="2"/>
    </resources>
    <test testId="activation1" testRevision="1"/>
  </session>
</request>
```

B.17 Результат сеанса: первая страница

Сценарий fragment-012; начальное состояние определяется только этим фрагментом и его пояснением. Чтение сохраненного результата session.start 1 в заданной эпохе.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000003007" action="session.get"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091">
  <selection startSeq="1"/>
</request>
```

B.18 Результат сеанса: результат и продолжение

Сценарий fragment-012; начальное состояние определяется только этим фрагментом и его пояснением. Снимок результата фиксирован. more=true требует продолжения даже при finished.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000003007"
  bootId="00000000000000000000000000000073" status="ok">
  <result snapshotId="00000000000000000000000000000094" stateRevision="40"
    more="true" cursor="page2" resourceCount="1" checkCount="1"
    observationCount="0" mode="siren" evaluatedAtMs="72000" executionCount="1"
    faultCount="0">
    <session id="00000000000000000000000000000581" startSeq="1"
      state="finished" onControlLoss="stop" configRevision="7" capabilityRevision="4"
      acceptedAtMs="50000" maxDurationMs="30000" result="succeeded"
      reason="completed"/>
    <resource id="ou1" output="off" evidence="reportedMode" observedAtMs="71000"
      bindingGeneration="2"/>
  </result>
</response>
```

B.19 Результат сеанса: запрос продолжения

Сценарий fragment-014; начальное состояние определяется только этим фрагментом и его пояснением. Выбор сеанса и snapshotId сохраняются.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000003008" action="session.get"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091">
  <selection startSeq="1" snapshotId="00000000000000000000000000000094"
    cursor="page2"/>
</request>
```

B.20 Результат сеанса: конец результата

Сценарий fragment-014; начальное состояние определяется только этим фрагментом и его пояснением. Последняя страница того же снимка; сводка session уже получена на первой. Счетчики относятся ко всему результату. timeline задает реальные границы исполнения; prepareTimeoutMs=2000, stopBeginMs=100, stopConfirmMs=3000.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000003008"
  bootId="00000000000000000000000000000073" status="ok">
  <result snapshotId="00000000000000000000000000000094" stateRevision="40"
    more="true" resourceCount="1" checkCount="1" observationCount="0"
    mode="siren" evaluatedAtMs="72000" executionCount="1" faultCount="0"
    cursor="proof">
    <check id="outputOff" status="passed" resourceId="ou1" phase="stopped"/>
    <timeline sessionId="00000000000000000000000000000581"
      prepareDeadlineMs="52000" runningAtMs="50100" runDeadlineMs="70100"
      stoppingAtMs="70100" stopIssuedAtMs="70120" stopBeginDeadlineMs="70200"
      stopConfirmDeadlineMs="73100" finishedAtMs="71100"/>
  </result>
</response>
```

B.21 Описание устройства: запрос

Сценарий fragment-016; начальное состояние определяется только этим фрагментом и его пояснением. После выбора профиля; владение НЕ ТРЕБУЕТСЯ.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000003100" action="device.describe"/>
```

B.22 Описание устройства: ответ

Сценарий fragment-016; начальное состояние определяется только этим фрагментом и его пояснением. Монотонное время относится к bootId ответа.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000003100"
  bootId="00000000000000000000000000000073" status="ok">
  <device deviceId="00000000000000000000000000000072" kind="vau" version="1.0"
    uptimeMs="50000" securityPolicy="optional"/>
  <info protocolName="Открытый протокол систем оповещения"/>
</response>
```

B.23 Возможности устройства: первая страница

Сценарий fragment-018; начальное состояние определяется только этим фрагментом и его пояснением. Запрос всего каталога возможностей УУ.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000003101" action="device.capabilities"/>
```

B.24 Возможности устройства: страница возможностей

Сценарий fragment-018; начальное состояние определяется только этим фрагментом и его пояснением. more=true: перечисление продолжается; эта страница не объявляет полный набор команд и лимитов.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000003101"
  bootId="00000000000000000000000000000073" status="ok">
  <capabilities snapshotId="00000000000000000000000000000095"
    capabilityRevision="4" more="true" cursor="page2">
    <mode id="siren"/>
    <command name="session.start"/>
    <resource id="ou1" ouId="ou1" bindingGeneration="2" conflictGroup="amplifier1"/>
    <limit name="heartbeatIntervalMs" value="5000"/>
  </capabilities>
</response>
```

B.25 Состояние устройства: запрос снимка

Сценарий fragment-020; начальное состояние определяется только этим фрагментом и его пояснением. Отдельное наблюдение состояния, без захвата владения.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000003102" action="device.state"/>
```

B.26 Состояние устройства: снимок и граница событий

Сценарий fragment-020; начальное состояние определяется только этим фрагментом и его пояснением. Исходная точка для последующего events.read afterSeq=207.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000003102"
  bootId="00000000000000000000000000000073" status="ok">
  <state snapshotId="00000000000000000000000000000096" stateRevision="31"
    eventSeq="207" capabilityRevision="4" uptimeMs="50000" resourceCount="1"
    alarmCount="0" sessionCount="0" more="false">
    <configuration configRevision="7" appliedRevision="7" onControlLoss="finish"/>
    <ownership state="free"/>
    <recovery state="ready"/>
    <resource id="ou1" output="off" evidence="reportedMode" observedAtMs="49900"/>
  </state>
</response>
```

B.27 Каталог: запрос описания точки

Сценарий fragment-022; начальное состояние определяется только этим фрагментом и его пояснением. Постоянное описание читается отдельно от измерений.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000002004" action="inventory.get">
  <selection pointId="ou1.mainsMissing"/>
</request>
```

B.28 Каталог: одна страница описания

Сценарий fragment-022; начальное состояние определяется только этим фрагментом и его пояснением. Самостоятельная запись facet=source укладывается в лимит атрибутов. more=true и cursor требуют следующей страницы того же snapshotId; это еще не полный каталог.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000002004"
  bootId="00000000000000000000000000000073" status="ok">
  <inventory snapshotId="00000000000000000000000000000093"
    capabilityRevision="4" bindingGeneration="2" more="true" cursor="page2">
    <point id="ou1.mainsMissing" facet="source" sourceKind="ouReply"
      presence="present" support="supported"/>
  </inventory>
</response>
```

B.29 Датчик: чтение сохраненного показания

Сценарий fragment-024; начальное состояние определяется только этим фрагментом и его пояснением. Чтение точки не запускает функциональный тест и не меняет возраст показания.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000002001" action="sensors.read">
  <selection>
    <point id="ou1.mainsMissing"/>
  </selection>
</request>
```

B.30 Датчик: значение и его качество

Сценарий fragment-024; начальное состояние определяется только этим фрагментом и его пояснением. complete относится к выбранному набору из одной точки. capabilityRevision обозначает версию состава каталога. observedAtMs - время получения наблюдения в монотонной шкале УУ; численные интервалы иллюстративны.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000002001"
  bootId="00000000000000000000000000000073" status="ok">
  <snapshot snapshotId="00000000000000000000000000000092"
    bindingGeneration="2" more="false" capabilityRevision="4" stateRevision="31"
    eventSeq="207" evaluatedAtMs="50000" complete="true">
    <point id="ou1.mainsMissing" value="true" validity="valid" freshness="fresh"
      sourceId="ou1" sourceKind="ouReply" sampleSeq="23" observedAtMs="49900"
      changedAtMs="43000" maxAgeMs="2000" type="bool"/>
  </snapshot>
</response>
```

B.31 События: запрос после обработанного номера

Сценарий fragment-026; начальное состояние определяется только этим фрагментом и его пояснением. Обычный запрос pull-доставки. Курсор относится к указанному bootId.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000002003" action="events.read">
  <events bootId="00000000000000000000000000000073" afterSeq="207"/>
</request>
```

B.32 Потеря взаимодействия с ОУ: событие в ответе на чтение

Сценарий fragment-026; начальное состояние определяется только этим фрагментом и его пояснением. Последнее аварийное значение сохранено. validity=unavailable и freshness=stale запрещают считать его свежим подтверждением состояния; sourceLost не очищает аварию. nextSeq сохраняется после обработки страницы.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000002003"
  bootId="00000000000000000000000000000073" status="ok">
  <events nextSeq="208" more="false" gap="false" oldestSeq="180">
    <event seq="208" stateRevision="32" capabilityRevision="4"
      type="sensor.qualityChanged">
      <point id="ou1.mainsMissing" value="true" validity="unavailable"
        freshness="stale" reason="sourceLost" sourceId="ou1"
        observedAtMs="49900" evaluatedAtMs="52500" type="bool"/>
    </event>
  </events>
</response>
```

B.33 Запуск: конечная сирена

Сценарий fragment-028; начальное состояние определяется только этим фрагментом и его пояснением. Полный session.start сирены. acceptBeforeMs относится к uptimeMs данного bootId; конфигурация и каталог зафиксированы ревизиями. onControlLoss=stop заранее сохранен на УУ.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000001042"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="1"
  action="session.start">
  <session mode="siren" purpose="exercise" acceptBeforeMs="60000"
    maxDurationMs="30000" configRevision="7" capabilityRevision="4">
    <resources>
      <resource id="ou1" bindingGeneration="2"/>
    </resources>
    <siren pattern="continuous" durationMs="20000" stopMode="duration"/>
  </session>
</request>
```

B.34 Принятие запуска

Сценарий fragment-028; начальное состояние определяется только этим фрагментом и его пояснением. accepted подтверждает прием. Переход в running и физические наблюдения ОУ передаются отдельно.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000001042"
  bootId="00000000000000000000000000000073" status="accepted">
  <session id="00000000000000000000000000000581" startSeq="1" state="starting"
    onControlLoss="stop" configRevision="7" capabilityRevision="4"
    acceptedAtMs="50000" maxDurationMs="30000"/>
</response>
```

B.35 Отмена после потери ответа на запуск

Сценарий fragment-030; начальное состояние определяется только этим фрагментом и его пояснением. Новый номер 2 отменяет session.start с номером 1 в том же bootId/controlEpoch.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000001045"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="2"
  action="session.cancel">
  <session startSeq="1"/>
</request>
```

B.36 Отмена без активного совпадения

Сценарий fragment-030; начальное состояние определяется только этим фрагментом и его пояснением. История неизвестна. Такой ответ не означает, что сирена ранее не включалась.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000001045"
  bootId="00000000000000000000000000000073" status="accepted">
  <cancellation startSeq="1" activeMatch="false" lateStartBlocked="true"
    history="unknown"/>
</response>
```

B.37 Остановка: sessionId известен

Сценарий fragment-032; начальное состояние определяется только этим фрагментом и его пояснением. Альтернатива cancel, а не следующий запрос с повторным номером 2 в той же последовательности.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000001044"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="2"
  action="session.stop">
  <session id="00000000000000000000000000000581"/>
</request>
```

B.38 Отказ: ресурс занят

Сценарий fragment-033; начальное состояние определяется только этим фрагментом и его пояснением. Альтернативный ответ на показанный session.start 1042; не следует после его принятия в той же трассе. АРМ принимает решение по устойчивому машинному коду; человекочитаемый текст не служит условием алгоритма.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000001042"
  bootId="00000000000000000000000000000073" status="error">
  <error code="busy" resource="session" detail="executionActive"/>
</response>
```

B.39 Аварии: запрос полного состояния

Сценарий fragment-034; начальное состояние определяется только этим фрагментом и его пояснением. Первая страница снимка перед восстановлением журнала.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000004100" action="device.state"/>
```

B.40 Аварии: контекст и ресурс

Сценарий fragment-034; начальное состояние определяется только этим фрагментом и его пояснением. alarmCount=1 относится ко всему снимку. До конца страниц список аварий не считается восстановленным.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000004100"
  bootId="00000000000000000000000000000073" status="ok">
  <state snapshotId="00000000000000000000000000000410" stateRevision="31"
    eventSeq="207" capabilityRevision="4" uptimeMs="50000" resourceCount="1"
    alarmCount="1" sessionCount="0" more="true" cursor="alarmPage">
    <configuration configRevision="7" appliedRevision="7" onControlLoss="finish"/>
    <ownership state="free"/>
    <recovery state="ready"/>
    <resource id="ou1" output="off" evidence="reportedMode" observedAtMs="49900"/>
  </state>
</response>
```

B.41 Аварии: запрос продолжения

Сценарий fragment-036; начальное состояние определяется только этим фрагментом и его пояснением. Тот же snapshotId и полученный cursor.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000004101" action="device.state">
  <selection snapshotId="00000000000000000000000000000410"
    cursor="alarmPage"/>
</request>
```

B.42 Аварии: активный эпизод из снимка

Сценарий fragment-036; начальное состояние определяется только этим фрагментом и его пояснением. Это существующая авария, а не новый фронт. Теперь получены все счетчики снимка.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000004101"
  bootId="00000000000000000000000000000073" status="ok">
  <state snapshotId="00000000000000000000000000000410" stateRevision="31"
    eventSeq="207" capabilityRevision="4" uptimeMs="50000" resourceCount="1"
    alarmCount="1" sessionCount="0" more="false">
    <alarm conditionId="mainsLoss"
      alarmId="00000000000000000000000000000600" pointId="ou1.mainsMissing"
      state="active" severity="critical" validity="valid" freshness="fresh"
      observedAtMs="49900" changedAtMs="43000" continuity="continuous"/>
  </state>
</response>
```

B.43 Аварии: события после снимка

Сценарий fragment-038; начальное состояние определяется только этим фрагментом и его пояснением. Запрос продолжает границу eventSeq=207 из полного снимка.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000004102" action="events.read">
  <events bootId="00000000000000000000000000000073" afterSeq="207"/>
</request>
```

B.44 Аварии: подтвержденное снятие

Сценарий fragment-038; начальное состояние определяется только этим фрагментом и его пояснением. Сохраняются conditionId и alarmId; новый stateRevision, полная запись качества.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000004102"
  bootId="00000000000000000000000000000073" status="ok">
  <events nextSeq="208" more="false" gap="false" oldestSeq="180">
    <event seq="208" stateRevision="32" capabilityRevision="4" type="alarm.cleared">
      <alarm conditionId="mainsLoss"
        alarmId="00000000000000000000000000000600" pointId="ou1.mainsMissing"
        state="cleared" severity="critical" validity="valid" freshness="fresh"
        observedAtMs="51000" changedAtMs="51000" continuity="continuous"/>
    </event>
  </events>
</response>
```

B.45 Перечисление: запрос каталога точки

Сценарий fragment-040; начальное состояние определяется только этим фрагментом и его пояснением. Каталог enum читается вместе со всеми четырьмя facet.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000004103" action="inventory.get">
  <selection pointId="ou1.powerMode"/>
</request>
```

B.46 Перечисление: описание точки

Сценарий fragment-040; начальное состояние определяется только этим фрагментом и его пояснением. Первая страница каталога; окончание еще не получено.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000004103"
  bootId="00000000000000000000000000000073" status="ok">
  <inventory snapshotId="00000000000000000000000000000411"
    capabilityRevision="4" bindingGeneration="2" more="true" cursor="enumPage">
    <point id="ou1.powerMode" facet="identity" resourceId="ou1" ouId="ou1"/>
    <point id="ou1.powerMode" facet="meaning" kind="powerMode" type="enum"/>
    <point id="ou1.powerMode" facet="source" sourceKind="ouReply"
      presence="present" support="supported"/>
  </inventory>
</response>
```

B.47 Перечисление: продолжение с тем же фильтром

Сценарий fragment-042; начальное состояние определяется только этим фрагментом и его пояснением. Нельзя потерять pointId при передаче cursor.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000004104" action="inventory.get">
  <selection pointId="ou1.powerMode"
    snapshotId="00000000000000000000000000000411" cursor="enumPage"/>
</request>
```

B.48 Перечисление: список значений

Сценарий fragment-042; начальное состояние определяется только этим фрагментом и его пояснением. Ключ (pointId, code); label - текст для отображения.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000004104"
  bootId="00000000000000000000000000000073" status="ok">
  <inventory snapshotId="00000000000000000000000000000411"
    capabilityRevision="4" bindingGeneration="2" more="false">
    <point id="ou1.powerMode" facet="acquisition" readMethod="cached"
      maxAgeMs="2000"/>
    <enum pointId="ou1.powerMode" code="mains" label="Основное питание"/>
    <enum pointId="ou1.powerMode" code="battery" label="Резервное питание"/>
  </inventory>
</response>
```

B.49 Порог: запрос каталога

Сценарий fragment-044; начальное состояние определяется только этим фрагментом и его пояснением. Числовая точка и связанное с ней условие.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000004105" action="inventory.get">
  <selection pointId="ou1.temperature"/>
</request>
```

B.50 Порог: постоянное описание

Сценарий fragment-044; начальное состояние определяется только этим фрагментом и его пояснением. Продолжение содержит способ чтения и порог.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000004105"
  bootId="00000000000000000000000000000073" status="ok">
  <inventory snapshotId="00000000000000000000000000000412"
    capabilityRevision="4" bindingGeneration="2" more="true" cursor="thresholdPage">
    <point id="ou1.temperature" facet="identity" resourceId="ou1" ouId="ou1"/>
    <point id="ou1.temperature" facet="meaning" kind="temperature" type="int"
      unit="C" scale="1"/>
    <point id="ou1.temperature" facet="source" sourceKind="ouReply"
      presence="present" support="supported"/>
  </inventory>
</response>
```

B.51 Порог: запрос продолжения

Сценарий fragment-046; начальное состояние определяется только этим фрагментом и его пояснением. Тот же фильтр и снимок.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000004106" action="inventory.get">
  <selection pointId="ou1.temperature"
    snapshotId="00000000000000000000000000000412" cursor="thresholdPage"/>
</request>
```

B.52 Порог: условие перегрева

Сценарий fragment-046; начальное состояние определяется только этим фрагментом и его пояснением. gt=80, снятие ниже 75. conditionId связывает описание условия и эпизоды alarm.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000004106"
  bootId="00000000000000000000000000000073" status="ok">
  <inventory snapshotId="00000000000000000000000000000412"
    capabilityRevision="4" bindingGeneration="2" more="false">
    <point id="ou1.temperature" facet="acquisition" readMethod="cached"
      maxAgeMs="2000"/>
    <threshold pointId="ou1.temperature" conditionId="overheat" operator="gt"
      value="80" hysteresis="5" severity="critical"/>
  </inventory>
</response>
```

B.53 Результат опроса с задействованием: запрос 1

Сценарий fragment-048; начальное состояние определяется только этим фрагментом и его пояснением. Часть полного комплекта execution-0.11.json. Сводка только на первой странице.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000001770" action="session.get"
  bootId="00000000000000000000000000000073">
  <selection sessionId="000000000000000000000000000016a9"/>
</request>
```

B.54 Результат опроса с задействованием: страница 1

Сценарий fragment-048; начальное состояние определяется только этим фрагментом и его пояснением. Полный фиксированный набор включает ресурс, проверки и три фазы наблюдения.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000001770"
  bootId="00000000000000000000000000000073" status="ok">
  <result snapshotId="000000000000000000000000000001f4" stateRevision="50"
    mode="test" resourceCount="1" checkCount="2" observationCount="3" more="true"
    cursor="p1" evaluatedAtMs="55000" executionCount="1" faultCount="0">
    <session id="000000000000000000000000000016a9" startSeq="3"
      state="finished" onControlLoss="stop" configRevision="7" capabilityRevision="4"
      acceptedAtMs="50000" maxDurationMs="10000" result="succeeded"
      reason="completed"/>
  </result>
</response>
```

B.55 Результат опроса с задействованием: запрос 2

Сценарий fragment-050; начальное состояние определяется только этим фрагментом и его пояснением. Часть полного комплекта execution-0.11.json. Сводка только на первой странице.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000001771" action="session.get"
  bootId="00000000000000000000000000000073">
  <selection sessionId="000000000000000000000000000016a9"
    snapshotId="000000000000000000000000000001f4" cursor="p1"/>
</request>
```

