Искра Мессенджер – корпоративный мессенджер с ИИ версия 1.0

Архитектурная спецификация для корпоративного локального внедрения

Содержание

1. Назначение документа

Документ устанавливает поддерживаемую промышленную архитектуру Искры как корпоративной платформы, устанавливаемой в инфраструктуре заказчика. Он предназначен для архитекторов, служб информационной безопасности, DevOps/SRE-команд, владельцев эксплуатации и технических руководителей проекта внедрения.

Спецификация фиксирует:

- из каких контуров состоит Искра;
- какие компоненты являются обязательными, а какие подключаются по сценарию;
- где проходит граница доверия между клиентами, серверным контуром, ИИ, медиа и внешними каналами;
- какие данные хранятся в PostgreSQL, Redis/NATS и объектном хранилище;
- как масштабируются Gateway, контур реального времени, ИИ-агенты, медиа и процессы фоновой обработки;
- какие требования предъявляются к сети, хранилищу, секретам, мониторингу, резервному копированию и DR;
- где проходит граница ответственности между поставщиком и командой эксплуатации заказчика.

Документ не является инструкцией по конкретной платформе оркестрации и не подменяет регламент администратора. Спецификация задает обязательные архитектурные границы, зависимости и эксплуатационные требования, которые должны быть реализованы в выбранном стандарте эксплуатации заказчика.

1.1 Статус требований

В документе используются три уровня требований:

Статус	Значение
Обязательно	Требование входит в поддерживаемую промышленную архитектуру Искры и должно выполняться для корпоративного локального внедрения
Опциональный модуль	Компонент или интеграция включается только при наличии соответствующего функционального сценария, но после включения выполняет требования этой спецификации
Профиль внедрения	Обязательный проектный артефакт для инсталляции заказчика. В нем фиксируются выбранные модули Искры, топология, зоны ответственности, лимиты, SLO, RPO/RTO, сроки хранения, параметры резервного копирования, масштабирования и интеграций

2. Обзор

Искра - коммуникационная платформа для команд, обращений, внешних каналов, звонков и ИИ-ассистентов. Пользователь работает в веб, десктоп или мобильном приложении, клиент может написать из внешнего канала, оператор видит обращение в едином интерфейсе, а ИИ-агент может принять первую линию, собрать контекст и передать разговор человеку.

Архитектура построена вокруг центрального серверного контура **Gateway**. Gateway хранит и проверяет основное бизнес-состояние: пользователей, пространства, права, диалоги, сообщения, очереди, ИИ-агентов, доставку и служебные события. В текущей реализации отдельные журналы событий ведутся в подсистемах Webchat и Event-organizer; единый платформенный журнал аудита не входит в базовое описание текущей реализации. Ресурсоемкие или интеграционные задачи вынесены в отдельные контуры:

- **клиентский слой**: Web, Desktop, Mobile, Webchat;
- **Gateway и основной домен**: авторизация, сессии, устройства, права, сообщения, очереди, доставка, API, события реального времени;
- **контур ИИ-агентов**: Assistant, среда выполнения агента, интеграции с LLM, инструменты, передача оператору;
- **контур данных и событий**: PostgreSQL, Redis, NATS, объектное хранилище;
- **медиа-контур**: LiveKit, выгрузка/запись, SIP/Kamailio при использовании телефонии;
- **каналы и bridge-сервисы**: Telegram, MAX, Webchat, телефония и другие внешние коннекторы;
- **операционный контур**: проверки работоспособности, метрики, журналы, трассировка, отчетность об ошибках, резервное копирование и восстановление.

Ключевой архитектурный принцип: **Gateway является управляющим контуром бизнес-решений, а специализированные сервисы исполняют свою часть работы, не обходя модель прав, состояния и реализованные журналы событий подсистем**.

3. Целевая локальная топология

Поддерживаемая промышленная топология Искры состоит из следующих сетевых зон. Конкретные имена VLAN, подсетей и балансировщиков задаются стандартами заказчика; архитектурные границы между зонами являются обязательными.

Зона	Компоненты	Назначение
Пользовательская зона	Web, Desktop, Mobile клиенты	Доступ сотрудников и операторов к Искре
Публичная/DMZ-зона	HTTPS входной контур, точка доступа Webchat, точки доступа вебхуков, публичная точка доступа LiveKit, SIP edge для опционального модуля телефонии	Прием внешнего трафика, завершение TLS, публикация ограниченного набора точек доступа

Прикладная зона	Gateway, Assistant, Event-organizer, Webchat service, bridge-сервисы, фоновые обработчики	Бизнес-логика, API, события реального времени, ИИ, фоновые задачи
Зона данных	PostgreSQL, Redis, NATS, объектное хранилище	Хранение состояния, очередей, событий, вложений и записей
Медиа-зона	узлы LiveKit, фоновые обработчики выгрузки, LiveKit SIP/Kamailio	Аудио/видео, записи, SIP-сценарии
Операционная зона	мониторинг, журналы, оповещения, хранилище резервных копий, отчетность об ошибках	Наблюдаемость, диагностика, операционный контроль и восстановление

Публично доступны только те точки доступа, которые нужны пользователям, внешним каналам и медиа-клиентам. Базы данных, очереди, внутренние сервисы и служебные API не публикуются наружу.

4. Общая схема

Рисунок 1. Архитектура исполнения Искры: клиенты обращаются к Gateway, Gateway управляет основными доменами и передает специализированные задачи сервисам ИИ, медиа, распознавания речи, хранилища и внешних интеграций.

На схеме важно не название каждого процесса, а граница ответственности:

- **Клиенты** обращаются к Искре по защищенному HTTPS/WebSocket соединению.
- **Gateway** является центральной точкой бизнес-логики, прав, сессий, устройств и состояния.
- **ИИ Агенты** работают как отдельная поверхность: у агента есть виртуальный пользователь, учетные данные, события, потоковые ответы и аренда обработки одного активного исполнителя.
- **Данные и очереди** сохраняют состояние и поддерживают фоновые процессы.
- **Медиа-контур** обслуживает звонки, конференции, записи и SIP-сценарии.
- **Каналы** подключают внешние источники обращений и событий.

5. Компонентная модель

5.1 Обязательные прикладные компоненты

Компонент	Роль в системе	Масштабирование
Gateway	Центральный серверный компонент: API, авторизация, сессии, устройства, права, пространства, диалоги, сообщения, очереди, доставка, события реального времени, выдача медиа-токенов	Несколько экземпляров за балансировщиком при общем слое состояния
Web	Пользовательский веб-интерфейс и слой обратного прокси для веб/API маршрутов	Несколько экземпляров, без локального состояния
Webchat service	Серверный компонент сценариев Webchat и публичных виджетов	Несколько экземпляров при общей дедупликации и общем слое состояния
Assistant	Сервис ИИ-ответов, сводок и агентов, интеграция с поставщиками LLM и управлением агентами	Несколько фоновых обработчиков, масштабирование по очередям и лимитам поставщиков
Event-organizer	Агентские и календарные сценарии, работающие через API среды выполнения ботов	Масштабирование как сервисный контур или фоновый обработчик
Transcription фоновый обработчик	Расшифровка аудио и подготовка данных для сводок	Пул обработчиков
Sticker normalizer	Нормализация медиа/стикеров	Пул обработчиков

5.2 Компоненты внешних каналов

Компонент	Роль	Особые требования
Telegram bridge	Прием/отправка событий Telegram через Gateway	Проверка вебхука и секрета источника, дедупликация внешних идентификаторов событий
MAX bridge	Прием/отправка событий MAX через Gateway	Проверка источника, идемпотентность, контроль лимитов частоты запросов
Публичная точка доступа Webchat	Обращения посетителей сайта	Сессионная защита, CORS/разрешенные источники, защита от повторов
SIP/Kamailio	Подключение телефонных сценариев и PBX	TCP/UDP доступность, NAT/публичный контакт, отдельная диагностика

5.3 Зависимости с состоянием

Зависимость	Назначение	Промышленное требование
PostgreSQL	Источник истины: пользователи, сессии, устройства, пространства, права, диалоги, сообщения, очереди, конфигурации ИИ, доставка, аренды обработки, журналы событий Webchat/Event-organizer	Надежное хранилище, резервное копирование/восстановление, мониторинг, пул соединений, контролируемые миграции
Redis	Временное состояние среды выполнения, присутствие, ограничение частоты запросов, служебные кэши	Промышленная инсталляция должна обеспечивать контролируемую доступность Redis; параметры отказоустойчивости, памяти и вытеснения данных фиксируются в профиле внедрения
NATS	События реального времени, очереди, распространение событий и сценарии потребителей	Мониторинг задержки, состояния потребителей, срока хранения, устойчивости к рестартам
Объектное хранилище / S3 /	Вложения, медиа-файлы,	Общий ресурс для всех

MinIO	записи, артефакты, OTA-файлы и файлы загрузки	экземпляров, политика жизненного цикла, резервное копирование или репликация
LiveKit	Медиа-контур: комнаты звонков, аудио/видео, сигнализация, передача медиа	Публичные точки доступа, UDP/TCP маршрутизация, мониторинг потерь пакетов, задержки и битрейта

5.4 Матрица компонентов Искры

Web / обратный прокси. Входящий интерфейс: HTTPS 443 для Web, API, маршрутов Webchat и OAuth обратных вызовов. Зависимости: Gateway, Assistant, публичный маршрут LiveKit. Состояние: без локального состояния. Отказ: пользователи не открывают Web/API через эту точку входа.