B.56 Результат опроса с задействованием: страница 2

Сценарий fragment-050; начальное состояние определяется только этим фрагментом и его пояснением. Полный фиксированный набор включает ресурс, проверки и три фазы наблюдения. timeline относится ко всему тесту; prepareTimeoutMs=2000, stopBeginMs=100, stopConfirmMs=3000.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000001771"
  bootId="00000000000000000000000000000073" status="ok">
  <result snapshotId="000000000000000000000000000001f4" stateRevision="50"
    mode="test" resourceCount="1" checkCount="2" observationCount="3" more="true"
    cursor="p2" evaluatedAtMs="55000" executionCount="1" faultCount="0">
    <resource id="ou1" bindingGeneration="2" output="off" evidence="reportedMode"
      observedAtMs="54000"/>
    <timeline sessionId="000000000000000000000000000016a9"
      prepareDeadlineMs="52000" runningAtMs="51000" runDeadlineMs="60000"
      stoppingAtMs="52000" stopIssuedAtMs="52020" stopBeginDeadlineMs="52100"
      stopConfirmDeadlineMs="55000" finishedAtMs="54000"/>
  </result>
</response>
```

B.57 Результат опроса с задействованием: запрос 3

Сценарий fragment-052; начальное состояние определяется только этим фрагментом и его пояснением. Часть полного комплекта execution-0.11.json. Сводка только на первой странице.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000001772" action="session.get"
  bootId="00000000000000000000000000000073">
  <selection sessionId="000000000000000000000000000016a9"
    snapshotId="000000000000000000000000000001f4" cursor="p2"/>
</request>
```

B.58 Результат опроса с задействованием: страница 3

Сценарий fragment-052; начальное состояние определяется только этим фрагментом и его пояснением. Полный фиксированный набор включает ресурс, проверки и три фазы наблюдения.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000001772"
  bootId="00000000000000000000000000000073" status="ok">
  <result snapshotId="000000000000000000000000000001f4" stateRevision="50"
    mode="test" resourceCount="1" checkCount="2" observationCount="3" more="true"
    cursor="p3" evaluatedAtMs="55000" executionCount="1" faultCount="0">
    <check id="activation" resourceId="ou1" phase="running" status="passed"/>
  </result>
</response>
```

B.59 Результат опроса с задействованием: запрос 4

Сценарий fragment-054; начальное состояние определяется только этим фрагментом и его пояснением. Часть полного комплекта execution-0.11.json. Сводка только на первой странице.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000001773" action="session.get"
  bootId="00000000000000000000000000000073">
  <selection sessionId="000000000000000000000000000016a9"
    snapshotId="000000000000000000000000000001f4" cursor="p3"/>
</request>
```

B.60 Результат опроса с задействованием: страница 4

Сценарий fragment-054; начальное состояние определяется только этим фрагментом и его пояснением. Полный фиксированный набор включает ресурс, проверки и три фазы наблюдения.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000001773"
  bootId="00000000000000000000000000000073" status="ok">
  <result snapshotId="000000000000000000000000000001f4" stateRevision="50"
    mode="test" resourceCount="1" checkCount="2" observationCount="3" more="true"
    cursor="p4" evaluatedAtMs="55000" executionCount="1" faultCount="0">
    <check id="outputOff" resourceId="ou1" phase="stopped" status="passed"/>
  </result>
</response>
```

B.61 Результат опроса с задействованием: запрос 5

Сценарий fragment-056; начальное состояние определяется только этим фрагментом и его пояснением. Часть полного комплекта execution-0.11.json. Сводка только на первой странице.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000001774" action="session.get"
  bootId="00000000000000000000000000000073">
  <selection sessionId="000000000000000000000000000016a9"
    snapshotId="000000000000000000000000000001f4" cursor="p4"/>
</request>
```

B.62 Результат опроса с задействованием: страница 5

Сценарий fragment-056; начальное состояние определяется только этим фрагментом и его пояснением. Полный фиксированный набор включает ресурс, проверки и три фазы наблюдения.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000001774"
  bootId="00000000000000000000000000000073" status="ok">
  <result snapshotId="000000000000000000000000000001f4" stateRevision="50"
    mode="test" resourceCount="1" checkCount="2" observationCount="3" more="true"
    cursor="p5" evaluatedAtMs="55000" executionCount="1" faultCount="0">
    <observation id="baselineMode" resourceId="ou1" pointId="ou1.active"
      phase="baseline" type="bool" sourceId="ou1" sourceKind="ouReply"
      value="false" validity="valid" freshness="fresh" observedAtMs="50000"/>
  </result>
</response>
```

B.63 Результат опроса с задействованием: запрос 6

Сценарий fragment-058; начальное состояние определяется только этим фрагментом и его пояснением. Часть полного комплекта execution-0.11.json. Сводка только на первой странице.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000001775" action="session.get"
  bootId="00000000000000000000000000000073">
  <selection sessionId="000000000000000000000000000016a9"
    snapshotId="000000000000000000000000000001f4" cursor="p5"/>
</request>
```

B.64 Результат опроса с задействованием: страница 6

Сценарий fragment-058; начальное состояние определяется только этим фрагментом и его пояснением. Полный фиксированный набор включает ресурс, проверки и три фазы наблюдения.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000001775"
  bootId="00000000000000000000000000000073" status="ok">
  <result snapshotId="000000000000000000000000000001f4" stateRevision="50"
    mode="test" resourceCount="1" checkCount="2" observationCount="3" more="true"
    cursor="p6" evaluatedAtMs="55000" executionCount="1" faultCount="0">
    <observation id="runningMode" resourceId="ou1" pointId="ou1.active"
      phase="running" type="bool" sourceId="ou1" sourceKind="ouReply" value="true"
      validity="valid" freshness="fresh" observedAtMs="51000"/>
  </result>
</response>
```

B.65 Результат опроса с задействованием: запрос 7

Сценарий fragment-060; начальное состояние определяется только этим фрагментом и его пояснением. Часть полного комплекта execution-0.11.json. Сводка только на первой странице.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000001776" action="session.get"
  bootId="00000000000000000000000000000073">
  <selection sessionId="000000000000000000000000000016a9"
    snapshotId="000000000000000000000000000001f4" cursor="p6"/>
</request>
```

B.66 Результат опроса с задействованием: страница 7

Сценарий fragment-060; начальное состояние определяется только этим фрагментом и его пояснением. Полный фиксированный набор включает ресурс, проверки и три фазы наблюдения.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000001776"
  bootId="00000000000000000000000000000073" status="ok">
  <result snapshotId="000000000000000000000000000001f4" stateRevision="50"
    mode="test" resourceCount="1" checkCount="2" observationCount="3" more="true"
    evaluatedAtMs="55000" executionCount="1" faultCount="0" cursor="proof">
    <observation id="stoppedMode" resourceId="ou1" pointId="ou1.active"
      phase="stopped" type="bool" sourceId="ou1" sourceKind="ouReply"
      value="false" validity="valid" freshness="fresh" observedAtMs="53000"/>
  </result>
</response>
```

B.67 Отмена: резерв последнего номера остановки

Сценарий fragment-062; начальное состояние определяется только этим фрагментом и его пояснением. session.start может занять последний обычный номер; резерв отмены остается доступным.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000000fde" action="session.cancel"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091"
  cmdSeq="18446744073709551614">
  <session startSeq="18446744073709551613"/>
</request>
```

B.68 Освобождение управления: зарезервированный номер

Сценарий fragment-063; начальное состояние определяется только этим фрагментом и его пояснением. Допустим только после подтвержденного off и сохранения закрытого результата.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000000fdf" action="control.release"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091"
  cmdSeq="18446744073709551615"/>
```

B.69 Программа: сирена и локальная речь

Сценарий fragment-064; начальное состояние определяется только этим фрагментом и его пояснением. Самостоятельная ветвь: controlEpoch заранее получен с purpose=real. Полный неизменный план принят до воздействия; новые session.start для его этапов не требуются.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000000fe0" action="session.start"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="10">
  <session mode="program" purpose="real" acceptBeforeMs="60000"
    maxDurationMs="120000" configRevision="7" capabilityRevision="4">
    <resources>
      <resource id="ou1" bindingGeneration="2"/>
    </resources>
    <program id="alarm1" revision="1"/>
  </session>
</request>
```

B.70 Паспорт физического исполнения

Сценарий fragment-065; начальное состояние определяется только этим фрагментом и его пояснением. Формат команды профиля 0.21.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000000fe1" action="profiles.get"
  bootId="00000000000000000000000000000073">
  <selection resourceId="ou1"/>
</request>
```

B.71 Каталог автономных программ

Сценарий fragment-066; начальное состояние определяется только этим фрагментом и его пояснением. Формат команды профиля 0.21.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000000fe2" action="programs.list"
  bootId="00000000000000000000000000000073">
  <selection programId="alarm1"/>
</request>
```

B.72 Каталог критериев теста

Сценарий fragment-067; начальное состояние определяется только этим фрагментом и его пояснением. Формат команды профиля 0.21.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000000fe3" action="tests.list"
  bootId="00000000000000000000000000000073">
  <selection testId="activation1"/>
</request>
```

B.73 Речевой поток: подготовка микрофона

Сценарий fragment-068; начальное состояние определяется только этим фрагментом и его пояснением. Самостоятельная речевая ветвь: controlEpoch получен с purpose=real. Формат команды профиля 0.21.

```
<request v="1" profile="open-alert-draft/0.21"
  id="000000000000000000000000000000fe4" action="audio.prepare"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="11">
  <stream
key="1111111111111111111111111111111111111111111111111111111111111111"
    source="microphone" codec="PCMA" packetMs="20" ssrc="01234567" firstSeq="0"
    firstTimestamp="0">
    <resources>
      <resource id="ou1" bindingGeneration="2"/>
    </resources>
    <route sourceAddress="192.0.2.1" destinationAddress="192.0.2.2"
      sourcePort="5004" destinationPort="5004" ttl="1"/>
    <buffer initialMs="400" maxMs="1000"/>
  </stream>
</request>
```

B.74 Речевой поток: запуск подготовленного потока

Сценарий fragment-069; начальное состояние определяется только этим фрагментом и его пояснением. Самостоятельная ветвь: controlEpoch заранее получен с purpose=real. Формат команды профиля 0.21.

```
<request v="1" profile="open-alert-draft/0.21"
  id="000000000000000000000000000000fe5" action="session.start"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="12">
  <session mode="stream" purpose="real" acceptBeforeMs="60000"
    maxDurationMs="60000" configRevision="7" capabilityRevision="4">
    <resources>
      <resource id="ou1" bindingGeneration="2"/>
    </resources>
    <stream mediaId="00000000000000000000000000000601"/>
  </session>
</request>
```

B.75 Речевой поток: конец источника

Сценарий fragment-070; начальное состояние определяется только этим фрагментом и его пояснением. Последний из 50 пакетов PCMA имеет END=1 и validSamples=160; sampleCount=8000. Альтернатива после уже отправленного полного пакета без END: отдельный маркер index=50, validSamples=0 и audio.finish lastPacketIndex=50. Квитанция EOF не является подтверждением off.

```
<request v="1" profile="open-alert-draft/0.21"
  id="000000000000000000000000000000fe6" action="audio.finish"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="13">
  <stream mediaId="00000000000000000000000000000601" lastPacketIndex="49"
    sampleCount="8000"/>
</request>
```

B.76 Опрос: воспроизводимое создание задания

Сценарий fragment-071; начальное состояние определяется только этим фрагментом и его пояснением. Формат команды профиля 0.21.

```
<request v="1" profile="open-alert-draft/0.21"
  id="000000000000000000000000000000fe7" action="poll.start"
  bootId="00000000000000000000000000000073">
  <poll clientOperationId="00000000000000000000000000000601"
    deadlineMs="60000" refresh="cache">
    <point id="ou1.active"/>
  </poll>
</request>
```

B.77 Файловый канал: запрос возможностей

Сценарий fragment-072; начальное состояние определяется только этим фрагментом и его пояснением. Пример bulk/1; права, контекст и сроки проверяются по разделу 16.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000001f41" action="transfer.capabilities">
</request>
```

B.78 Файловый канал: возможности

Сценарий fragment-072; начальное состояние определяется только этим фрагментом и его пояснением. Пример bulk/1; права, контекст и сроки проверяются по разделу 16.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000001f41"
  bootId="00000000000000000000000000000073" status="ok">
  <transfer profile="bulk/1" port="10002" maxBlockBytes="4096"
    maxWindowBlocks="4" maxInFlightBytes="16384"/>
</response>
```

B.79 Размещение записи: начало bulk/1

Сценарий fragment-074; начальное состояние определяется только этим фрагментом и его пояснением. 4096 нулевых байтов и хвост 01 02 03; хеш соответствует этим 4099 байтам PCMA. Пример не воспроизводится до отдельного session.start.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000001f42" action="content.begin"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="1">
  <content id="00000000000000000000000000000802" revision="1" codec="PCMA"
    bytes="4099" sampleCount="4099" hashAlgorithm="streebog256"
hash="707422b4e615499d38be2bd279d807d6bebb426e6fdc0c8c1a711bbb34ea25a7"/>
  <transfer profile="bulk/1" blockBytes="4096" windowBlocks="4"
    windowRoundTripMaxMs="100"/>
</request>
```

B.80 Размещение записи: прием создан

Сценарий fragment-074; начальное состояние определяется только этим фрагментом и его пояснением. Пример bulk/1; права, контекст и сроки проверяются по разделу 16.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000001f42"
  bootId="00000000000000000000000000000073" status="ok">
  <upload id="00000000000000000000000000000801" receivedBytes="0"
    remainingBudgetMs="600100"/>
</response>
```

B.81 Файловое соединение: привязка

Сценарий fragment-076; начальное состояние определяется только этим фрагментом и его пояснением. Пример bulk/1; права, контекст и сроки проверяются по разделу 16.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000001f43" action="transfer.hello"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091">
  <transfer id="00000000000000000000000000000801" kind="content"/>
</request>
```

B.82 Файловое соединение: готовность

Сценарий fragment-076; начальное состояние определяется только этим фрагментом и его пояснением. Пример bulk/1; права, контекст и сроки проверяются по разделу 16.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000001f43"
  bootId="00000000000000000000000000000073" status="ok">
  <ready id="00000000000000000000000000000801" blockBytes="4096"
    windowBlocks="4" receivedBytes="0" remainingBudgetMs="600100"/>
</response>
```

B.83 Размещение записи: чтение состояния

Сценарий fragment-078; начальное состояние определяется только этим фрагментом и его пояснением. Пример bulk/1; права, контекст и сроки проверяются по разделу 16.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000001f44" action="content.status">
  <selection uploadId="00000000000000000000000000000801"/>
</request>
```

B.84 Размещение записи: все байты сохранены

Сценарий fragment-078; начальное состояние определяется только этим фрагментом и его пояснением. Полный ACK подтверждает префикс. Публикация неизменной записи требует content.commit и проверки хеша/формата.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000001f44"
  bootId="00000000000000000000000000000073" status="ok">
  <upload id="00000000000000000000000000000801" state="complete"
    receivedBytes="4099" totalBytes="4099" remainingBudgetMs="600000"/>
</response>
```

B.85 Конфигурация: создание приема bulk/1

Сценарий fragment-080; начальное состояние определяется только этим фрагментом и его пояснением. Передается канонический JSON-пакет config из conformance-0.21.json. Длина и хеш проверяются по его байтам; пример относится к отдельной транзакции.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000001f45" action="config.stage"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="1">
  <transaction id="00000000000000000000000000000803" baseRevision="1"
    totalBytes="3263" hashAlgorithm="streebog256"
hash="3cf889bc83878dbe47ede2042acb3be4a209bd919fc04e9e4a6f6fd93a561b02"/>
  <transfer profile="bulk/1" blockBytes="4096" windowBlocks="4"
    windowRoundTripMaxMs="100"/>
</request>
```

B.86 Сирена: запрос подтверждения исполнения

Сценарий fragment-081; начальное состояние определяется только этим фрагментом и его пояснением. Продолжение того же полного снимка.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000c3a1" action="session.get"
  bootId="00000000000000000000000000000073">
  <selection sessionId="00000000000000000000000000000581"
    snapshotId="00000000000000000000000000000094" cursor="proof"/>
</request>
```

B.87 Сирена: подтверждение исполнения и полнота

Сценарий fragment-081; начальное состояние определяется только этим фрагментом и его пояснением. Подтверждение начала хранится отдельно от завершения.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000c3a1"
  bootId="00000000000000000000000000000073" status="ok">
  <result snapshotId="00000000000000000000000000000094" stateRevision="40"
    more="false" resourceCount="1" checkCount="1" observationCount="0"
    mode="siren" evaluatedAtMs="72000" executionCount="1" faultCount="0">
    <execution resourceId="ou1" status="confirmed" criterionId="outputActive"
      sourceId="ou1" evidence="reportedMode" observedAtMs="50100"/>
    <completion state="complete"/>
  </result>
</response>
```

B.88 Тест: запрос подтверждения исполнения

Сценарий fragment-083; начальное состояние определяется только этим фрагментом и его пояснением. Продолжение того же полного снимка.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000c3a3" action="session.get"
  bootId="00000000000000000000000000000073">
  <selection sessionId="00000000000000000000000000016a9"
    snapshotId="00000000000000000000000000001f4" cursor="proof"/>
</request>
```

B.89 Тест: подтверждение исполнения и полнота

Сценарий fragment-083; начальное состояние определяется только этим фрагментом и его пояснением. Подтверждение начала хранится отдельно от завершения.

```
<response v="1" profile="open-alert-draft/0.21"
  to="0000000000000000000000000000c3a3"
  bootId="00000000000000000000000000000073" status="ok">
  <result snapshotId="00000000000000000000000000001f4" stateRevision="50"
    mode="test" resourceCount="1" checkCount="2" observationCount="3"
    more="false" evaluatedAtMs="55000" executionCount="1" faultCount="0">
    <execution resourceId="ou1" status="confirmed" criterionId="outputActive"
      sourceId="ou1" evidence="reportedMode" observedAtMs="51000"/>
    <completion state="complete"/>
  </result>
</response>
```

B.90 Отказ: forbidden/roleDenied

Сценарий fragment-085; начальное состояние определяется только этим фрагментом и его пояснением. Отказ до воздействия. Сведения о другом владельце не передаются.

```
<response v="1" profile="open-alert-draft/0.21"
  to="0000000000000000000000000000c78d"
  bootId="00000000000000000000000000000073" status="error">
  <error code="forbidden" detail="roleDenied"/>
</response>
```

B.91 Отказ: forbidden/scopeDenied

Сценарий fragment-086; начальное состояние определяется только этим фрагментом и его пояснением. Отказ до воздействия. Сведения о другом владельце не передаются.

```
<response v="1" profile="open-alert-draft/0.21"
  to="0000000000000000000000000000c78e"
  bootId="00000000000000000000000000000073" status="error">
  <error code="forbidden" detail="scopeDenied"/>
</response>
```

B.92 Отказ: forbidden/ownerMismatch

Сценарий fragment-087; начальное состояние определяется только этим фрагментом и его пояснением. Отказ до воздействия. Сведения о другом владельце не передаются.

```
<response v="1" profile="open-alert-draft/0.21"
  to="0000000000000000000000000000c78f"
  bootId="00000000000000000000000000000073" status="error">
  <error code="forbidden" detail="ownerMismatch"/>
</response>
```

B.93 Отказ: busy/ownershipHeld

Сценарий fragment-088; начальное состояние определяется только этим фрагментом и его пояснением. Отказ до воздействия. Сведения о другом владельце не передаются.

```
<response v="1" profile="open-alert-draft/0.21"
  to="0000000000000000000000000000c790"
  bootId="00000000000000000000000000000073" status="error">
  <error code="busy" detail="ownershipHeld"/>
</response>
```

B.94 Отказ: busy/executionActive

Сценарий fragment-089; начальное состояние определяется только этим фрагментом и его пояснением. Отказ до воздействия. Сведения о другом владельце не передаются.

```
<response v="1" profile="open-alert-draft/0.21"
  to="0000000000000000000000000000c791"
  bootId="00000000000000000000000000000073" status="error">
  <error code="busy" detail="executionActive"/>
</response>
```

B.95 Отказ: busy/localControl

Сценарий fragment-090; начальное состояние определяется только этим фрагментом и его пояснением. Отказ до воздействия. Сведения о другом владельце не передаются.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000c792"
  bootId="00000000000000000000000000000073" status="error">
  <error code="busy" detail="localControl"/>
</response>
```

B.96 Отказ: busy/serviceActive

Сценарий fragment-091; начальное состояние определяется только этим фрагментом и его пояснением. Отказ до воздействия. Сведения о другом владельце не передаются.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000c793"
  bootId="00000000000000000000000000000073" status="error">
  <error code="busy" detail="serviceActive"/>
</response>
```

B.97 Отказ: busy/capacityExhausted

Сценарий fragment-092; начальное состояние определяется только этим фрагментом и его пояснением. Отказ до воздействия. Сведения о другом владельце не передаются.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000c794"
  bootId="00000000000000000000000000000073" status="error">
  <error code="busy" detail="capacityExhausted"/>
</response>
```

B.98 Запуск: сирена до команды оператора

Сценарий fragment-093; начальное состояние определяется только этим фрагментом и его пояснением. Длительность не подменяется большим таймером; maxDurationMs остается пределом безопасности.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000002715"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="1"
  action="session.start">
  <session mode="siren" purpose="exercise" acceptBeforeMs="60000"
    maxDurationMs="30000" configRevision="7" capabilityRevision="4">
    <resources>
      <resource id="ou1" bindingGeneration="2"/>
    </resources>
    <siren pattern="continuous" stopMode="command"/>
  </session>
</request>
```

B.99 Событие: подтверждено начало работы

Сценарий fragment-094; начальное состояние определяется только этим фрагментом и его пояснением. Сеанс еще выполняется. Получение события не закрывает его и не подтверждает выключение.

```
<response v="1" profile="open-alert-draft/0.21"
  to="0000000000000000000000000000c796"
  bootId="00000000000000000000000000000073" status="ok">
  <events nextSeq="41" oldestSeq="1" more="false" gap="false">
    <event seq="41" stateRevision="41" type="session.execution"
      capabilityRevision="4">
      <execution sessionId="00000000000000000000000000000581"
        resourceId="ou1" status="confirmed" criterionId="outputActive" sourceId="ou1"
        evidence="reportedMode" observedAtMs="50100"/>
    </event>
  </events>
</response>
```

B.100 Событие: потеря фазы после начала

Сценарий fragment-095; начальное состояние определяется только этим фрагментом и его пояснением. Альтернативная аварийная ветвь. Ранее полученное confirmed сохраняется; текущий статус аварийный.

```
<response v="1" profile="open-alert-draft/0.21"
  to="0000000000000000000000000000c797"
  bootId="00000000000000000000000000000073" status="ok">
  <events nextSeq="42" oldestSeq="1" more="false" gap="false">
    <event seq="42" stateRevision="42" type="session.fault" capabilityRevision="4">
      <fault sessionId="00000000000000000000000000000581"
        id="00000000000000000000000000000981" code="phaseLost"
        resourceId="ou1" effect="failed" severity="critical" occurredAtMs="52000"
        conditionId="phaseA" alarmId="00000000000000000000000000000982"/>
    </event>
  </events>
</response>
```

B.101 Сирена работала: штатная остановка, страница 1

Сценарий fragment-096; начальное состояние определяется только этим фрагментом и его пояснением. Полный самостоятельный снимок: подтверждение начала, полнота, выключение и неисправность проверяются раздельно. Продолжение читается по snapshotId/cursor предыдущей страницы.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000989680"
  bootId="00000000000000000000000000000073" status="ok">
  <result snapshotId="00000000000000000000000000019258" stateRevision="50"
    mode="siren" evaluatedAtMs="55000" resourceCount="1" checkCount="0"
    observationCount="0" executionCount="1" faultCount="0" more="true" cursor="p1">
    <session id="000000000000000000000000000186a0" startSeq="1"
      state="finished" onControlLoss="stop" configRevision="7" capabilityRevision="4"
      acceptedAtMs="50000" maxDurationMs="30000" result="succeeded"
      reason="operatorStop"/>
  </result>
</response>
```

B.102 Сирена работала: штатная остановка, страница 2

Сценарий fragment-097; начальное состояние определяется только этим фрагментом и его пояснением. Полный самостоятельный снимок: подтверждение начала, полнота, выключение и неисправность проверяются раздельно. Продолжение читается по snapshotId/cursor предыдущей страницы.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000989681"
  bootId="00000000000000000000000000000073" status="ok">
  <result snapshotId="00000000000000000000000000019258" stateRevision="50"
    mode="siren" evaluatedAtMs="55000" resourceCount="1" checkCount="0"
    observationCount="0" executionCount="1" faultCount="0" more="true" cursor="p2">
    <timeline sessionId="000000000000000000000000000186a0"
      prepareDeadlineMs="52000" runningAtMs="50100" runDeadlineMs="70100"
      stopRequestedAtMs="53000" stoppingAtMs="53000" stopIssuedAtMs="53020"
      stopBeginDeadlineMs="53100" stopConfirmDeadlineMs="56000"
      finishedAtMs="54000"/>
  </result>
</response>
```

B.103 Сирена работала: штатная остановка, страница 3

Сценарий fragment-098; начальное состояние определяется только этим фрагментом и его пояснением. Полный самостоятельный снимок: подтверждение начала, полнота, выключение и неисправность проверяются раздельно. Продолжение читается по snapshotId/cursor предыдущей страницы.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000989682"
  bootId="00000000000000000000000000000073" status="ok">
  <result snapshotId="00000000000000000000000000019258" stateRevision="50"
    mode="siren" evaluatedAtMs="55000" resourceCount="1" checkCount="0"
    observationCount="0" executionCount="1" faultCount="0" more="true" cursor="p3">
    <resource id="ou1" bindingGeneration="2" output="off" evidence="reportedMode"
      observedAtMs="54000"/>
  </result>
</response>
```

B.104 Сирена работала: штатная остановка, страница 4

Сценарий fragment-099; начальное состояние определяется только этим фрагментом и его пояснением. Полный самостоятельный снимок: подтверждение начала, полнота, выключение и неисправность проверяются раздельно. Продолжение читается по snapshotId/cursor предыдущей страницы.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000989683"
  bootId="00000000000000000000000000000073" status="ok">
  <result snapshotId="00000000000000000000000000019258" stateRevision="50"
    mode="siren" evaluatedAtMs="55000" resourceCount="1" checkCount="0"
    observationCount="0" executionCount="1" faultCount="0" more="true" cursor="p4">
    <execution resourceId="ou1" status="confirmed" criterionId="outputActive"
      evidence="reportedMode" sourceId="ou1" observedAtMs="50100"/>
  </result>
</response>
```

B.105 Сирена работала: штатная остановка, страница 5

Сценарий fragment-100; начальное состояние определяется только этим фрагментом и его пояснением. Полный самостоятельный снимок: подтверждение начала, полнота, выключение и неисправность проверяются раздельно. Продолжение читается по snapshotId/cursor предыдущей страницы.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000989684"
  bootId="00000000000000000000000000000073" status="ok">
  <result snapshotId="00000000000000000000000000019258" stateRevision="50"
    mode="siren" evaluatedAtMs="55000" resourceCount="1" checkCount="0"
    observationCount="0" executionCount="1" faultCount="0" more="false">
    <completion state="interrupted"/>
  </result>
</response>
```

B.106 Речь началась: файл прерван оператором, страница 1

Сценарий fragment-101; начальное состояние определяется только этим фрагментом и его пояснением. Полный самостоятельный снимок: подтверждение начала, полнота, выключение и неисправность проверяются раздельно. Продолжение читается по snapshotId/cursor предыдущей страницы.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000009897ac"
  bootId="00000000000000000000000000000073" status="ok">
  <result snapshotId="00000000000000000000000001925b" stateRevision="50"
    mode="recording" evaluatedAtMs="55000" resourceCount="1" checkCount="0"
    observationCount="0" executionCount="1" faultCount="0" more="true" cursor="p1">
    <session id="00000000000000000000000000186a3" startSeq="1"
      state="finished" onControlLoss="stop" configRevision="7" capabilityRevision="4"
      acceptedAtMs="50000" maxDurationMs="30000" result="succeeded"
      reason="operatorStop"/>
  </result>
</response>
```

B.107 Речь началась: файл прерван оператором, страница 2

Сценарий fragment-102; начальное состояние определяется только этим фрагментом и его пояснением. Полный самостоятельный снимок: подтверждение начала, полнота, выключение и неисправность проверяются раздельно. Продолжение читается по snapshotId/cursor предыдущей страницы.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000009897ad"
  bootId="00000000000000000000000000000073" status="ok">
  <result snapshotId="0000000000000000000000000001925b" stateRevision="50"
    mode="recording" evaluatedAtMs="55000" resourceCount="1" checkCount="0"
    observationCount="0" executionCount="1" faultCount="0" more="true" cursor="p2">
    <timeline sessionId="00000000000000000000000000186a3"
      prepareDeadlineMs="52000" runningAtMs="50100" runDeadlineMs="70100"
      stopRequestedAtMs="53000" stoppingAtMs="53000" stopIssuedAtMs="53020"
      stopBeginDeadlineMs="53100" stopConfirmDeadlineMs="56000"
      finishedAtMs="54000"/>
  </result>
</response>
```

B.108 Речь началась: файл прерван оператором, страница 3

Сценарий fragment-103; начальное состояние определяется только этим фрагментом и его пояснением. Полный самостоятельный снимок: подтверждение начала, полнота, выключение и неисправность проверяются раздельно. Продолжение читается по snapshotId/cursor предыдущей страницы.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000009897ae"
  bootId="00000000000000000000000000000073" status="ok">
  <result snapshotId="0000000000000000000000000001925b" stateRevision="50"
    mode="recording" evaluatedAtMs="55000" resourceCount="1" checkCount="0"
    observationCount="0" executionCount="1" faultCount="0" more="true" cursor="p3">
    <resource id="ou1" bindingGeneration="2" output="off" evidence="reportedMode"
      observedAtMs="54000"/>
  </result>
</response>
```

B.109 Речь началась: файл прерван оператором, страница 4

Сценарий fragment-104; начальное состояние определяется только этим фрагментом и его пояснением. Полный самостоятельный снимок: подтверждение начала, полнота, выключение и неисправность проверяются раздельно. Продолжение читается по snapshotId/cursor предыдущей страницы.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000009897af"
  bootId="00000000000000000000000000000073" status="ok">
  <result snapshotId="00000000000000000000000000001925b" stateRevision="50"
    mode="recording" evaluatedAtMs="55000" resourceCount="1" checkCount="0"
    observationCount="0" executionCount="1" faultCount="0" more="true" cursor="p4">
    <execution resourceId="ou1" status="confirmed" criterionId="outputActive"
      evidence="reportedMode" sourceId="ou1" observedAtMs="50100"/>
  </result>
</response>
```

B.110 Речь началась: файл прерван оператором, страница 5

Сценарий fragment-105; начальное состояние определяется только этим фрагментом и его пояснением. Полный самостоятельный снимок: подтверждение начала, полнота, выключение и неисправность проверяются раздельно. Продолжение читается по snapshotId/cursor предыдущей страницы.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000009897b0"
  bootId="00000000000000000000000000000073" status="ok">
  <result snapshotId="00000000000000000000000000001925b" stateRevision="50"
    mode="recording" evaluatedAtMs="55000" resourceCount="1" checkCount="0"
    observationCount="0" executionCount="1" faultCount="0" more="false">
    <completion state="interrupted"/>
  </result>
</response>
```

B.111 Сирена работала: последующий отказ фазы, страница 1

Сценарий fragment-106; начальное состояние определяется только этим фрагментом и его пояснением. Полный самостоятельный снимок: подтверждение начала, полнота, выключение и неисправность проверяются раздельно. Продолжение читается по snapshotId/cursor предыдущей страницы.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000989810"
  bootId="00000000000000000000000000000073" status="ok">
  <result snapshotId="00000000000000000000000000001925c" stateRevision="50"
    mode="siren" evaluatedAtMs="55000" resourceCount="1" checkCount="0"
    observationCount="0" executionCount="1" faultCount="1" more="true" cursor="p1">
    <session id="00000000000000000000000000186a4" startSeq="1"
      state="finished" onControlLoss="stop" configRevision="7" capabilityRevision="4"
      acceptedAtMs="50000" maxDurationMs="30000" result="failed"
      reason="equipmentFault"/>
  </result>
</response>
```

B.112 Сирена работала: последующий отказ фазы, страница 2

Сценарий fragment-107; начальное состояние определяется только этим фрагментом и его пояснением. Полный самостоятельный снимок: подтверждение начала, полнота, выключение и неисправность проверяются раздельно. Продолжение читается по snapshotId/cursor предыдущей страницы.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000989811"
  bootId="00000000000000000000000000000073" status="ok">
  <result snapshotId="00000000000000000000000000001925c" stateRevision="50"
    mode="siren" evaluatedAtMs="55000" resourceCount="1" checkCount="0"
    observationCount="0" executionCount="1" faultCount="1" more="true" cursor="p2">
    <timeline sessionId="00000000000000000000000000186a4"
      prepareDeadlineMs="52000" runningAtMs="50100" runDeadlineMs="70100"
      stoppingAtMs="52000" stopIssuedAtMs="52020" stopBeginDeadlineMs="52100"
      stopConfirmDeadlineMs="55000" finishedAtMs="54000"/>
  </result>
</response>
```

B.113 Сирена работала: последующий отказ фазы, страница 3

Сценарий fragment-108; начальное состояние определяется только этим фрагментом и его пояснением. Полный самостоятельный снимок: подтверждение начала, полнота, выключение и неисправность проверяются раздельно. Продолжение читается по snapshotId/cursor предыдущей страницы.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000989812"
  bootId="00000000000000000000000000000073" status="ok">
  <result snapshotId="00000000000000000000000000001925c" stateRevision="50"
    mode="siren" evaluatedAtMs="55000" resourceCount="1" checkCount="0"
    observationCount="0" executionCount="1" faultCount="1" more="true" cursor="p3">
    <resource id="ou1" bindingGeneration="2" output="off" evidence="reportedMode"
      observedAtMs="54000"/>
  </result>
</response>
```

B.114 Сирена работала: последующий отказ фазы, страница 4

Сценарий fragment-109; начальное состояние определяется только этим фрагментом и его пояснением. Полный самостоятельный снимок: подтверждение начала, полнота, выключение и неисправность проверяются раздельно. Продолжение читается по snapshotId/cursor предыдущей страницы.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000989813"
  bootId="00000000000000000000000000000073" status="ok">
  <result snapshotId="00000000000000000000000000001925c" stateRevision="50"
    mode="siren" evaluatedAtMs="55000" resourceCount="1" checkCount="0"
    observationCount="0" executionCount="1" faultCount="1" more="true" cursor="p4">
    <execution resourceId="ou1" status="confirmed" criterionId="outputActive"
      evidence="reportedMode" sourceId="ou1" observedAtMs="50100"/>
  </result>
</response>
```

B.115 Сирена работала: последующий отказ фазы, страница 5

Сценарий fragment-110; начальное состояние определяется только этим фрагментом и его пояснением. Полный самостоятельный снимок: подтверждение начала, полнота, выключение и неисправность проверяются раздельно. Продолжение читается по snapshotId/cursor предыдущей страницы.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000989814"
  bootId="00000000000000000000000000000073" status="ok">
  <result snapshotId="0000000000000000000000000001925c" stateRevision="50"
    mode="siren" evaluatedAtMs="55000" resourceCount="1" checkCount="0"
    observationCount="0" executionCount="1" faultCount="1" more="true" cursor="p5">
    <completion state="interrupted"/>
  </result>
</response>
```

B.116 Сирена работала: последующий отказ фазы, страница 6

Сценарий fragment-111; начальное состояние определяется только этим фрагментом и его пояснением. Полный самостоятельный снимок: подтверждение начала, полнота, выключение и неисправность проверяются раздельно. Продолжение читается по snapshotId/cursor предыдущей страницы.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000989815"
  bootId="00000000000000000000000000000073" status="ok">
  <result snapshotId="0000000000000000000000000001925c" stateRevision="50"
    mode="siren" evaluatedAtMs="55000" resourceCount="1" checkCount="0"
    observationCount="0" executionCount="1" faultCount="1" more="false">
    <fault id="0000000000000000000000000018a8c" resourceId="ou1"
      code="phaseLost" effect="failed" severity="critical" occurredAtMs="52000"
      clearedAtMs="53500" conditionId="phaseA"
      alarmId="00000000000000000000000000000982"/>
  </result>
</response>
```

B.117 Открытый канал: начальный обмен без ключей

Сценарий fragment-116; начальное состояние определяется только этим фрагментом и его пояснением. Клиент явно выбрал plain; securityPolicy УУ допускает открытый TCP. TLS handshake отсутствует.

```
<request v="1" profile="bootstrap/1" id="00000000000000000000000000003f00"
  action="protocol.hello">
  <hello clientId="00000000000000000000000000003f10">
    <profile id="open-alert-draft/0.21"/>
  </hello>
</request>
```

B.118 Открытый канал: подтверждение режима

Сценарий fragment-116; начальное состояние определяется только этим фрагментом и его пояснением. Это согласование формата, а не доказательство подлинности устройства.