Gateway. Входящий интерфейс: HTTP 18 080 внутри контура; снаружи через HTTPS входной контур. Зависимости: PostgreSQL, Redis, NATS, объектное хранилище, LiveKit, Assistant, фоновые обработчики, bridge-сервисы. Состояние: без локального состояния; бизнес-состояние хранится в PostgreSQL. Отказ: недоступны API, авторизация, сессии, сообщения, события реального времени, выдача медиа-токенов и доставка.

Webchat service. Входящий интерфейс: HTTP 18 082 внутри контура. Зависимости: Gateway, PostgreSQL, LiveKit. Состояние: без локального состояния; сессии Webchat хранятся в PostgreSQL и защищаются секретами Webchat. Отказ: посетители сайта не создают обращения через Webchat.

Assistant. Входящий интерфейс: HTTP 8081 внутри контура; маршруты обратных вызовов OAuth публикуются через обратный прокси. Зависимости: PostgreSQL, поставщики LLM, API среды выполнения ботов Gateway. Состояние: без локального состояния; настройки агентов и состояние потоков проходят через Gateway/PostgreSQL. Отказ: не работают ИИ-ответы, сводки, оркестрация инструментов и OAuth агентов.

Event-organizer. Входящий интерфейс: HTTP 18 084 внутри контура. Зависимости: API среды выполнения ботов Gateway, Assistant, PostgreSQL. Состояние: без локального состояния, сервисный контур или фоновый обработчик. Отказ: останавливаются событийные и календарные сценарии агентов.

Telegram bridge. Входящий интерфейс: HTTP 18 110 внутри контура; вебхук публикуется через HTTPS входной контур. Зависимости: Gateway, Telegram API. Состояние: без локального состояния; idempotency ведется через Gateway/БД. Отказ: Telegram-обращения не принимаются или не доставляются.

MAX bridge. Входящий интерфейс: HTTP 18 120 внутри контура; вебхук публикуется через HTTPS входной контур. Зависимости: Gateway, MAX Platform API. Состояние: без локального состояния; idempotency ведется через Gateway/БД. Отказ: MAX-обращения не принимаются или не доставляются.

Transcription фоновый обработчик. Входящий интерфейс: HTTP 8091 API проверки работоспособности/сервисный API внутри контура. Зависимости: объектное хранилище, Gateway/PostgreSQL, поставщик распознавания речи при включении. Состояние: очередь/БД. Отказ: не создаются транскрипции и зависящие сводки.

Sticker normalizer. Входящий интерфейс: HTTP 8092 API проверки работоспособности/сервисный API внутри контура. Зависимости: Gateway, объектное хранилище. Состояние: очередь/БД. Отказ: медиа/стикеры остаются без нормализации.

LiveKit. Входящий интерфейс: HTTP/WS 7880, TCP 7881, UDP диапазон медиа. Зависимости: Redis, объектное хранилище через выгрузку, вебхук Gateway. Состояние: медиа-сессии. Отказ: сообщения работают; звонки, комнаты, записи и SIP-медиа недоступны.

LiveKit выгрузка. Входящий интерфейс: внутренний фоновый обработчик. Зависимости: LiveKit, объектное хранилище. Состояние: состояние задания. Отказ: не создаются записи звонков и экспорт медиа.

PostgreSQL. Входящий интерфейс: TCP 5432 внутри зоны данных. Зависимости: постоянное хранилище, инструменты резервного копирования. Состояние: источник истины Искры. Отказ: продукт останавливается для операций, требующих состояния.

Redis. Входящий интерфейс: TCP 6379 внутри зоны данных/среды выполнения. Зависимости: политика памяти и отказоустойчивости. Состояние: временное состояние среды выполнения. Отказ: деградируют присутствие, ограничение частоты запросов, координация медиа и кэши.

NATS. Входящий интерфейс: TCP 4222; мониторинг 8222. Зависимости: хранилище/политика JetStream при включении. Состояние: слой событий. Отказ: деградируют распространение событий реального времени, очереди и доставка потребителям.