```
<response v="1" profile="bootstrap/1" to="00000000000000000000000000003f00"
  bootId="00000000000000000000000000000073" status="ok">
  <hello selectedProfile="open-alert-draft/0.21"
    deviceId="00000000000000000000000000000072" security="plain"/>
</response>
```

B.119 Каталог тестов: совокупность эффектов

Сценарий fragment-118; начальное состояние определяется только этим фрагментом и его пояснением. Все эффекты перечислены в одном отсортированном поле. Критерии и наблюдения следуют на следующих страницах того же снимка.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000003f01"
  bootId="00000000000000000000000000000073" status="ok">
  <tests snapshotId="00000000000000000000000000003f02" capabilityRevision="4"
    more="true" cursor="p2">
    <test id="soundCheck" revision="1" resourceId="ou1" kind="activationPoll"
      maxDurationMs="3000" stopBeginMs="100" stopConfirmMs="3000"
      effects="audible,readBus,switchOutput"/>
  </tests>
</response>
```

В.120 Освободить неиспользованную подготовку: запрос

Самостоятельный сценарий. До запроса: подготовка существует, session.start не принят.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000190000" action="audio.discard"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="1">
  <prepared mediaId="00000000000000000000000000000900"/>
</request>
```

В.121 Освободить неиспользованную подготовку: ответ

Самостоятельный сценарий. До запроса: подготовка существует, session.start не принят.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000190000"
  bootId="00000000000000000000000000000073" status="ok">
  <discarded mediaId="00000000000000000000000000000900"/>
</response>
```

В.122 Подготовить групповую речь: запрос

Самостоятельный сценарий. До запроса: защищенный канал; шаблон speech1/1 установлен; часы имеют погрешность до 500 мс; ключ заведомо тестовый.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000190001" action="audio.group.prepare"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="1">
  <group id="speech1" revision="1"
    streamId="00000000000000000000000000000901" source="microphone"
key="0101010101010101010101010101010101010101010101010101010101010101"/>
</request>
```

B.123 Подготовить групповую речь: ответ

Самостоятельный сценарий. До запроса: защищенный канал; шаблон speech1/1 установлен; часы имеют погрешность до 500 мс; ключ заведомо тестовый.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000190001"
  bootId="00000000000000000000000000000073" status="ok">
  <prepared mediaId="00000000000000000000000000000900"
    expiresAtMs="110000" configRevision="7" capabilityRevision="4"/>
</response>
```

B.124 Прочитать состояние потока: запрос

Самостоятельный сценарий. До запроса: подготовка действительна; session.start еще не принят.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000190002" action="audio.state"
  bootId="00000000000000000000000000000073">
  <selection mediaId="00000000000000000000000000000900"/>
</request>
```

B.125 Прочитать состояние потока: ответ

Самостоятельный сценарий. До запроса: подготовка действительна; session.start еще не принят.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000190002"
  bootId="00000000000000000000000000000073" status="ok">
  <audio mediaId="00000000000000000000000000000900" state="prepared"
    receivedPackets="0" lostPackets="0" latePackets="0" duplicatePackets="0"
    emittedSamples="0" bufferedMs="0" lastPacketAtMs="0"/>
</response>
```

B.126 Уточнить решение по команде: запрос

Самостоятельный сценарий. До запроса: эпоха известна; номер уже использован, подробное решение вытеснено.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000190003" action="command.get"
  bootId="00000000000000000000000000000073">
  <selection controlEpoch="00000000000000000000000000000091" cmdSeq="1"/>
</request>
```

B.127 Уточнить решение по команде: ответ

Самостоятельный сценарий. До запроса: эпоха известна; номер уже использован, подробное решение вытеснено.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000190003"
  bootId="00000000000000000000000000000073" status="ok">
  <command cmdSeq="1" state="expired"/>
</response>
```

B.128 Отбросить подготовленные настройки: запрос

Самостоятельный сценарий. До запроса: незафиксированная транзакция существует; действующая ревизия 7.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000190004" action="config.cancel"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="1">
  <transaction id="00000000000000000000000000000803"/>
</request>
```

B.129 Отбросить подготовленные настройки: ответ

Самостоятельный сценарий. До запроса: незафиксированная транзакция существует; действующая ревизия 7.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000190004"
  bootId="00000000000000000000000000000073" status="ok">
  <cancelled id="00000000000000000000000000000803"/>
</response>
```

B.130 Применить подготовленные настройки: запрос

Самостоятельный сценарий. До запроса: полный пакет получен и проверен; baseRevision=7; ресурсы off.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000190005" action="config.commit"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="1">
  <transaction id="00000000000000000000000000000803"/>
</request>
```

B.131 Применить подготовленные настройки: ответ

Самостоятельный сценарий. До запроса: полный пакет получен и проверен; baseRevision=7; ресурсы off.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000190005"
  bootId="00000000000000000000000000000073" status="ok">
  <committed id="00000000000000000000000000000803" configRevision="8"
    appliedRevision="8"/>
</response>
```

B.132 Прочитать действующую конфигурацию: запрос

Самостоятельный сценарий. До запроса: прием пакета создан, данные еще не переданы.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000190006" action="config.get"
  bootId="00000000000000000000000000000073">
  <selection transactionId="00000000000000000000000000000803"/>
</request>
```

B.133 Прочитать действующую конфигурацию: ответ

Самостоятельный сценарий. До запроса: прием пакета создан, данные еще не переданы.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000190006"
  bootId="00000000000000000000000000000073" status="ok">
  <configuration snapshotId="00000000000000000000000000000912"
    configRevision="7" appliedRevision="7" more="false">
    <transaction id="00000000000000000000000000000803" state="receiving"
      receivedBytes="0" remainingBudgetMs="60000"/>
  </configuration>
</response>
```

B.134 Отменить размещение записи: запрос

Самостоятельный сценарий. До запроса: прием записи существует, commit не принят.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000190007" action="content.cancel"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="1">
  <upload id="00000000000000000000000000000803"/>
</request>
```

B.135 Отменить размещение записи: ответ

Самостоятельный сценарий. До запроса: прием записи существует, commit не принят.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000190007"
  bootId="00000000000000000000000000000073" status="ok">
  <upload id="00000000000000000000000000000803" state="cancelled"/>
</response>
```

B.136 Проверить и опубликовать запись: запрос

Самостоятельный сценарий. До запроса: байты записи получены, хеш и формат проверены.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000190008" action="content.commit"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="1">
  <upload id="00000000000000000000000000000803"/>
</request>
```

B.137 Проверить и опубликовать запись: ответ

Самостоятельный сценарий. До запроса: байты записи получены, хеш и формат проверены.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000190008"
  bootId="00000000000000000000000000000073" status="ok">
  <content id="message1" revision="1" state="committed"/>
</response>
```

B.138 Прочитать манифест и доказательства проверки записи: запрос

Самостоятельный сценарий. До запроса: такой записи в доступной области нет.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000190009" action="content.describe"
  bootId="00000000000000000000000000000073">
  <selection contentId="message1" revision="1"/>
</request>
```

B.139 Прочитать манифест и доказательства проверки записи: ответ

Самостоятельный сценарий. До запроса: такой записи в доступной области нет.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000190009"
  bootId="00000000000000000000000000000073" status="error">
  <error code="notFound"/>
</response>
```

B.140 Инженерная диагностика: запрос

Самостоятельный сценарий. До запроса: в области controller нет объявленных инженерных метрик.

```
<request v="1" profile="open-alert-draft/0.21"
  id="0000000000000000000000000019000a" action="diagnostics.read"
  bootId="00000000000000000000000000000073">
  <selection scope="controller"/>
</request>
```

B.141 Инженерная диагностика: ответ

Самостоятельный сценарий. До запроса: в области controller нет объявленных инженерных метрик.

```
<response v="1" profile="open-alert-draft/0.21"
  to="0000000000000000000000000019000a"
  bootId="00000000000000000000000000000073" status="ok">
  <diagnostics snapshotId="00000000000000000000000000000910"
    uptimeMs="50000" more="false"/>
</response>
```

B.142 Начать прием образа: запрос

Самостоятельный сценарий. До запроса: местное обслуживание разрешено; test-publisher из векторов подписи доверен; тестовый файл abc.

```
<request v="1" profile="open-alert-draft/0.21" id="0000000000000000000000000019000b" action="firmware.begin" bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="1">
  <image id="00000000000000000000000000000803" deviceKind="siren" hardwareRevision="test-board" version="1" bytes="3" hashAlgorithm="streebog256"
    hash="4e2919cf137ed41ec4fb6270c61826cc4fffb660341e0af3688cd0626d23b481"
    signature="a977258936fa94761fb5262eb718e4ae977f19acff0c7c2271dcfb7d999b420207c7b8cd4ada7f51e1651c8fac5fd99e7a52c5cf77f2b76433d2428de3ff4e09" keyId="test-publisher"/>
  <transfer profile="bulk/1" blockBytes="4096" windowBlocks="4" windowRoundTripMaxMs="5000"/>
</request>
```

B.143 Начать прием образа: ответ

Самостоятельный сценарий. До запроса: местное обслуживание разрешено; test-publisher из векторов подписи доверен; тестовый файл abc.

```
<response v="1" profile="open-alert-draft/0.21"
  to="0000000000000000000000000019000b"
  bootId="00000000000000000000000000000073" status="ok">
  <upload id="00000000000000000000000000000803" state="receiving"
    receivedBytes="0" remainingBudgetMs="60000"/>
</response>
```

B.144 Отменить прием или проверку образа: запрос

Самостоятельный сценарий. До запроса: receiving, receivedBytes=0; местное обслуживание разрешено.

```
<request v="1" profile="open-alert-draft/0.21"
  id="0000000000000000000000000019000c" action="firmware.cancel"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="1">
  <upload id="00000000000000000000000000000803"/>
</request>
```

B.145 Отменить прием или проверку образа: ответ

Самостоятельный сценарий. До запроса: receiving, receivedBytes=0; местное обслуживание разрешено.

```
<response v="1" profile="open-alert-draft/0.21"
  to="0000000000000000000000000019000c"
  bootId="00000000000000000000000000000073" status="ok">
  <upload id="00000000000000000000000000000803" state="cancelled"/>
</response>
```

B.146 Активировать проверенный образ: запрос

Самостоятельный сценарий. До запроса: образ verified, off подтвержден, местное разрешение и независимый тракт тревоги проверены.

```
<request v="1" profile="open-alert-draft/0.21"
  id="0000000000000000000000000019000d" action="firmware.commit"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="1">
  <upload id="00000000000000000000000000000803"/>
</request>
```

B.147 Активировать проверенный образ: ответ

Самостоятельный сценарий. До запроса: образ verified, off подтвержден, местное разрешение и независимый тракт тревоги проверены.

```
<response v="1" profile="open-alert-draft/0.21"
  to="0000000000000000000000000019000d"
  bootId="00000000000000000000000000000073" status="accepted">
  <operation id="00000000000000000000000000000803"/>
</response>
```

B.148 Уточнить состояние обновления: запрос

Самостоятельный сценарий. До запроса: манифест firmware-manifest из векторов проверен, активация не начата.

```
<request v="1" profile="open-alert-draft/0.21"
  id="0000000000000000000000000019000e" action="firmware.status"
  bootId="00000000000000000000000000000073">
  <selection operationId="00000000000000000000000000000803"/>
</request>
```

B.149 Уточнить состояние обновления: ответ

Самостоятельный сценарий. До запроса: манифест firmware-manifest из векторов проверен, активация не начата.

```
<response v="1" profile="open-alert-draft/0.21"
  to="0000000000000000000000000019000e"
  bootId="00000000000000000000000000000073" status="ok">
  <firmware operationId="00000000000000000000000000000803" state="verified"
    receivedBytes="3" version="1" remainingBudgetMs="50000"/>
</response>
```

B.150 Проверить принятый образ: запрос

Самостоятельный сценарий. До запроса: три байта abc получены; манифест совпадает с firmware.begin и вектором подписи.

```
<request v="1" profile="open-alert-draft/0.21"
  id="0000000000000000000000000019000f" action="firmware.verify"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="1">
  <upload id="00000000000000000000000000000803"/>
</request>
```

B.151 Проверить принятый образ: ответ

Самостоятельный сценарий. До запроса: три байта abc получены; манифест совпадает с firmware.begin и вектором подписи.

```
<response v="1" profile="open-alert-draft/0.21"
  to="0000000000000000000000000019000f"
  bootId="00000000000000000000000000000073" status="ok">
  <upload id="00000000000000000000000000000803" state="verified"/>
</response>
```

B.152 Журнал устройства: запрос

Самостоятельный сценарий. До запроса: журнал текущей загрузки пуст.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000190010" action="log.read"
  bootId="00000000000000000000000000000073">
  <selection afterSeq="0" limit="8"/>
</request>
```

B.153 Журнал устройства: ответ

Самостоятельный сценарий. До запроса: журнал текущей загрузки пуст.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000190010"
  bootId="00000000000000000000000000000073" status="ok">
  <log oldestSeq="1" nextSeq="1" gap="false" more="false"/>
</response>
```

B.154 Отменить сетевой переход: запрос

Самостоятельный сценарий. До запроса: пробное применение существует и еще не подтверждено.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000190011" action="network.cancel"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="1">
  <network id="00000000000000000000000000000803"/>
</request>
```

B.155 Отменить сетевой переход: ответ

Самостоятельный сценарий. До запроса: пробное применение существует и еще не подтверждено.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000190011"
  bootId="00000000000000000000000000000073" status="ok">
  <network id="00000000000000000000000000000803" state="cancelled"/>
</response>
```

B.156 Подтвердить новый сетевой путь: запрос

Самостоятельный сценарий. До запроса: пробное применение действует до 80000; этот канал установлен на новом адресе.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000190012" action="network.confirm"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="1">
  <network id="00000000000000000000000000000803"/>
</request>
```

B.157 Подтвердить новый сетевой путь: ответ

Самостоятельный сценарий. До запроса: пробное применение действует до 80000; этот канал установлен на новом адресе.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000190012"
  bootId="00000000000000000000000000000073" status="ok">
  <network id="00000000000000000000000000000803" state="committed"/>
</response>
```

B.158 Подготовить смену сети: запрос

Самостоятельный сценарий. До запроса: ресурсы off, местное разрешение обслуживания, текущий адрес 192.0.2.2.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000190013" action="network.stage"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="1">
  <network id="00000000000000000000000000000803" address="192.0.2.3"
    prefix="24" gateway="192.0.2.1" confirmMs="30000"/>
</request>
```

B.159 Подготовить смену сети: ответ

Самостоятельный сценарий. До запроса: ресурсы off, местное разрешение обслуживания, текущий адрес 192.0.2.2.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000190013"
  bootId="00000000000000000000000000000073" status="ok">
  <staged id="00000000000000000000000000000803" expiresAtMs="80000"/>
</response>
```

B.160 Отменить свой опрос: запрос

Самостоятельный сценарий. До запроса: задание принадлежит этому участнику и ожидает опроса.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000190014" action="poll.cancel"
  bootId="00000000000000000000000000000073">
  <poll operationId="00000000000000000000000000000803"
    clientOperationId="00000000000000000000000000000812"/>
</request>
```

B.161 Отменить свой опрос: ответ

Самостоятельный сценарий. До запроса: задание принадлежит этому участнику и ожидает опроса.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000190014"
  bootId="00000000000000000000000000000073" status="ok">
  <cancelled operationId="00000000000000000000000000000803"/>
</response>
```

B.162 Результаты опроса: запрос

Самостоятельный сценарий. До запроса: задание опроса одной точки принято, наблюдение еще не получено.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000190015" action="poll.get"
  bootId="00000000000000000000000000000073">
  <selection operationId="00000000000000000000000000000803"/>
</request>
```

B.163 Результаты опроса: ответ

Самостоятельный сценарий. До запроса: задание опроса одной точки принято, наблюдение еще не получено.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000190015"
  bootId="00000000000000000000000000000073" status="ok">
  <poll snapshotId="00000000000000000000000000000911"
    operationId="00000000000000000000000000000803" state="queued"
    expectedCount="1" complete="false" evaluatedAtMs="50000" more="false"/>
</response>
```

B.164 Оценить готовность конкретного плана: запрос

Самостоятельный сценарий. До запроса: требуемое свежее наблюдение off временно недоступно; воздействия нет.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000190016" action="readiness.get"
  bootId="00000000000000000000000000000073">
  <plan mode="siren" configRevision="7" capabilityRevision="4"
    maxDurationMs="30000">
    <resources>
      <resource id="ou1" bindingGeneration="2"/>
    </resources>
    <siren pattern="continuous" durationMs="20000" stopMode="duration"/>
  </plan>
</request>
```

B.165 Оценить готовность конкретного плана: ответ

Самостоятельный сценарий. До запроса: требуемое свежее наблюдение off временно недоступно; воздействия нет.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000190016"
  bootId="00000000000000000000000000000073" status="ok">
  <readiness snapshotId="00000000000000000000000000001913"
    evaluatedAtMs="50000" validUntilMs="51000" stateRevision="1" more="false">
    <summary state="unknown"/>
    <resource id="ou1" bindingGeneration="2" state="unknown"
      reason="sourceUnavailable"/>
    <evidence resourceId="ou1" pointId="output1" required="true" observedAtMs="0"
      validUntilMs="0" validity="unavailable"/>
  </readiness>
</response>
```

B.166 Установить время: запрос

Самостоятельный сценарий. До запроса: административная установка часов разрешена; uptimeMs не меняется.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000190017" action="time.set"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="1">
  <clock utcMs="1791676800000" errorMs="100"/>
</request>
```

B.167 Установить время: ответ

Самостоятельный сценарий. До запроса: административная установка часов разрешена; uptimeMs не меняется.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000190017"
  bootId="00000000000000000000000000000073" status="ok">
  <clock utcMs="1791676800000" errorMs="100"/>
</response>
```

B.168 Подготовить групповую речь по открытому каналу: запрос

Самостоятельный сценарий: optional, явно выбран plain; шаблон speech1/1 имеет security=plain и не содержит ключей. Открытый поток не удостоверяет источник.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000192000" action="audio.group.prepare"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="1">
  <group id="speech1" revision="1"
    streamId="00000000000000000000000000000901" source="microphone"/>
</request>
```

B.169 Подготовить групповую речь по открытому каналу: ответ

Самостоятельный сценарий: optional, явно выбран plain; шаблон speech1/1 имеет security=plain и не содержит ключей. Открытый поток не удостоверяет источник.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000192000"
  bootId="00000000000000000000000000000073" status="ok">
  <prepared mediaId="00000000000000000000000000000900"
    expiresAtMs="110000" configRevision="7" capabilityRevision="4"/>
</response>
```

B.170 Состояние с владельцем: страница 1, запрос

Сценарий state-held-two-pages. Владение real выдано при uptimeMs=50000, действующего сеанса нет; единственный ресурс off. Снимок фиксируется в этом состоянии и передается двумя страницами.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000200001" action="device.state"
  bootId="00000000000000000000000000000073"/>
```

B.171 Состояние с владельцем: страница 1, ответ

Сценарий state-held-two-pages. Владение real выдано при uptimeMs=50000, действующего сеанса нет; единственный ресурс off. Снимок фиксируется в этом состоянии и передается двумя страницами.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000200001"
  bootId="00000000000000000000000000000073" status="ok">
  <state snapshotId="00000000000000000000000000200001" stateRevision="31"
    eventSeq="207" capabilityRevision="4" uptimeMs="50000" resourceCount="1"
    alarmCount="0" sessionCount="0" more="true" cursor="resources">
    <configuration configRevision="7" appliedRevision="7" onControlLoss="finish"/>
    <ownership state="held" principalId="00000000000000000000000000000082"
      controlEpoch="00000000000000000000000000000091" purpose="real"
      idleDeadlineMs="60000"/>
    <recovery state="ready"/>
  </state>
</response>
```

B.172 Состояние с владельцем: страница 2, запрос

Сценарий state-held-two-pages. Владение real выдано при uptimeMs=50000, действующего сеанса нет; единственный ресурс off. Снимок фиксируется в этом состоянии и передается двумя страницами.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000200002" action="device.state"
  bootId="00000000000000000000000000000073">
  <selection snapshotId="00000000000000000000000000200001" cursor="resources"/>
</request>
```

B.173 Состояние с владельцем: страница 2, ответ

Сценарий state-held-two-pages. Владение real выдано при uptimeMs=50000, действующего сеанса нет; единственный ресурс off. Снимок фиксируется в этом состоянии и передается двумя страницами.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000200002"
  bootId="00000000000000000000000000000073" status="ok">
  <state snapshotId="00000000000000000000000000200001" stateRevision="31"
    eventSeq="207" capabilityRevision="4" uptimeMs="50000" resourceCount="1"
    alarmCount="0" sessionCount="0" more="false">
    <resource id="ou1" output="off" evidence="reportedMode" observedAtMs="49900"/>
  </state>
</response>
```

B.174 Подготовка следующего источника в речевом сеансе

sourceSwitch=true объявлен; прежний источник paused. Владение и исполнение сохраняются, mediaId и ключ нового источника должны быть свежими.

```
<request v="1" profile="open-alert-draft/0.21"
  id="0000000000000000000000000000ac01" action="audio.prepare"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="11">
  <stream startup="outputReady"
key="1111111111111111111111111111111111111111111111111111111111111111"
    source="microphone" sampleRate="8000" holdOutput="true"
    sessionId="00000000000000000000000000000101" codec="PCMA"
    packetMs="20" ssrc="01234567" firstSeq="0" firstTimestamp="0">
    <resources>
      <resource id="ou1" bindingGeneration="2"/>
    </resources>
    <route sourceAddress="192.0.2.1" destinationAddress="192.0.2.2"
      sourcePort="5004" destinationPort="5004" ttl="1"/>
    <buffer initialMs="400" maxMs="1000"/>
  </stream>
</request>
```