Объектное хранилище. Входящий интерфейс: S3 API 9000; консоль 9001 при публикации операционного доступа. Зависимости: постоянное объектное хранилище. Состояние: вложения, записи, артефакты. Отказ: вложения, записи, импорт, OTA-файлы и файлы загрузки недоступны.

Prometheus/Grafana/GlitchTip. Входящий интерфейс: операционные точки доступа. Зависимости: все сервисы `/healthz` и метрики. Состояние: данные наблюдаемости. Отказ: теряется наблюдаемость и отчетность об ошибках, но не бизнес-данные Искры.

5.5 Портово-протокольная матрица

Публичные порты могут быть переопубликованы через входной контур, балансировщик нагрузки или корпоративный обратный прокси. Внутренние порты ниже являются сервисными портами Искры, используемыми в типовой схеме внедрения.

Источник	Назначение	Протокол / порт	Использование в Искре
Web, Desktop, Mobile клиенты	Публичный входной контур	HTTPS 443	Веб-интерфейс, REST API, загрузки и файлы загрузки
Web, Desktop, Mobile клиенты	Реальное время Gateway	WSS 443	События реального времени, присутствие, обновления доставки
Браузерный виджет Webchat	Публичный входной контур	HTTPS 443	Создание обращений посетителей сайта
Поставщики внешних	Входной контур	HTTPS 443	События вебхуков

каналов	вебхуков bridge-сервисов		Telegram/MAX
Обратный прокси	Gateway	HTTP 18 080	`/v1/*`, `/api/*`, `/ws`, `/embed`, `/healthz`
Обратный прокси	Assistant	HTTP 8081	Маршруты обратных вызовов OAuth: Bitrix, HeadHunter, календарные интеграции
Обратный прокси	LiveKit	HTTP/WebSocket 7880 или HTTPS 443 через домен LiveKit	Сигнализация медиа
Gateway	PostgreSQL	TCP 5432	Пользователи, сессии, устройства, пространства, права, диалоги, сообщения, аренды обработки
Gateway / Assistant / фоновые обработчики	Redis	TCP 6379	Присутствие, лимиты частоты запросов, служебные кэши, координация медиа
Gateway / сервисы	NATS	TCP 4222	Распространение событий реального времени, события фоновых обработчиков, доставка потребителям
Мониторинг	NATS monitor	HTTP 8222	Работоспособность и мониторинг NATS
Gateway	Объектное хранилище	S3 API 9000 или S3 точка доступа заказчика	Вложения, записи, импорт, артефакты
Gateway	Assistant	HTTP 8081	Вызовы ИИ, сводок и сервиса агентов
Gateway	Event-organizer	HTTP 18 084	Оркестрация событий агентов
Gateway	Webchat service	HTTP 18 082	Уведомления доставки и сессии Webchat
Gateway	Transcription фоновый обработчик	HTTP 8091	Задания расшифровки аудио

Gateway	Sticker normalizer	HTTP 8092	Задания нормализации стикеров и медиа
Assistant / Event-organizer	API среды выполнения ботов Gateway	HTTP 18 080	Сообщения агентов, состояние потоков, запросы на действия, передача оператору
Telegram bridge	Gateway	HTTP 18 080	Входящая/исходящая доставка Telegram
MAX bridge	Gateway	HTTP 18 080	Входящая/исходящая доставка MAX
Telegram bridge	Telegram API	HTTPS 443 outbound	API поставщика и файлы
MAX bridge	MAX Platform API	HTTPS 443 outbound	API поставщика
Клиенты	LiveKit media	UDP 62 000-62 050; TCP 7881 fallback	Передача аудио/видео
LiveKit	Gateway	HTTP 18 080	Обратные вызовы вебхуков медиа-комнат
SIP/PBX	Модуль SIP/Kamailio	SIP UDP/TCP 5060/5062 по профилю внедрения	Телефонные сценарии
Prometheus / blackbox	Сервисы	HTTP `/healthz` на 18 080, 18 082, 8081, 18 084, 18 110, 18 120, 8091, 8092	Проверки доступности
Операционные пользователи	Grafana	HTTPS 443 или внутренний 3000	Панели мониторинга и оповещения

6. Основные пользовательские сценарии

Единое рабочее место команды

Сотрудник может открыть Искру в браузере, Desktop-приложении или на телефоне. Все клиенты работают с одним серверным состоянием: список диалогов, очереди обращений, сообщения, звонки и уведомления синхронизируются через API и события реального времени.