B.175 Пауза источника без отбоя

Меняется только источник в одном подтвержденном речевом сеансе. Команда владельца с обычным cmdSeq.

```
<request v="1" profile="open-alert-draft/0.21"
  id="0000000000000000000000000000ac02" action="audio.pause"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="11">
  <stream mediaId="00000000000000000000000000000111"
    sessionId="00000000000000000000000000000101"/>
</request>
```

B.176 Ответ: Пауза источника без отбоя

Меняется только источник в одном подтвержденном речевом сеансе. Команда владельца с обычным cmdSeq.

```
<response v="1" profile="open-alert-draft/0.21"
  to="0000000000000000000000000000ac02"
  bootId="00000000000000000000000000000073" status="ok">
  <paused mediaId="00000000000000000000000000000111"
    sessionId="00000000000000000000000000000101" startSeq="2"/>
</response>
```

B.177 Запуск следующего источника без включения ОУ

Меняется только источник в одном подтвержденном речевом сеансе. Команда владельца с обычным cmdSeq.

```
<request v="1" profile="open-alert-draft/0.21"
  id="0000000000000000000000000000ac03" action="audio.start"
  bootId="00000000000000000000000000000073"
  controlEpoch="00000000000000000000000000000091" cmdSeq="11">
  <stream mediaId="00000000000000000000000000000111"
    sessionId="00000000000000000000000000000101"/>
</request>
```

B.178 Ответ: Запуск следующего источника без включения ОУ

Меняется только источник в одном подтвержденном речевом сеансе. Команда владельца с обычным cmdSeq.

```
<response v="1" profile="open-alert-draft/0.21"
  to="0000000000000000000000000000ac03"
  bootId="00000000000000000000000000000073" status="ok">
  <started mediaId="00000000000000000000000000000111"
    sessionId="00000000000000000000000000000101" startSeq="2"/>
</response>
```

B.179 Замена прочитанного снимка датчиков

УУ объявило limit snapshotReplace=1. Предыдущий снимок с той же выборкой полностью обработан; replaces разрешает освободить только его. Непрочитанные снимки остаются действительными.

```
<request v="1" profile="open-alert-draft/0.21"
  id="dddddddddddddddddddddddddddddddd" action="sensors.read"
  bootId="77777777777777777777777777777777">
  <selection replaces="66666666666666666666666666666666">
    <point id="ou1.tamper"/>
  </selection>
</request>
```

B.180 Включение инициативной доставки

Согласованный снимок уже обработан до eventSeq=41. Это запрос включения подписки; будущие события не требуют новых запросов.

```
<request v="1" profile="open-alert-draft/0.21"
  id="0000000000000000000000000000ee01" action="events.subscribe"
  bootId="0000000000000000000000000000ee00">
  <subscription subscriptionSeq="1" afterSeq="41"/>
</request>
```

B.181 Подписка принята

УУ полностью отправляет этот ответ перед первым инициативным уведомлением.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000ee01"
  bootId="00000000000000000000000000ee00" status="ok">
  <subscription subscriptionSeq="1" id="00000000000000000000000000ee10"
    afterSeq="41"/>
</response>
```

B.182 УУ сообщает сработку НСД

Предыдущий курсор 41. Сообщение отправлено УУ без events.read. Тот же alarmId связывает сработку и снятие.

```
<notification v="1" profile="open-alert-draft/0.21"
  bootId="00000000000000000000000000ee00"
  subscriptionId="00000000000000000000000000ee10">
  <events nextSeq="42" more="false" gap="false" oldestSeq="1">
    <event seq="42" stateRevision="42" capabilityRevision="4" type="alarm.raised">
      <alarm conditionId="tamper" alarmId="00000000000000000000000000ee20"
        pointId="ou1.tamper" state="active" severity="critical" validity="valid"
        freshness="fresh" observedAtMs="50100" changedAtMs="50100"
        continuity="continuous"/>
    </event>
  </events>
</notification>
```

B.183 АРМ подтверждает событие 42

Отправляется только после проверки и обработки страницы. Ответа на notificationAck нет; владение и слот команды не затрагиваются.

```
<notificationAck v="1" profile="open-alert-draft/0.21"
  bootId="00000000000000000000000000ee00"
  subscriptionId="00000000000000000000000000ee10" nextSeq="42"/>
```

B.184 УУ сообщает возврат НСД в норму

Предыдущий курсор 42. Сообщение отправлено УУ без events.read. Тот же alarmId связывает сработку и снятие.

```
<notification v="1" profile="open-alert-draft/0.21"
  bootId="00000000000000000000000000ee00"
  subscriptionId="00000000000000000000000000ee10">
  <events nextSeq="43" more="false" gap="false" oldestSeq="1">
    <event seq="43" stateRevision="43" capabilityRevision="4" type="alarm.cleared">
      <alarm conditionId="tamper" alarmId="00000000000000000000000000ee20"
        pointId="ou1.tamper" state="cleared" severity="critical" validity="valid"
        freshness="fresh" observedAtMs="51000" changedAtMs="51000"
        continuity="continuous"/>
    </event>
  </events>
</notification>
```

B.185 АРМ подтверждает событие 43

Отправляется только после проверки и обработки страницы. Ответа на notificationAck нет; владение и слот команды не затрагиваются.

```
<notificationAck v="1" profile="open-alert-draft/0.21"
  bootId="00000000000000000000000000ee00"
  subscriptionId="00000000000000000000000000ee10" nextSeq="43"/>
```

B.186 Инициативное сообщение о разрыве истории

Подтвержденный курсор 43, доступная история начинается с 50. Нужен новый согласованный снимок и новая подписка; notificationAck этот gap не снимает.

```
<notification v="1" profile="open-alert-draft/0.21"
  bootId="00000000000000000000000000ee00"
  subscriptionId="00000000000000000000000000ee10">
  <events nextSeq="43" more="false" gap="true" oldestSeq="50"/>
</notification>
```

B.187 Отмена подписки

Запрос отменяет только данную подписку на этом соединении; активное оповещение сохраняется.

```
<request v="1" profile="open-alert-draft/0.21"
  id="00000000000000000000000000ee02" action="events.unsubscribe"
  bootId="00000000000000000000000000ee00">
  <subscription id="00000000000000000000000000ee10"/>
</request>
```

B.188 Подписка отменена

После этого ответа уведомления прежней подписки запрещены. Неизвестный id имеет такой же безопасный ответ.

```
<response v="1" profile="open-alert-draft/0.21"
  to="00000000000000000000000000ee02"
  bootId="00000000000000000000000000ee00" status="ok">
  <subscription id="00000000000000000000000000ee10"/>
</response>
```

Приложение Г

(справочное)

Матрица требований

Матрица связывает сценарии, обязательные требования и критерии приемки.

Таблица 75 - Соответствие сценариев, требований и критериев приемки

ID	Сценарий	Требование	Критерий	Нормативные пункты
R01	Реальное оповещение, учения, звучащая и тихая проверки	Цель, права, допустимое физическое действие; учебный статус не означает тишину	Учения во время реального оповещения	пп. 8.1–8.59; пп. 5.40–5.45
R02	АРМ, резерв, местные кнопки, микрофон, внешние контакты	Источники, идентификация, удержание и освобождение управления	Местная кнопка во время сетевой записи; отпускание тангенты без АРМ	пп. 5.16–5.18; пп. 27.1–27.12; пп. 15.1–15.4
R03	Право прерывания	Текущий узел либо вышестоящий инициатор именно этого запуска через нижестоящий; высота и приоритет не дают права на чужой запуск	Вышестоящий отменяет свой запуск; попытка остановить независимо запущенное нижестоящим оповещение отклоняется	пп. 5.1–5.45; пп. 9.9–9.10

Продолжение таблицы 75

ID	Сценарий	Требование	Критерий	Нормативные пункты
R04	Прибор, группа, отдельные выходы	Адресные ресурсы с bindingGeneration; 1..4 ссылки, кодирование максимальных полей в пределах XML	session.start записи с четырьмя ID по 32 символа и максимальными uint64 укладывается в 1024 байта	пп. 6.1–6.13; пп. 19.7–19.19
R05	Сирена, запись, речь, последовательность	Сирена: редакция аппаратного профиля, фазы и допуски; durationMs от running с паузами, maxDurationMs от acceptedAtMs. Составные программы отдельным расширением	Неполный последний цикл, задержка подтверждения running, совпадение пределов, неподдержанная длительность; сирена, пауза и запись в профиле программы	пп. 10.6–10.15; пп. 26.1–26.16; пп. 15.10–15.22
R06	Запись по номеру и обновление содержимого	Локальная запись доступна до session.start; неизменные contentId/revision/hash, атомарное размещение, проверка и готовность. Передача файла во время оповещения является отдельным потоковым режимом	Обрыв загрузки, смена содержимого слота и повтор session.start не запускают неполный/новый объект; M01	пп. 15.5–15.9; пп. 22.4–22.7; пп. 16.23–16.26

Продолжение таблицы 75

ID	Сценарий	Требование	Критерий	Нормативные пункты
R07	Исправление и отмена предупреждения	Связь предупреждения АРМ с сеансами, область отмены, правила замены	Исправленное сообщение приходит во время старого; отмена одного не останавливает другое	пп. 21.4–21.15; пп. 9.9–9.10
R08	Немедленный групповой запуск	Каждый получатель независимо принимает session.start в acceptBeforeMs; единый момент физического запуска по UTC в 0.21 не предоставляется	Отказ одного получателя не превращается в общий успех; АРМ показывает точный состав принявших и исполнивших	пп. 8.1–8.59; пп. 21.4–21.15
R09	Потеря управления при сирене или записи	Настройка stop/finish на приборе, снимок ревизии на запуск; разрыв транспорта отдельно от lease; таблица переходов и непродлеваемые сроки	Наблюдатель остается; resume до, на границе и после lease; восстановление при draining не разрешает новый session.start и не отменяет stopping	пп. 5.19–5.35; пп. 8.4–8.9
R10	Потеря аудио, управления или обоих каналов	Конечные сроки потери управления и аудио; небольшой ограниченный jitter-буфер микрофона, учет возраста пакета и политики восстановления	Потери не скрываются буфером; поздний звук не воспроизводится, малый буфер проверяется на заданном канале; M03/M05/M07/M10	пп. 15.35–15.39; пп. 15.47–15.68

Продолжение таблицы 75

ID	Сценарий	Требование	Критерий	Нормативные пункты
R11	Повтор команды, потеря ответа	Две границы приема, сохраненное решение, сравнение нормализованного содержимого до предметной валидации	Повторы unsupported/ invalidArgument возвращают прежний отказ; изменение ошибочного поля дает conflict	пп. 6.5–6.13; пп. 9.11–9.13
R12	Питание, перезапуск, часы	bootId, постоянное намерение до физического запуска, атомарная фиксация, обнаружение оборванной записи и восстановление unknown	Отключение питания на каждой границе записи и воздействия; повреждение Flash не создает повторный запуск или ложный off	пп. 8.10–8.13; пп. 8.14–8.22
R13	Отказ усилителя, линии, накопителя	Доступные измерения, их свежесть, причина завершения	Команда принята, усилитель не включился; нельзя утверждать, что население услышало сообщение	пп. 13.29–13.42; пп. 13.1–13.42
R14	Обслуживание во время тревоги	Допустимые настройки и загрузки, прерывание сервисных операций	Срочный запуск во время загрузки файла или подготовки прошивки	пп. 5.36–5.39; пп. 16.3–16.19

Продолжение таблицы 75

ID	Сценарий	Требование	Критерий	Нормативные пункты
R15	Два АРМ, разделение сети	Один активный экземпляр, acquireSeq/attachSeq, lease, освобождение и независимое ограждение резервируемого АРМ	Потерян ответ acquire; две копии пытаются resume; возврат изолированной копии не дает ей доступа	пп. 5.16–5.18; пп. 6.5–6.13
R16	Пропуски событий, устаревшее состояние	events.read: afterSeq/nextSeq без snapshotId; снимки отдельно; обнаружение пропуска и история результата	Повтор afterSeq при новых событиях и переполнении кольца; gap требует снимка, idle не доказывает успешного окончания	пп. 19.72–19.73; пп. 13.4–13.12
R17	Замена прибора, ключей, отзыв доступа	Индивидуальное доверие, поколения и отзыв ключей; российская защита хранения и переноса секретов	Новый прибор по прежнему IP, плановая ротация с resume, отзыв старого ключа, утрата всех ключей и восстановление без возврата отозванного поколения.	пп. 14.40–14.49; пп. 14.77–14.78; пп. 14.79–14.81
R18	Подделка, повторы, перегрузка	Российский TLS-профиль: Кузнечик-MGM, Стрибог-256, индивидуальный PSK, GC256A; ограниченные записи и обработка	Чужой ключ, несовместимый криптонабор, поврежденная метка, повтор записи и исчерпание энтропии не приводят к исполнению или отключению защиты	пп. 14.50–14.54; пп. 9.31–9.36

Продолжение таблицы 75

ID	Сценарий	Требование	Критерий	Нормативные пункты
R19	Большая группа, слабый канал, multicast	Без общего протокольного предела парка, с конечными квотами развертывания; RTP/UDP multicast для массовой речи, расчет unicast-полосы и pps, медленные ветви, явная модель доверия, медиаключи при их использовании и дозирование ответов	1/10/100/1000/4000/10000 УУ; массовые ответы, допуск источника, смена группового ключа при его использовании, задержка и нагрузка реальных ветвей; M08/M09/M10	пп. 15.29–15.34; пп. 25.1–25.3
R20	Обновление и восстановление	Совместимый подлинный образ, права, допустимость отката, состояние загрузки	Питание пропало при записи; подлинный образ для другой платы	пп. 16.3–16.19; пп. 16.23–16.26
R21	Неизвестная функция или версия	Ядро, capabilities, расширения, точные отказы	Прибор без синхронного запуска НЕ ДОЛЖЕН вместо него запускаться немедленно	п. 20.10; пп. 19.2–19.6

Продолжение таблицы 75

ID	Сценарий	Требование	Критерий	Нормативные пункты
R22	Разбор инцидента	Участник, команда, сеанс, результат, качество времени; resultRetentionMs в одной загрузке отдельно от постоянного журнала восстановления и архива АРМ. Емкость покрывает скользящее окно завершений и резерв принятого исполнения; занятая история не задерживает следующий этап в пределах заявленной нагрузки	Release не сокращает срок результата; reboot дает staleBoot, журнал сохраняет незакрытое воздействие; поздний запуск запрещен и после утраты подробного решения	пп. 8.23–8.27; пп. 19.53–19.63
R23	Сторонний сервис и новый тип подключения	Расширяемые адаптеры АРМ, общий контракт, идентичность, права, сопоставление событий и запусков	Повтор webhook, перезапуск адаптера, отмена от другого сервиса, устаревшее сообщение после восстановления	пп. 17.1–17.3; пп. 20.1–20.10
R24	Подтверждения узла, устройств и оператора	Разные виды подтверждений, источник, объект и время; ручное подтверждение не заменяет телеметрию	Узел принял запуск, часть устройств отказала, оператор подтвердил получение	пп. 8.45–8.59; пп. 8.1–8.59

Продолжение таблицы 75

ID	Сценарий	Требование	Критерий	Нормативные пункты
R25	Подготовленный тракт и фактический прием речи	Готовность тракта, первый пакет, выдача отсчетов и физическое наблюдение различаются; три способа речи и приемка M01-M10; wire-медиапрофиль rtp-alert/5 задает END, validSamples, EOF и дрейф часов	Готовый тракт без аудио не объявляет воспроизведение; потери середины/хвоста учитываются и дают mediaPacketLoss, но не обрывают поток. Полный проход известного EOF и off допускают succeeded/completed с предупреждением. Изменение очереди не запускает таймер отказа часов; M02/M03/M04.	пп. 15.10–15.22; пп. 15.83–15.93
R26	Состав оборудования и датчиков	Четыре facet точки, строгие presence/support/readMethod; enum и threshold с точными полями и связями	Полный каталог без пропуска facet, повторов enum и порогов на несуществующие точки	пп. 11.3–11.6; пп. 19.43–19.52
R27	Достоверность показаний	Значение, источник, возраст и качество; полный выбранный набор sensors.read, недоступность отдельной записью	Лишняя или потерянная точка и дубликат между страницами обнаруживаются; устаревшее значение не становится свежей нормой	пп. 11.1–11.6; пп. 19.64–19.71

Продолжение таблицы 75

ID	Сценарий	Требование	Критерий	Нормативные пункты
R28	Пассивный свежий опрос	pollId, предел времени, ожидаемые точки, отдельные результаты и complete как признак покрытия	Ответила половина источников; наблюдатель не удерживает связь владельца	пп. 12.7–12.12; пп. 19.64–19.71
R29	Опросы без запуска и с запуском ОУ	Два явных сценария; activationPoll через общий сеанс, фазы, права, полнота и подтвержденное выключение; ресурс с bindingGeneration, checks и observations со счетчиками. Для mode=test обязателен полный каталог tests.list по разделу 12	ОУ включилось, измерения собраны, выключение неизвестно: общий успех запрещен; результат сопоставлен с session.start/accepted и фиксированным планом; пропуск записи запрещает успех	пп. 12.7–12.12; пп. 12.21–12.24

Продолжение таблицы 75

ID	Сценарий	Требование	Критерий	Нормативные пункты
R30	Аварийные эпизоды	conditionId/alarmId, active/unknown в снимке, raised/cleared/qualityChanged в журнале, alarmCount и граница eventSeq; bool/int64/uint64/enum проверяются по типу и согласованному каталогу	Подключение при действующей аварии и gap журнала восстанавливают полный активный набор без выдуманного нового фронта	пп. 12.13–12.16; пп. 19.39–19.42
R31	Инженерная диагностика	Слои контроллера, сети, ОУ, хранения и медиа; область счетчиков	Перезапуск, насыщение счетчика, неизвестная crash capsule	пп. 13.27–13.28
R32	Автономный мониторинг и массовые опросы	Периоды источников, фильтры, квоты и бюджет совместно с речью	Шторм изменений и несколько наблюдателей не блокируют управление	пп. 13.4–13.12; пп. 25.1–25.3
R33	Согласованная телеметрия	Нефильтруемый журнал; согласованный снимок с контрольными счетчиками, неизменными страницами и конечным восстановлением	Пропуск аварии, смена ревизии, повтор cursor или смешение страниц обнаруживаются; неполный снимок не очищает аварии	пп. 19.28–19.38; пп. 13.4–13.12

Продолжение таблицы 75

ID	Сценарий	Требование	Критерий	Нормативные пункты
R34	Готовность отдельных функций	Раздельно командная доступность и готовность сирены/записи/речи с причинами	УУ отвечает, усилитель неисправен, часть фидеров недоступна	пп. 22.4–22.7
R35	Функциональные части УУ	Сетевой контракт АРМ/конфигуратора и требования к исполнительному тракту; внутренний интерфейс и число плат выбирает изготовитель	Новый производитель не требует переопределения базовой команды АРМ	пп. 10.3–10.5; пп. 4.1–4.12
R36	Независимые отказы сети и исполнительного тракта	Раздельные состояния контроллера, взаимодействия с ОУ и самого ОУ в пределах объявленных наблюдений, без автоматического повторного session.start	УУ на связи, ОУ не отвечает; затем взаимодействие с ОУ восстанавливается	пп. 10.3–10.5; пп. 15.138–15.140

Продолжение таблицы 75

ID	Сценарий	Требование	Критерий	Нормативные пункты
R37	Привязка и смена профиля ОУ	ouId, драйвер, поколение привязки, каталог и реальные функции; известная АРМ и УУ редакция аппаратного профиля сигнала до session.start	Поздний ответ прежнего ОУ после перенастройки не подтверждает новое исполнение	пп. 10.6–10.15; пп. 24.1–24.18
R38	Собственные запросы УУ к пульту	Двунаправленная корреляция, объявленные виды запросов, права и пределы ожидания	Одновременно идут запрос АРМ и запрос УУ, ответы не смешиваются	пп. 7.1–7.35; пп. 20.7–20.9
R39	Семейство совместимых профилей	profileId/версия, зависимости, согласование обязательных функций и лимитов, единая семантика результатов	Нет совместимой версии: управление не начинается и защита не ослабляется	п. 20.10
R40	Логическая цель и локальный адрес	targetId отдельно от deviceId и locator, область сети, реестр привязок и выявление конфликтов	Два одинаковых частных IP в разных муниципалитетах не смешивают команды и результаты	пп. 21.1–21.15

Продолжение таблицы 75

ID	Сценарий	Требование	Критерий	Нормативные пункты
R41	Контролируемый путь через АРМ	Ответственный узел, ревизия маршрута, hopLimit, циклы и исходный срок приема	Ретрансляция по циклу или после срока не создает поздний запуск	пп. 21.4–21.15
R42	Предпусковая оценка конкретного плана	readiness.get, возраст и источник доказательств, ready/ degraded/blocked/unknown, быстрый повтор условий на session.start	После ready изменились внутренний интерфейс, контент или привязка: старое заключение не разрешает запуск	пп. 22.4–22.7
R43	Проверяемый манифест контента	Локальный неизменный объект contentId/revision/hash до session.start; точные байты, streebog256, готовность и естественный конец; сетевое размещение использует полный контракт bulk/1.	Обрыв размещения, неверный хеш, замена слота, неподтвержденный EOF; номер записи не равен verified; M01/M04	пп. 22.4–22.7; пп. 15.5–15.9
R44	Каскадное наблюдение	Отдельные права чтения, сквозная цель, источник и возраст исходного показания	Посредник повторяет старый кеш и не делает его свежим	пп. 23.1–23.4; пп. 21.4–21.15

Продолжение таблицы 75

ID	Сценарий	Требование	Критерий	Нормативные пункты
R45	Сводка без утраты детализации	Зафиксированный набор целей, взаимно исключающие статусы, ревизии, страницы первичных результатов	Повтор, поздний ответ и два маршрута не увеличивают число успешно оповещенных целей	пп. 23.1–23.4; пп. 19.53–19.63
R46	Обнаружение и доверенная регистрация	Discovery отдельно от допуска, квоты, проверка идентичности, наладка привязки ОУ	Найденное устройство или чужое объявление не получает полномочий управления	пп. 24.1–24.18; пп. 14.62–14.71
R47	Переносимый пакет конфигурации	Схема, ревизия, ссылки, манифесты, область полномочий, проверяемый импорт без секретов	Импорт с отсутствующей записью, дубликатом цели или циклом отклоняется до применения	пп. 24.14–24.18
R48	Согласованный состав групп	Уполномоченный автор, groupRevision и digest, зафиксированное множество получателей	Одноименная группа на другом АРМ имеет другой состав: молчаливая подмена запрещена	пп. 24.14–24.18; пп. 21.1–21.15

Продолжение таблицы 75

ID	Сценарий	Требование	Критерий	Нормативные пункты
R49	Фактическое применение и замена оборудования	configRevision/appliedRevision/ onControlLoss в обязательном device.state, контроль применения перед session.start	Первый session.start без config.get: ревизии прочитаны из ядра; неприменная конфигурация дает unsafeState	пп. 24.1–24.18; пп. 6.1–6.13
R50	Измеряемый паспорт масштаба	Паспорт нагрузки и сети, отдельный heartbeat, локальный прием отмены, начало session.stop и подтверждение off; реестр limit, snapshotMaxPages/ snapshotReadMaxMs, скорость журнала и квоты снимков	TCP head-of-line blocking и полный отказ сети учитываются отдельно; finish не выдается за остановку; полный снимок укладывается в TTL с запасом, переход от теста к тревоге измерен также при отказе владельца	пп. 25.1–25.3
R51	Этапы и проходы программы	planRevision, stepId, проход, переходы, отмена таймеров, межэтапное владение; автономная программа отдельным профилем	Поздний session.stop сирены после подготовки речи; отмена в паузе перед повтором записи; U01-U02	пп. 26.6–26.16

Продолжение таблицы 75

ID	Сценарий	Требование	Критерий	Нормативные пункты
R52	Тангента и право передавать речь	Разрешение источника отдельно от владельца; подготовка без захвата допустима заранее. Одно действующее намерение нажатия ожидает prepared, готовность медиаканала и согласованный outputReady; отмена исключает поздний START. Новое поколение при новом нажатии; wire-схема audio.prepare/session.start/ audio.finish и rtp-alert/5 фиксирует жизненный цикл	Поздняя выдача, два источника, потеря говорящего, аудио после отпускания; U03, M05/M06/M09, microphone-pending-preparation	пп. 15.1–15.4; пп. 15.125–15.137
R53	Резервные пути к одному ОУ	Единый физический ресурс и логическая цель, неизвестный результат первого пути, исключение двух исполнителей	Ethernet-запуск выполнен без ответа, попытка повторить через внешний радиошлюз; U05	пп. 21.4–21.15; пп. 5.16–5.18

Продолжение таблицы 75

ID	Сценарий	Требование	Критерий	Нормативные пункты
R54	Неопределенность нижнего протокола	Политика повторов каждого воздействия на ОУ, достаточность состояния, поздние ответы и восстановление	session.start исполнен, ACK потерян; общий режим записи не подтверждает конкретный контент; U04	пп. 10.3–10.5; пп. 8.14–8.22
R55	Срабатывание, снимок и повтор внешнего события	Источник/эпизод, ID и ревизия правила, фронт/уровень, дребезг, повторное взведение, возврат питания	Replay и снимок не являются новым запуском; активный вход после питания требует правила; U08-U09	пп. 27.10–27.12; пп. 12.13–12.16
R56	Независимые реализации АРМ	Логический узел отдельно от процесса и оператора; ядро, расширения, схемы, открытая приемка	Разные АРМ одинаково трактуют состояния и отклоняют неизвестную обязательную функцию; U11	пп. 18.1–18.20; п. 20.10
R57	Отказ привязки медиасоединения	Непривязанный канал не изменяет действующий поток; один экземпляр соединения на mediaId.	Чужой principalId с верным mediaId, повтор того же владельца и поздний close старого канала.	пп. 15.138–15.140

Продолжение таблицы 75

ID	Сценарий	Требование	Критерий	Нормативные пункты
R58	Учение перед реальной тревогой	Техническая проверка ограничена 30000 мс; длительное учение до 1080000 требует независимого готового тракта тревоги.	Проверяются обе границы, отсутствие независимого тракта, автоматическое освобождение и сохранение места real.	пп. 5.40–5.45
R59	Полный пакет локальных правил	open-alert-config/5 содержит controllers и localRules, ссылки, полярность и правила edge/level.	Отсутствующая политика, чужой deviceId, неизвестная программа или повтор входа отклоняют пакет.	пп. 24.14–24.18; пп. 27.1–27.12
R60	Устойчивый вход после reboot	level оценивает новое физическое наблюдение после ready; edge ожидает повторного взведения.	Активный уровень при загрузке, один запуск за эпизод, снятие ожидающего уровня, неизвестный вход.	пп. 27.10–27.12
R61	Достоверность успешного результата	succeeded требует passed всех обязательных проверок и подтвержденного off всех выбранных ресурсов.	Полный по счетчикам снимок с failed-проверкой, onObserved или чужой ссылкой отклоняется.	пп. 19.125–19.132; пп. 19.53–19.63

Продолжение таблицы 75

ID	Сценарий	Требование	Критерий	Нормативные пункты
R62	Оконная передача файлов	bulk/1: блок 4096..65536 байт, окно 4..64, явное согласование; DATA/ACK/ERROR и границы кадра.	Четыре блока без ACK принимаются; выход за окно, пропуск, перекрытие, неверное направление/размер отклоняются.	пп. 16.30–16.40
R63	Устойчивый префикс загрузки	ACK подтверждает сохраненные данные и журнал; незавершенный хвост не повреждает ранее подтвержденное.	Потеря ACK и питания; новый hello возвращает устойчивый префикс; повреждение дает storageFault.	пп. 16.41–16.45
R64	Изоляция файлового канала	Привязка к principalId, эпохе и экземплярам двух соединений; независимое обслуживание управления.	Чужой/повторный hello и поздний callback не меняют передачу; real отзывает сервис за 500 мс.	пп. 16.27–16.29
R65	Сроки окна и памяти	Бюджет по числу окон, ограничение буферов и непрерываемого шага; ACK не продлевает lease.	Неполный DATA, медленный ACK, частичные подтверждения и бессодержательные повторы не снимают срок старейшего блока.	пп. 16.23–16.26

Продолжение таблицы 75

ID	Сценарий	Требование	Критерий	Нормативные пункты
R66	Завершение и отмена файловой операции	Полный ACK не равен staged/verified/committed. Проверка до 60000 мс согласована с idle, независимым lease и общими сроками; cancel и commit имеют определенные исходы.	Проверка дольше 10000 мс завершается при живом lease; срок проверки, потеря lease и real-прерывание блокируют поздний результат; новая попытка не продлевает общие сроки.	пп. 16.20–16.22; пп. 16.46–16.57
R67	Подтверждение исполнения	Принятие, начало работы, полнота и завершение передаются раздельно.	accepted без воздействия; подтвержденная работа до session.stop; отмена до session.start; частичный запуск четырех ресурсов.	пп. 8.45–8.59
R68	Штатная остановка	succeeded/operatorStop только для подтвержденной работы siren/recording/stream и off в срок без отказа.	Остановленная речь имеет interrupted; остановленная программа не получает успех всей программы.	пп. 8.45–8.59
R69	Неисправность после запуска	Историческое confirmed сохраняется, fault и аварийный итог не стираются.	Потеря фазы после начала, session.stop после отказа, последующее снятие, неизвестный off и независимая авария чужой линии.	пп. 13.29–13.42

Продолжение таблицы 75

ID	Сценарий	Требование	Критерий	Нормативные пункты
R70	Полномочия и занятость	forbidden предшествует проверке занятости; причины отказов машиночитаемы.	Нет роли, запрет области, чужой session.stop/resume; busy своего нового session.start; точный повтор; неизменность таймеров и highWater при forbidden.	пп. 9.14–9.30
R71	Предельные размеры	Все команды и целые записи в максимально длинной допустимой сериализации укладываются в 1024 байта.	Все действия, режимы session.start и readiness, четыре ресурса, восемь точек, uint64, UTF-8 и расширение XML-сущностей; 1025 байт отклоняются.	пп. 7.24–7.31
R72	Сирена до команды остановки	stopMode=command не использует durationMs; независимый maxDurationMs и stop при потере управления. Для речевого ОУ с signalKind=builtIn подтвержденный конец собственного цикла завершает сирену успешно.	Нормальный session.stop, предел вместо session.stop, потеря lease, запрет command в автономной программе.	пп. 19.20–19.27

Продолжение таблицы 75

ID	Сценарий	Требование	Критерий	Нормативные пункты
R73	Окно номера	cmdSeqWindow=4096; отказ вне окна не расходует состояние.	Границы highWater+4096/+4097, точный повтор, резерв session.stop/ control.release, uint64 и новая эпоха.	пп. 6.5–6.13
R74	Привязка и резерв	Живая привязка не вытесняется resume; внешнее ограждение остается обязательным.	Живой, отсоединенный и истекший канал; отказ сохраняет attachSeq/ lease; поздний обработчик; несколько УУ.	пп. 5.16–5.18
R75	Сигнал	signalProfile ссылается на profiles; signalSpec на полный реестр сигналов config/5.	Отсутствие, другая ревизия, неверный тип/частота, несогласованный паспорт отклоняются.	пп. 10.16–10.22
R76	Медиапривязка	Неверная UDP-привязка отбрасывается и не вытесняет поток.	Успех, чужой principalId, занятая привязка, потерянный ответ, неверный профиль.	пп. 15.138–15.140

Продолжение таблицы 75

ID	Сценарий	Требование	Критерий	Нормативные пункты
R77	Доверие	Первичное доверие закрепляется независимо; выдача секрета, staged, пробный TLS, активация и рабочее подключение различаются; импорт open-alert-trust/2 атомарен.	Чистое устройство, неверная идентичность, секрет только на одной стороне, staged без управления, потеря подтверждения активации, откат поколения и обрыв питания.	пп. 14.40–14.49; пп. 14.62–14.71; пп. 14.74–14.76; пп. 14.82–14.88
R78	Подпись и издатель	Кодирование подписи однозначно; отзыв перепроверяется до commit.	Проверочные значения двумя реализациями; перестановки r/s, измененный манифест, отозванный издатель.	пп. 14.12–14.14; пп. 14.40–14.49
R79	Единицы и сроки	UTF-8 ограничен байтами; session.stop и backoff имеют конечные границы.	128/129 байт, непарный суррогат, общий/технический session.stop, шторм переподключений и независимый слот отмены.	пп. 7.11–7.23
R80	Согласованность комплекта	Общие XML, закрытые причины и все артефакты согласованы с манифестом.	exercise/stop, programAborted только notStarted, ссылки на текущие digest/signature-векторы.	пп. 18.1–18.20

Продолжение таблицы 75

ID	Сценарий	Требование	Критерий	Нормативные пункты
R81	Открытый и защищенный обмен	Режим plain задается явно; secureOnly запрещает открытое управление.	Чистое УУ при optional; неверный TLS не становится plain; открытый канал не наследует TLS-владение.	пп. 14.55–14.61
R82	Парное доверие без секретов в пакете	Пакет доверия не содержит секретных ключей.	Схема отвергает pskHex; изменение deviceId/principalId/rootId меняет PSK; сверка KDF с векторами.	пп. 14.40–14.49; пп. 14.72–14.73
R83	Свежесть речевых пакетов	rtp-alert/5 ограничивает возраст каждого выдаваемого пакета.	Задержки TCP, истечение часов, подмена метки, тишина и session.stop; отдельно для tls и plain.	пп. 15.47–15.68
R84	Допуск handshake	Неаутентифицированные TCP-сокеты не занимают контексты PSK-DHE.	Два молчащих подключения, повтор ClientHello и неверный binder не занимают дорогой резерв владельца.	пп. 14.50–14.54

Продолжение таблицы 75

ID	Сценарий	Требование	Критерий	Нормативные пункты
R85	Слоты и кеш решений	Чтения и heartbeat не расходуют cmdSeq; резервные номера обходят обычное окно.	Повторы ping, занятый слот, скачок выше окна и гонки session.start/ session.cancel.	пп. 9.31–9.36
R86	Повтор освобождения	Новый канал того же участника возвращает сохраненное решение control.release без владения.	Принятый и непринятый release, новый канал, смена principalId, режима и bootId.	пп. 9.31–9.36
R87	Канал от 64 кбит/с	Сжатая речь, управление и накладные расходы укладываются в рассчитанную полосу.	GSM/80, задержки и потери, пауза фоновой передачи, верхний срок session.stop.	пп. 7.32–7.35; пп. 15.29–15.34

Окончание таблицы 75

ID	Сценарий	Требование	Критерий	Нормативные пункты
R88	Групповая речь	Подпись источника проверяется отдельно от общего ключа; ACK на пакет запрещены.	Подделка получателем, одна и две потери частей, повтор, END, отзыв и сбор результатов.	пп. 15.29–15.34; пп. 15.10–15.22
R89	Схема и примеры	XSD, поля запросов, закрытые словари и все команды имеют проверяемое покрытие.	Неверный status, неизвестное поле, чужой to, обязательные начальные состояния и сквозные трассы.	пп. 19.84–19.124; пп. 7.1–7.35
R90	Инициативные уведомления УУ	При eventsPush=1 УУ передает notification без запроса очередного события; согласование, подтверждение, повтор, gap, права и приоритет управления обязательны.	Сработка и возврат без events.read, потеря подтверждения, переподключение, переполнение журнала, старые клиенты и START/STOP под нагрузкой.	пп. 13.13–13.26; пп. 18.19–18.20

Приложение Д

(справочное)

Пояснения к требованиям

Справочные пояснения связывают правила протокола с устраняемой неоднозначностью. Они не вводят дополнительных полномочий или гарантий; нормативные условия, состояния и исключения определены в указанных разделах.

Д.1 Кто вправе воздействовать на ОУ?

Аутентификация, права участника, владение и происхождение запуска проверяются раздельно. Положение в иерархии и знание sessionId не заменяют полномочий; отмена порученного запуска не дает права прерывать самостоятельное местное оповещение.

УУ проверяет непосредственный аутентифицированный АРМ. Проверка межузловой цепочки поручений выполняется на уровне АРМ.

5 Владение и право прерывания

Д.2 К какому исполнению относится команда?

Контекст bootId/controlEpoch и номера команд отделяют исполнения и повторы. session.cancel адресует исходный session.start через startSeq, в том числе когда ответ с sessionId потерян. Старая отмена не переносится на новый запуск.

Удаление подробного результата не разрешает повторное воздействие старым номером. resultExpired означает отсутствие доступной подробности, а не отсутствие исполнения.

9 Повторы, отмена и неизвестный исход

Д.3 Что делать при неизвестном исходе?

Потерянный ответ допускает как неисполнение, так и уже начавшееся воздействие. Требуются сверка состояния и предусмотренное восстановление. Новый независимый session.start запрещен при занятости или неподтвержденном безопасном состоянии ресурса.

Отправка session.stop не доказывает off. До подтверждения выключения неопределенность физического состояния сохраняется.

8 Сеанс, физический результат и потеря связи

Д.4 Что подтверждает ответ?

Прием команды, начало исполнения, выдача сигнала и физическое наблюдение имеют разные доказательства. Результат сообщается вместе с доступным уровнем наблюдения; успешная сетевая доставка не подменяет физический результат.

Связь с УУ не доказывает исправность ОУ. Сетевые квитанции и цифровая полнота аудио не подтверждают слышимость и понимание сообщения населением.

15 Речь, записи и время; подраздел «Приемочные испытания речи»

Д.5 Почему подключение не дает новое владение?

principalId, instanceId, controlEpoch, attachSeq и bootId решают разные задачи. Resume восстанавливает тот же контекст; дублер ДОЛЖЕН предварительно исключить прежнего активного писателя.

Общий ключ и успешный handshake не доказывают, что другая копия АРМ больше не управляет устройством.

5 Владение и право прерывания; подраздел «Переподключение и дублер»

Д.6 Что гарантирует защита от повторов?

Дедупликация сетевой команды предотвращает повторное решение управляющего ядра в заданном контексте. Для физического интерфейса отдельно учитываются идентификация команды, потерянный ответ и возможность наблюдать результат.

Старое ОУ без корреляции команд не получает гарантию однократного физического воздействия от одной лишь дедупликации Ethernet. Общий режим записи не доказывает выбранное содержимое.

10 Профили ОУ: ВАУ и Сирена

Д.7 Почему значение сопровождается качеством?

Значение, качество, свежесть, источник и наличие точки различаются. Полнота снимка и непрерывность журнала проверяются отдельно; неполный снимок не очищает аварии.

Полный набор диагностических данных может описывать неисправное оборудование. Полнота передачи и исправность являются независимыми признаками.

13 Диагностика, события и снимки

Приложение Е

(справочное)

Сведения об издании и правах использования

Наименование

Открытый протокол систем оповещения. Техническая спецификация

Организация-разработчик

ООО "СОТКОМ-ЮГ"

Статус

Проект стандарта организации. Для публичного обсуждения

Обозначение

Обозначение СТО не присвоено

Утверждение

Не утвержден. Дата введения в действие не установлена

Настоящий документ подготовлен как проект СТО. Его размещение для обсуждения не является утверждением стандарта организации или национального стандарта. Файловая и онлайн-редакции используют общие технические требования, схемы и примеры; различаются их оформление и навигация.

Уточнение редакции 0.21 от 22 октября 2026 г. разграничивает счетчики пакетов и отсчетов, регулирование очереди, свежесть звука и аппаратные отказы. Установлены единые правила начала короткого потока с потерями, проверки позднего EOF, ожидания подготовки микрофона и чтения результата остановки после истечения владения. Задержка от действия оператора до подтвержденного off измеряется с учетом восстановления связи. Добавлено явно согласуемое sourceSwitch для перехода между источниками без отбоя ОУ. XML расширен командами audio.pause/audio.start и параметрами holdOutput/sessionId; без согласования действует прежний порядок. UDP и режимы 8 кГц сохранены.

Федеральный закон от 29.06.2015 № 162-ФЗ, статья 21

Правовая основа разработки, утверждения, изменения и применения СТО. Порядок устанавливает организация. Экспертиза техническим комитетом и регистрация в фонде стандартов не объявляются выполненными.

ГОСТ Р 1.4-2004. Стандартизация в Российской Федерации. Стандарты организаций. Общие положения

Основа организации разработки и применения стандарта организации.

ГОСТ Р 1.5-2012. Стандартизация в Российской Федерации. Стандарты национальные. Правила построения, изложения, оформления и обозначения

Для файловой версии проекта выбраны правила структуры, нумерации, изложения требований, таблиц, приложений и ссылок. Область самого ГОСТ - национальные и предварительные национальные стандарты; применение выбранных правил к СТО закрепляется при утверждении в организации.

Редакция документа, версия wire-профиля и версия реализации учитываются раздельно. Номер документа и дата указываются в файле и онлайн-версии; комплект редакции связывает текст, схемы, словари, примеры и проверки контрольными суммами. Изменение технического требования отражается в журнале редакций. До утверждения сохраняется пометка проекта; обозначение СТО, приказ и дату введения присваивает организация.

Открытая безвозмездная лицензия на материалы протокола, редакция 3

1. Лицензиар — ООО «СОТКОМ-ЮГ». По статье 1286.1 Гражданского кодекса Российской Федерации лицензиар предоставляет любому лицу простую (неисключительную) безвозмездную лицензию на опубликованные им текст спецификации «Открытый протокол систем оповещения», схемы, словари, примеры, тестовые векторы и исходные тексты проверок, прямо сопровождаемые этой лицензией. Сторонние произведения, на которые даны ссылки, в предмет лицензии не входят.

2. Разрешены воспроизведение полностью и частично на любых носителях, распространение экземпляров, доведение до всеобщего сведения, публичный показ, перевод, переработка, включение в состав других произведений и программ, использование исходных и переработанных материалов в коммерческой и некоммерческой деятельности. Разрешено создавать и распространять производные материалы и предоставлять права на них, включая включенные исходные материалы, в пределах перечисленных прав. Отдельное разрешение и выплата вознаграждения не требуются.

3. Использование разрешено без территориальных ограничений. Срок — весь срок действия исключительного права на соответствующий материал. Совершение любого из перечисленных в пункте 2 действий после ознакомления с условиями является акцептом лицензии; письменная форма договора при этом считается соблюденной. Простое чтение документа не возлагает дополнительных обязанностей.

4. Обязательное размещение фирменного наименования, логотипа, ссылки на лицензиара или рекламной подписи в сетевых сообщениях, программах, изделиях и

их интерфейсах не устанавливается. При передаче копии лицензируемых материалов получателю предоставляются настоящие условия; допустимо включить их в сопроводительный файл. Измененная спецификация обозначается как измененная, чтобы ее нельзя было принять за исходную редакцию. Личные неимущественные права авторов, охраняемые законом, сохраняются; эта лицензия не является отказом от них.

5. Независимая реализация описанного метода обмена не требует заключения договора с разработчиком спецификации, членства в организации или раскрытия исходного кода реализации. По пункту 5 статьи 1259 ГК РФ идеи, методы, процессы, системы, способы и технические решения сами по себе не охраняются авторским правом. Лицензия распространяется на охраняемую форму материалов и не устанавливает исключительного права на сам способ обмена.

6. Лицензия не предоставляет права на товарные знаки третьих лиц и не означает сертификацию, поддержку или одобрение конкретного изделия. Лицензиар предоставляет права только в отношении принадлежащих ему материалов. Самостоятельная реализация может распространяться на выбранных ее правообладателем условиях.

7. Материалы предоставляются в опубликованном состоянии. Совместимость и пригодность реализации проверяются испытаниями; бесплатное предоставление материалов само по себе не является обязательством сопровождения или гарантией работоспособности оборудования. Ограничения ответственности применяются только в пределах, допускаемых законодательством Российской Федерации. К лицензии применяется право Российской Федерации.

8. При соблюдении настоящих условий предоставленные права сохраняются на весь установленный срок и не могут быть отозваны по усмотрению лицензиара. Снятие материалов с публикации, замена редакции, изменение условий для последующих публикаций или прекращение сопровождения не прекращают ранее предоставленные права. Прекращение либо ограничение договора допускается только по основаниям и в порядке, которые прямо установлены применимыми императивными нормами закона. Пункт 4 статьи 1286.1 ГК РФ относится к предоставлению прав за пределами открытой лицензии и не является общим правом произвольного отзыва у соблюдающего условия пользователя.

9. В отношении принадлежащих ему патентных прав, без использования которых невозможно выполнить обязательные требования данной опубликованной редакции, лицензиар принимает безвозмездное обязательство предоставить любому

реализатору простую неисключительную лицензию на изготовление, применение, предложение к продаже, продажу, ввоз и иное введение соответствующих изделий в оборот, а также использование соответствующего способа. Объем ограничен необходимыми требованиями этой редакции; территория и срок соответствуют территории и оставшемуся сроку действия конкретного патента. Лицензиар не предъявляет к добросовестному реализатору требования о платежах или прекращении такого использования на основании контролируемых им необходимых патентных прав и обязуется оформить требуемый письменный договор без лицензионных платежей и ограничения круга реализаторов. После установления конкретного патента указываются его номер, существенные пункты формулы и способы использования; стороны выполняют государственную регистрацию предоставления права в случаях статей 1232 и 1369 ГК РФ. Настоящий пункт не подменяет требуемую регистрацию уже состоявшимся предоставлением права и не распоряжается патентами третьих лиц.

10. Название протокола допускается использовать для достоверного указания поддерживаемой редакции и профиля. Измененный несовместимый вариант обозначается отдельно и не выдается за эту редакцию. Упоминание совместимости не означает одобрения или сертификации лицензиаром. Этот документ не удостоверяет регистрацию названия в качестве товарного знака и не устанавливает исключительного права на описательное словосочетание.

Правовая основа условий использования: ГК РФ, часть четвертая, статьи 1232, 1235, 1236, 1259, 1265, 1286.1, 1367 и 1369. Полный русский текст лицензии приведен здесь и в файле license-0.21-ru.txt; обращение к иностранному тексту лицензии не требуется.

Таблица 76 - Правовая основа лицензии

Правовой источник	**Применение**
ГК РФ, часть четвертая — Роспатент	Пределы авторского права и открытая лицензия.

Нормативный комплект фиксируется файлами specification, vocabulary, schemas и vectors, связанными контрольными суммами манифеста публикации. JS служит редактируемым источником, но реализация не обязана выполнять JavaScript. Форматирование чисел детерминировано и не использует Intl/локаль системы; генерация JSON сравнивается при разных LC_ALL и TZ. Побайтовая воспроизводимость

PDF не заявляется: опубликованный бинарный файл имеет собственную контрольную сумму, а полнота текста сверяется с общим источником.

Учет патентных ограничений

Авторские права на материалы и патентные права регулируются отдельно. Пункт 9 условий использования устанавливает собственное обязательство лицензиара по необходимым контролируемым им патентным правам, отказу от притязаний и оформлению безвозмездной лицензии с требуемой регистрацией; он не объявляет регистрационные действия уже выполненными. Редактор ведет открытый реестр известных заявленных ограничений: номер патента или заявки, правообладатель, затрагиваемые пункты, территория, срок и источник сведений. Предложение изменения содержит раскрытие известных автору существенных ограничений либо указание, что такие сведения ему не известны. Отсутствие записи не доказывает отсутствие прав третьих лиц.

Для включения решения, использование которого требует патентных прав участника, правообладатель представляет отдельное письменное обязательство предоставить любому реализатору необходимые права без лицензионных платежей и на одинаковых условиях. Публикуются перечень контролируемых прав, разрешенные действия, территория, срок, порядок заключения договора и предусмотренной законом регистрации. Такое обязательство действует только от имени подписавшего его правообладателя; участие в обсуждении не означает распоряжения чужими правами.

При известном существенном ограничении без указанных условий решение не включается в обязательный профиль: выбирается независимый вариант либо расширение не получает статус совместимого. Сообщение о новом ограничении регистрируется с анализом влияния, условиями правообладателя и решением по редакции. Публичные контакты для предложений используются и для раскрытия ограничений. Обещание отсутствия чужих патентов не дается.

Открытость и порядок развития

Открытый протокол систем оповещения — открытая спецификация протокола для систем оповещения. Текущий криптографический профиль предназначен для российского рынка. Открытость означает общедоступность документа, свободную независимую реализацию, переработку и отсутствие обязательной маркировки

изготовителем спецификации. Выбор российских алгоритмов сам по себе не подтверждает выполнение требований регуляторов или сертификацию СКЗИ.

Предложения от любого участника принимаются через опубликованные контакты сайта gksotcom.ru с темой «Открытый протокол систем оповещения: изменение». Предложение указывает редакцию, пункт, проблему, предлагаемый текст, влияние на совместимость и воспроизводимый пример. Редактор присваивает номер ОА-год-порядковый номер и публикует предложение, решение и обоснование в журнале изменений; контактные данные и секреты без согласия автора не публикуются. Отклонение требует технического или правового основания. Членство, закупка оборудования и договор о неразглашении для участия не требуются.

Изменение нормативного контракта публикуется как предложение не менее чем на 30 календарных дней до следующего стабильного выпуска. Исправление опасной ошибки может выпускаться немедленно с описанием несовместимости. Редактор отвечает за выпуск и не объявляет свое решение голосованием независимого органа. Независимая ветвь разрешена лицензией; несовместимая ветвь получает собственный идентификатор профиля и не выдается за исходную редакцию.

Редакция 0.21 устанавливает профиль open-alert-draft/0.21, речевой профиль rtp-alert/5 и пакет доверия open-alert-trust/2. Комплект идентифицируется редакцией и SHA-256 манифеста publication-0.21.json. Несовместимое изменение формата или семантики требует нового идентификатора wire-профиля; исправление оформления его не меняет. При проверке соответствия указывается манифест использованного комплекта.

Выбранные правила файлового оформления: ГОСТ Р 1.5-2012, пункты 5.2-5.4, 5.6.3 и 5.9. Основной текст Arial 14 пт, приложения и примечания 12 пт; поля 20 мм; межстрочный интервал 1,5. В обычных русских словах применяются переносы; в заголовках, идентификаторах, числах и значениях XML переносы запрещены. Код XML использует отдельный стиль Arial 12 пт с отступами вложенности. Единственный пример с неразрывной 128-значной подписью размещен на отдельном альбомном листе А3 по выбранному допуску ГОСТ 1.5-2001, пункт 6.3.1; остальные страницы имеют формат А4. Значение подписи не сокращается и не разбивается. Обозначения национального стандарта и реквизиты государственного утверждения не используются.

Приложение Ж

(обязательное при поддержке групповой передачи)

Групповая передача речи

Настоящее приложение устанавливает полный контракт дополнительной групповой передачи речи. Базовая одноадресная связь сохраняется; объявленная поддержка group-audio/2 либо group-audio-open/1 требует соблюдения всех относящихся к выбранному профилю положений приложения.

Ж.1 Назначение и совместимость

Ж.1.1 Групповая передача дополняет базовую одноадресную связь с каждым УУ. Она предназначена для одной речи, одновременно адресованной многим получателям, когда сеть поддерживает маршрутизацию групп IPv4. Групповой пакет не выдает владение, не запускает ОУ и не заменяет индивидуальные результаты. Обязательная совместимость основного протокола не требует поддержки этого расширения.

Ж.1.2 group-audio/2 передает открытый звук по UDP, проверяет метку Кузнечик-MGM без шифрования и подпись источника ГОСТ Р 34.10-2012/GC256A со Стрибог-256. Подпись доказывает источник даже перед получателем, знающим ключ потока. Команды подготовки и запуска защищены TLS. group-audio-open/1 передает звук без метки и подписи только при явном plain и optional. Скрытое переключение запрещено.

Ж.1.3 Маршрутизаторы и узлы распределения могут копировать исходные датаграммы без их изменения. Перекодирование или новое подписание создает другой источник и новый поток с отдельным разрешением. Для защищенного варианта подпись исходного разрешенного источника проверяет каждый УУ до декодирования. Доверие промежуточному узлу не заменяет эту проверку.

Ж.2 Конфигурация и подготовка

Ж.2.1 Неизменный объект mediaGroups пакета конфигурации задает id, revision, deviceId, sourcePrincipalId, security=tls|plain, sourceAddress, sourcePort, groupAddress, destinationPort, ttl и resources. Только при security=tls обязательны sourceKeyId, publicX, publicY; при plain эти поля запрещены. id/sourceKeyId — token до 32 байт; revision — uint64 больше нуля; идентичности — 32 строчных hex; publicX/publicY — 64 строчных hex целочисленных координат GC256A в порядке big-endian,

проверяемые на принадлежность кривой и подгруппе. resources содержит 1..4 различных resourceId/bindingGeneration данного deviceId; физическая привязка проверяется до commit. Открытые ключи не являются секретами. Изменение защищенного шаблона, источника и его ключа требует дополнительно права trustAdmin по TLS либо местной процедуры ввода при off. Открытый канал не устанавливает такое доверие; изменение иных полей конфигурации не дает обхода этой проверки.

Ж.2.2 sourceAddress — разрешенный одноадресный IPv4; groupAddress — административная группа 239.0.0.0/8; sourcePort/destinationPort — 1..65535, ttl — 1..255. Сеть явно разрешает маршрут от этого источника до группы. До успешного ответа audio.group.prepare УУ выбирает локальный приемный интерфейс, открывает UDP-прием и выполняет присоединение IGMPv3 INCLUDE(sourceAddress) к groupAddress; стек ДОЛЖЕН принять подписку и поставить первоначальный отчет в очередь отправки. Если сеть использует IGMPv2, ДОПУСКАЕТСЯ присоединение к группе с обязательной программной фильтрацией источника. Неудача местного присоединения дает sourceUnavailable, без prepared. Это подтверждает местную подготовку, а не доставку по всем коммутаторам: фактический прием блока 0 проверяется в пределах prepareTimeoutMs. Источник начинает поток после получения prepared от выбранных получателей; неподготовленный получатель исключается явно. До session.start УУ не выдает звук. После discard, истечения подготовки, завершения, отзыва или reboot подписка снимается, когда не осталось другого действующего местного контекста этой группы. IGMP и фильтрация адреса не доказывают автора. Запоздалое присоединение к уже звучащему потоку в этом профиле НЕ ДОПУСКАЕТСЯ: ТРЕБУЕТСЯ новая подготовка и новый поток.

Ж.2.3 audio.group.prepare является изменяющей командой владельца. Ее единственный элемент group содержит id, revision, streamId, source=recording| microphone и, только для recording, totalSamples. В режиме tls обязателен key — 64 строчных hex нового случайного 256-битного ключа данного потока; при plain key запрещен. Группа выбирает ресурсы из конфигурации, поэтому запрос не повторяет их длинное описание. Действуют bootId/controlEpoch/cmdSeq, права на все ресурсы и равенство sourcePrincipalId текущему участнику; для plain сравнивается только объявленный clientId без доказательства личности.

Ж.2.4 Режим шаблона совпадает с режимом управляющего соединения; несовпадение дает forbidden. УУ выбирает group-audio/2 при tls и group-audio-open/1 при plain, проверяет поддержку соответствующего профиля, конфигурацию, часы и квоты, затем возвращает prepared mediaId/expiresAtMs/configRevision/

capabilityRevision. mediaId индивидуален для УУ; streamId общий у получателей одного потока, содержит 32 строчных hex и создается случайно заново при каждой подготовке источника. Один streamId не используется повторно. Срок подготовки — 60000 мс. Повтор команды не продлевает срок. Во всех получателях один streamId обозначает одинаковые кодек, источник, ключ и шкалу блоков; иное значение отвергается как conflict.

Ж.2.5 group-audio/2 РАЗРЕШАЕТСЯ только с защищенным управлением. Ключ передается каждому УУ по его собственному TLS-соединению с парным секретом, не включается в публичные пакеты доверия, конфигурацию, журналы и диагностические выгрузки. Хранятся лишь необходимый оперативный ключ и хеш нормализованного запроса для дедупликации; ключ уничтожается при discard, завершении, отзыве, истечении подготовки или reboot. Он не является общим секретом парка и не используется для управления, другой группы или нового потока. Приватный ключ подписи остается только у источника.

Ж.2.6 audio.state дополнительно содержит groupStats с uint64 recoveredParts, receivedParityParts и missingDataParts. receivedPackets считает принятые проверенные звуковые части 0..3, receivedParityParts — части 4; FEC-восстановление увеличивает recoveredParts, а невосстановленный к сроку пропуск — missingDataParts и lostPackets. Поздние и повторные датаграммы учитываются отдельно и не увеличивают принятые части. clock имеет общие показатели частоты ЦАП; отсутствие оценки до начала звука не подменяется нулевой погрешностью. Индивидуальный session.get сохраняет критерии исполнения и полноту звука.

Ж.2.7 session.start(mode=stream) с полученным mediaId и точными resources запускает обычный автомат исполнения; onControlLoss=stop. audio.discard, audio.state, audio.finish и session.stop/cancel сохраняют общие назначения. media.hello для группового профиля не используется. В audio.finish lastPacketIndex означает последний blockIndex, а sampleCount — число отсчетов источника. У каждого получателя session.start, session.stop, ошибки и результат самостоятельны; потеря связи с одним не создает успех остальным.

Ж.3 Формат датаграммы и подпись источника

Таблица 77 - Ж.3 Формат датаграммы и подпись источника

Смещение / длина	Поле и представление
0 / 4	ASCII GA02 для защищенного профиля, GO01 для открытого.
4 / 16	streamId: 16 байт, полученных декодированием 32 hex.
20 / 4	blockIndex: uint32 BE, начинается с 0, увеличивается на 1; переполнение запрещено.
24 / 8	capturedUtcMs: uint64 BE, время захвата первого отсчета блока по UTC источника.
32 / 2	validSamples: uint16 BE, число действительных отсчетов всего блока: 3200, кроме конечного.
34 / 2	clockErrorMs: uint16 BE, верхняя погрешность UTC источника, 0..500.
36 / 1	shardIndex: 0..3 для звуковых частей, 4 для восстановительной части.
37 / 1	flags: бит 0 — END; остальные биты равны нулю.
38 / 2	Резерв: два нулевых байта.
40 / 165	В обоих режимах: открытая звуковая или восстановительная часть, без шифрования.
205 / 16	Только защищенный профиль: полная метка MGM.
221 / 64	Только защищенный профиль: подпись r32 LE || s32 LE, как в общем профиле подписи.

Ж.3.1 Длина UDP payload равна ровно 285 байтам при GA02 и 205 при GO01; иных длин, расширений или IP-фрагментации нет. MTU тракта ДОЛЖЕН вмещать датаграмму с IP/UDP и применяемой инкапсуляцией. Алгоритмы подписи, проверки точки, диапазонов r/s и хеша совпадают с основным документом. Подписывается ASCII OA-GROUP/2, затем нулевой байт, затем первые 221 байт датаграммы в исходном порядке. Подпись не включает сама себя. Перевод текста, пересериализация заголовка и разворот половин подписи запрещены.

Ж.3.2 MGM используется только для аутентификации: ключ потока, тег 16 байт, associated data - первые 205 байт датаграммы, включая заголовок и открытый звук; длина шифруемого текста равна нулю. ICN: семь нулевых байтов ||

blockIndex:uint32 BE || shardIndex:uint8 || четыре нулевых байта. Срок потока не превышает 86400000 мс и 216000 блоков; затем нужны новые streamId, ключ и подготовка. ICN уникален для каждой пары blockIndex/shardIndex под новым ключом. Повтор допускает только точную копию датаграммы; повторная аутентификация измененного содержимого с тем же ICN запрещена.

Ж.3.3 Порядок приема: проверка входного интерфейса и точного совпадения IPv4 sourceAddress, destination=groupAddress, UDP sourcePort и destinationPort с подготовленным шаблоном; затем длина и фиксированные поля, наличие действующего streamId, допустимое окно индексов и времени, затем MGM, затем проверка подписи разрешенного sourceKeyId, затем размещение части. Отказ MGM прекращает обработку до дорогой проверки подписи; успешная MGM не заменяет подпись. До успешных криптографических проверок не обновляются окно повторов, таймер аудио и состояние воспроизведения. Знание группового ключа, IP-адреса, идентификатора потока или метки MGM не заменяет подпись. Открытый профиль сохраняет проверки формата и контекста, но не дает гарантии происхождения или целостности.

Ж.4 Блоки, восстановление потерь и очереди

Ж.4.1 Кодек — GSM 06.10, RFC 3551, 8000 Гц, один канал. Блок охватывает 400 мс, то есть 3200 отсчетов и 20 кодековых кадров по 33 байта. Он разделяется на четыре последовательные части по 165 байт: пять кадров, 800 отсчетов и 100 мс. Часть 4 равна побайтовому XOR четырех открытых звуковых частей. Заголовки пяти частей имеют одинаковые streamId, blockIndex, capturedUtcMs, validSamples, clockErrorMs и flags, различается только shardIndex. Все пять частей отдельно защищаются.

Ж.4.2 Без потерь воспроизводятся проверенные части 0..3. При потере ровно одной звуковой части и наличии трех других плюс части 4 восстанавливаются ее 165 байт XOR. Доказательство источника восстановленных байтов состоит из подписанных и прошедших MGM трех частей и восстановительной части того же блока; отдельная выдуманная подпись отсутствующей части не создается. Общие поля каждой новой проверенной части сверяются с уже принятой частью того же блока. Если все пять частей доступны до выдачи, до первого отсчета проверяется нулевой XOR. Несогласованные поля или ненулевой XOR означают ошибку источника: дальнейшая выдача прекращается, невоспроизведенные очереди очищаются, сеанс завершается failed/mediaIncomplete при подтвержденном off. Уже выданный звук не

отменяется задним числом и сохраняется как состоявшееся исполнение с неполным завершением. Четыре целые проверенные звуковые части разрешено воспроизвести без ожидания восстановительной; ее отсутствие само по себе не является ошибкой. Если противоречие обнаружено после начала выдачи еще актуального блока, применяется то же прекращение остатка; опоздавшая часть уже закрытого блока отбрасывается без пересмотра законченной истории. Две потери этим кодом не восстанавливаются. Подпись общего ключа группы не используется: получатель не имеет ключа подписи источника.

Ж.4.3 Источник накапливает один блок и отправляет пять частей равномерно с интервалом 80 мс, начиная не позднее 400 мс от первого захваченного отсчета. Захват следующего блока идет параллельно; в сети передается 12,5 датаграммы в секунду. Паузы микрофона кодируются как звук тишины, VAD/DTX запрещены. После захвата последнего частичного блока он отправляется по тому же порядку; END=1, validSamples=0..3200. Последний используемый GSM-кадр дополняется тишиной до 160 отсчетов, неиспользуемые кадры и части содержат нулевые байты и не декодируются. При нулевой речи передается конечный блок 0 с validSamples=0.

Ж.4.4 Получатель хранит не более четырех блоков и 20 проверенных частей. Положение блока задается blockIndex*3200; заполнение начинается с блока 0. Начальный буфер 800 мс, максимум 1600 мс звука; отдельно считаются заголовки, защита, очереди и ЦАП. Окно принимает индексы от nextBlock до nextBlock+3. Поздний блок ниже nextBlock не воспроизводится; дальний индекс не сдвигает окно. Неверная подпись не резервирует индекс. Входная очередь до проверки — не более 5 датаграмм, приемник обеспечивает 25 проверок подписи в секунду с сохранением обслуживания session.stop. Это проверяемая квота, а не обещание неуязвимости к насыщению канала.

Ж.4.5 Восстановление выполняется только до срока выдачи соответствующих отсчетов. Пропуск, который уже нельзя восстановить, маскируется тишиной и учитывается один раз в lostPackets и итоговой полноте; позднее прибытие не повторяет звучание. FEC-восстановленная часть учитывается как восстановленная и не является потерей звука. Повторных запросов NACK и подтверждений ACK на медиадатаграммы нет. Удаление невоспроизведенных данных из-за переполнения не скрывается: поток прекращается с mediaIncomplete, подтверждается off.

Ж.5 Время, окончание и отзыв

Ж.5.1 Источник и УУ имеют UTC с подтвержденной верхней погрешностью не более 500 мс. Способ синхронизации фиксируется в паспорте и проверяется на фактическом канале; при tls источник времени ДОЛЖЕН быть аутентифицирован либо локально доверен. Просто полученная метка времени без учета задержки не доказывает эту погрешность. При отсутствии доказанной границы подготовка получает sourceUnavailable; утрата границы во время работы дает mediaClockFault. Возраст оценки часов вместе с дрейфом включается в публикуемую границу; неподтвержденное NTP-время не считается точным.

Ж.5.2 Верхняя оценка возраста первого отсчета A=nowUtcMs+receiverErrorMs-capturedUtcMs+clockErrorMs. При A<-2 либо A>5000 блок отвергается; вычисления знаковые, без переполнения. При выдаче проверяется возраст именно выдаваемых отсчетов, с учетом их положения в блоке. Для последнего действительного отсчета части j=min(validSamples,800*(shardIndex+1))-1, где j — номер от начала блока. Его расчетное время захвата capturedUtcMs+j/8 мс; верхнее время фактической выдачи ЦАП обозначено emitUtcMs+receiverErrorMs. Проверяется -16 <= 8*(emitUtcMs+receiverErrorMs-capturedUtcMs+clockErrorMs)-j <= 40000; арифметика целочисленная знаковая без округления. Проверка распространяется на все отсчеты, включая первый; заранее рассчитанный срок учитывает аппаратную очередь, ресемплинг и дрейф. При нарушении отсчет не выдается, поток прекращается с mediaClockFault. Для части с нулем действительных отсчетов проверки выдачи нет. Для восстановленной части используется ее индекс0..3, а не индекс4 пакета XOR. При откате UTC либо скачке, нарушающем подтвержденную границу, сеанс останавливается с mediaClockFault; уже начатая остановка не отменяется исправлением часов. capturedUtcMs отражает захват звука, а не отправку очереди.

Ж.5.3 END и audio.finish задают одну конечную границу: sampleCount=lastPacketIndex*3200+validSamples конечного блока. Для recording она совпадает с totalSamples подготовки. Противоречие дает conflict для команды либо mediaIncomplete для потока. Достаточно одного принятого объявления границы, однако успех полного воспроизведения требует всех отсчетов до нее, полученных либо восстановленных в срок. audio.finish повторяется индивидуально до подтверждения в общем порядке; потеря END не превращается в бесконечное ожидание. После первого принятого конца отводится не более 5000 мс на данные и выдачу хвоста,

при сохранении maxDurationMs и предела возраста. Короткий полный поток начинает выдачу после получения всех его частей без ожидания 800 мс.

Ж.5.4 Непрерывное отсутствие пригодного нового аудио 5000 мс вызывает sourceUnavailable для микрофона либо mediaIncomplete для записи. Подпись дубликата, FEC без новых пригодных отсчетов и служебный опрос не продлевают этот срок. Каждый УУ применяет свой lease и session.stop; пакет группы не возобновляет отозванный mediaId. session.stop очищает звуковые и аппаратные очереди в физических бюджетах основного документа, без дренирования.

Ж.5.5 При исключении получателя или компрометации ключа источник прекращает прежний поток, создаются новые streamId и ключ, оставшиеся получатели проходят новую индивидуальную подготовку и session.start. Передача нового звука со старым ключом после отзыва запрещена. Отозванный публичный ключ источника прекращает прием немедленно. Утерянное подтверждение session.stop сохраняет неопределенный физический результат данного УУ; новое разрешение не отменяет блокировку его ресурса.

Ж.6 Полоса и обратная связь

Ж.6.1 Защищенная датаграмма требует 285 байт UDP payload, 8 UDP, 20 IPv4 и 38 Ethernet с преамбулой и межкадровым интервалом: 351 байт на тракте без VLAN и туннелей. Пять частей на 400 мс, включая восстановительную, дают 35,1 кбит/с. С резервом 30% — 45,63 кбит/с; с выделенными 12 кбит/с управления — 57,63 кбит/с. Открытый вариант: 271 байт, 27,1 кбит/с, 35,23 с резервом. Дополнительные заголовки и фактические потери включаются в расчет отдельно. Если они превышают остаток, данный поток на канале 64 кбит/с НЕ ДОПУСКАЕТСЯ.

Ж.6.2 На общем участке маршрутизируемого группового дерева передается одна копия потока, на конечных ветвях — по одной. Число получателей не умножает исходящий медиатрафик источника, но влияет на состояние маршрутизаторов, команды подготовки и результаты. Протокол не обещает одновременно передать тысячи индивидуальных команд через один общий канал 64 кбит/с за срок одного кадра. Для таких групп рассчитываются общие участки, параллельная подготовка и емкость узлов распределения; сам по себе multicast не сокращает адресное управление.

Ж.6.3 Начало потока не порождает массового группового запроса ответа. Готовность, ошибки, audio.state и итог читаются по индивидуальным управляющим соединениям с ограничением частоты у АРМ. На каждом узком участке резервирует-

ся полоса управления; опросы распределяются по времени с равномерным случайным смещением в интервале опроса, session.start/session.stop имеют приоритет. Итоговый отчет содержит также устройства, от которых результата нет. Ни размер группы, ни отсутствие сообщения об ошибке не считаются квитанцией успеха.

Ж.7 Критерии приемки

Ж.7.1 Проверяются точные байты заголовка, MGM, подписи и ICN, несовпадение streamId, измененная половина подписи, неверная точка ключа и повтор ICN с измененным содержимым. Получатель с групповым ключом и без приватного ключа источника не создает принимаемую поддельную речь.

Ж.7.2 Для каждого из четырех положений теряется одна часть и восстанавливаются ровно исходные байты. Проверяются две потери, повреждение parity, повторы, перестановка, далекий индекс, позднее восстановление, конечный частичный и пустой блок.

Ж.7.3 Проверяются пределы возраста и ошибки UTC, задержка очередей до ЦАП, потеря синхронизации, непрерывная тишина, потеря END и audio.finish, session.stop во время накопления и выдачи, отзыв одного получателя и источника.

Ж.7.4 На фактическом тракте измеряются скорости обоих направлений, задержка, полоса управления, 25 проверок подписи в секунду и аппаратные сроки. Испытание группы проверяет отсутствие ответов на каждый медиапакет и индивидуальную полноту итогового отчета.

Библиография и комплект редакции

ГОСТ Р 42.3.05-2023. Протоколы информационного обмена. Общие требования

https://meganorm.ru/mega_doc/norm/gost-r_gosudarstvennyj-standart/8/gost_r_42_3_05-2023_natsionalnyy_standart_rossiyskoy.html

Федеральный закон № 162-ФЗ, статья 26. Правила применения документов стандартизации

https://base.garant.ru/71108018/3ac805f6d87af32d44de92b042d51285/

RFC 5740. Надежная групповая доставка и ограничение обратной связи

https://www.rfc-editor.org/rfc/rfc5740.html

RFC 8785: каноническое представление JSON

https://www.rfc-editor.org/rfc/rfc8785.html

Р 1323565.1.030-2020 и RFC 9367: российские алгоритмы в TLS 1.3

https://tc26.ru/news/novosti-tk26/opublikovan-rfc-9367-opredelyayushchiy-poryadok-raboty-tls-1-3-s-gost.html

RFC 9367: российские криптонаборы TLS 1.3

https://www.rfc-editor.org/rfc/rfc9367.html

RFC 7801: Кузнечик

https://www.rfc-editor.org/rfc/rfc7801.html

RFC 9058: MGM

https://www.rfc-editor.org/rfc/rfc9058.html

RFC 6986: Стрибог

https://www.rfc-editor.org/rfc/rfc6986.html

RFC 7836: российские функции выработки ключей

https://www.rfc-editor.org/rfc/rfc7836.html

RFC 8449: ограничение размера записей TLS

https://www.rfc-editor.org/rfc/rfc8449.html

RFC 9325: безопасное применение TLS

https://www.rfc-editor.org/rfc/rfc9325.html

RFC 2119 и нормативные уровни требований

https://www.rfc-editor.org/rfc/rfc2119.html

RFC 8174: применение нормативных ключевых слов

https://www.rfc-editor.org/rfc/rfc8174.html

RFC 8446: TLS 1.3

https://www.rfc-editor.org/rfc/rfc8446.html

RFC 7250: raw public keys

https://www.rfc-editor.org/rfc/rfc7250.html

RFC 9147: DTLS 1.3

https://www.rfc-editor.org/rfc/rfc9147.html

RFC 3711: SRTP

https://www.rfc-editor.org/rfc/rfc3711.html

RFC 3552: угрозы, аутентификация и полномочия

https://www.rfc-editor.org/rfc/rfc3552.html

RFC 8342: желаемая конфигурация и операционное состояние

https://www.rfc-editor.org/rfc/rfc8342.html

CAP 1.2: модель предупреждения

https://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2-os.html

EDXL-DE 2.0: распространение сообщений

https://docs.oasis-open.org/emergency/edxl-de/v2.0/edxl-de-v2.0.html

XML 1.0: синтаксис

https://www.w3.org/TR/xml/

RFC 9293: TCP

https://www.rfc-editor.org/rfc/rfc9293.html

RFC 3551: форматы PCMA и GSM для RTP

https://www.rfc-editor.org/rfc/rfc3551.html

RFC 4571: обрамление RTP в соединении TCP

https://www.rfc-editor.org/rfc/rfc4571.html

RFC 3550: RTP

https://www.rfc-editor.org/rfc/rfc3550.html

RFC 6241: NETCONF, пример транзакционной настройки

https://www.rfc-editor.org/rfc/rfc6241.html

RFC 5116: AEAD

https://www.rfc-editor.org/rfc/rfc5116.html

Таблица 78 - Библиография и комплект редакции

Артефакт	**Имя файла**
Пределы числовых полей и пакета	numeric-bounds-0.21.json
Связь требований с нормативными положениями	requirement-trace-0.21.json
Сквозные трассы управления и повторов	control-traces-0.21.json
XML-схема управляющих сообщений	wire-0.21.xsd
Реестр структур XML и команд	wire-contract-0.21.json
Групповая речь: криптографические векторы	multicast-vectors-0.21.json
Групповая речь: нормативный текст и числовые параметры	multicast-0.21.json
Полный русский текст открытой лицензии	license-0.21-ru.txt
Исходные тексты проверок совместимости	reference-checks-0.21.zip
Векторы парных ключей и рамок часов	security-vectors-0.21.json
Схема первичного доверия и ключей издателя	trust-schema-0.21.json
Проверочные значения подписи ГОСТ и кодирования	signature-vectors-0.21.json

Окончание таблицы 78

Артефакт	**Имя файла**
Проверочные значения Стрибог-256	digest-vectors-0.21.json
Подтверждение исполнения и исходы остановки	execution-vectors-0.21.json
Предельные размеры команд и записей	wire-budget-0.21.json
Двоичные кадры и окна bulk/1	bulk-vectors-0.21.json
Граничные значения и эталонные трассы	conformance-0.21.json
Закрытая схема конфигурации open-alert-config/5	config-schema-0.21.json
Межузловые сообщения node-alert/2	node-schema-0.21.json
Документ PDF, редакция 0.21	open-alert-0.21.pdf
Документ DOCX, редакция 0.21	open-alert-0.21.docx
Состав редакции и контрольные суммы	publication-0.21.json
Структурированный текст спецификации	specification-0.21.json
XML-примеры	examples-0.21.json
Словарь типов и ошибок	vocabulary-0.21.json
Параметры защиты управляющего канала	security-0.21.json

Журнал редакции

0.21 / 22 октября 2026

Редакция 0.21 переводит живую одноадресную речь на rtp-alert/5 по UDP. Команды подготовки и запуска защищены TLS, звуковые байты не шифруются. Метка HMAC-Стрибог-256 удостоверяет пакеты; закреплены сеансовый ключ, привязка маршрута, окно повторов и подтверждение часов при потерях. Групповой профиль group-audio/2 также передает открытый звук с меткой и подписью источника. Старые rtp-alert/4, group-audio/1 и media.hello несовместимы с этой редакцией. Сохраняются явный plain, правила владения, физического результата, российская криптография и

свободный выбор внутреннего интерфейса. Обновлены примеры, схемы, проверки и единый PDF. Уточнение от 18 октября 2026 г. согласует общий срок TLS 60000 мс, protocol.hello 10000 мс, простой управляющего канала 30000 мс и отдельный heartbeat без продления владения. Клиенты сохраняют общий backoff TCP/TLS/ resume по deviceId. Формат кадров и криптографические алгоритмы сохранены; комплект различается по дате и SHA-256 манифеста. Уточнение от 19 октября добавляет явно согласуемый startup=outputReady для rtp-alert/5: АРМ начинает источник после подтверждения речевого режима ОУ, а начальный буфер отсчитывается от первого принятого звукового пакета. Пределы подготовки и свежести речи сохранены. Уточнение от 20 октября определяет нормальное завершение собственного цикла речевого ОУ с signalKind=builtIn и stopMode=command. Потери UDP учитываются отдельно от результата завершения: полный проход известного конца с подтвержденным off допускает предупреждение mediaPacketLoss. Изменение очереди не запускает таймер отказа. Предварительное декодирование не накапливает паузы на пропусках, а отсоединение управления сохраняет уже связанный поток до истечения lease. Повтор чтения результата не вызывает повторного START. Форматы XML и UDP сохранены. Дополнение от 22 октября вводит согласуемый sourceSwitch=true: holdOutput=true сохраняет речевой сеанс при EOF или audio.pause, а новый источник запускается через audio.prepare и audio.start без повторного включения ОУ. Срок владения, полный STOP и исходный предел исполнения сохраняются; прежние клиенты продолжают прежний порядок. Дополнение от 22 октября задает согласуемые инициативные уведомления eventsPush=1: events.subscribe/events.unsubscribe, самостоятельный notification от УУ и notificationAck от АРМ по действующему соединению. Сохранены журнал, восстановление после gap, приоритет управления и совместимость с клиентами без подписки.

Текст формируется из тех же данных, что онлайн-документ. Различается оформление и нумерация разделов. Параметры протокола и XML-примеры не переписываются вручную. Контрольные суммы комплекта приведены в publication-0.21.json.

Организация-разработчик: ООО "СОТКОМ-ЮГ"

Ключевые слова: оповещение населения, протокол, АРМ, устройство управления, оконечное устройство, сеанс, телеметрия, информационный обмен.