Обращения из разных каналов

Клиент может прийти из Webchat, Telegram, MAX, публичного профиля или телефонного канала. Внешний канал передает событие в Искру, Gateway связывает его с пространством, очередью и диалогом, а оператор видит обращение в привычном списке разговоров.

ИИ-агент

ИИ-агент представлен в Искре как виртуальный участник, в том числе как виртуальный оператора. Он получает только разрешенный контекст, отвечает потоковым сообщением, создает запрос на действие для действий с подтверждением и передает обращение оператору, если сценарий выходит за пределы автоматизации.

Звонки и конференции

Звонки и конференции управляются из Искры, но медиапоток обслуживает отдельный LiveKit-контур. Gateway выдает права и токены, а медиа-контур отвечает за комнаты, аудио/видео, запись и SIP-связь.

7. Клиентский слой и публичные точки доступа

Искра поддерживает несколько пользовательских поверхностей.

Поверхность	Для кого	Что дает
Web	администраторы, операторы, руководители	Настройка пространства, очередей, каналов, работа с обращениями
Desktop	операторы и сотрудники поддержки	Постоянное рабочее окно, уведомления, быстрый доступ к диалогам
Mobile	сотрудники в движении, руководители, дежурные	Быстрые ответы, звонки, push-уведомления
Webchat	посетители сайта	Обращение в команду без установки приложения

Клиенты не хранят серверные секреты. Они получают пользовательские токены, работают через публичные API и получают изменения через контур реального времени. Desktop- и Mobile-клиенты рассматриваются как недоверенная среда: любая операция повторно проверяется на Gateway.

Публичный контур промышленной инсталляции включает:

- HTTPS точку доступа для Web и API;
- WebSocket точку доступа для реального времени;
- HTTPS точку доступа для Webchat и вебхуков внешних каналов;
- публичную точку доступа LiveKit для сигнализации медиа;
- UDP/TCP порты медиа для LiveKit;
- SIP точка доступа для опционального модуля телефонии.

8. Gateway как центр бизнес-состояния

Gateway - центральный управляющий контур Искры. Он отвечает за то, чтобы у каждого действия был правильный пользователь, правильные права, правильный объект и правильное состояние.

В Gateway сходятся:

- авторизация, сессии и устройства;
- пользователи, профили, пространства, участники и роли;
- диалоги, сообщения, очереди обращений, доставка и состояние непрочитанных сообщений;
- события реального времени;
- конфигурация ИИ-агентов как виртуальных пользователей;
- выдача доступов к звонкам и конференциям;
- подключение внешних каналов;
- биллинг, OTA-файлы, файлы загрузки и служебные обратные вызовы;
- журналы событий подсистем Webchat/Event-organizer.

Для корпоративного внедрения это означает, что управление строится вокруг единого центра: кто вошел, кто написал, кто ответил, какой агент участвовал, когда обращение было передано оператору, какой внешний канал создал событие и какие системные действия были выполнены.

9. Данные и владение состоянием

Искра разделяет постоянное бизнес-состояние, временное состояние среды выполнения, события и медиа-объекты.

Тип данных	Где находится	Примеры	Требования
Постоянное бизнес-состояние	PostgreSQL	пользователи, сессии, устройства, пространства, роли, диалоги, сообщения, очереди, агенты, аренды обработки, журналы событий подсистем	Резервное копирование, проверка восстановления, миграции, индексы, пул соединений
Временное состояние среды выполнения	Redis	присутствие, ограничение частоты запросов, служебные кэши	Контроль памяти, TTL, политика вытеснения
События и распространение событий	NATS	распространение событий реального времени, события потребителей, события фоновых	Срок хранения, мониторинг задержки, повторная доставка

		обработчиков	
Объекты и медиа	Объектное хранилище	вложения, записи, артефакты, OTA-файлы и файлы загрузки	Общий доступ для всех экземпляров, жизненный цикл, резервное копирование/репликация
Медиа-сессии	LiveKit	комнаты звонков, участники, SIP-потоки	Сетевой мониторинг, планирование емкости

Ключевой принцип: Gateway управляет бизнес-состоянием, а специализированные контуры обрабатывают свою часть работы. Например, LiveKit не решает, кто имеет право участвовать в звонке; это решает Искра, а затем выдает нужный доступ.

Локальный диск прикладных контейнеров или VM не должен использоваться как источник истины. Все данные, которые должны пережить рестарт, перенос или масштабирование экземпляров, должны находиться в PostgreSQL, объектном хранилище или другом согласованном сервисе с состоянием.

10. Реальное время и синхронизация

Контур реального времени нужен для того, чтобы операторы и пользователи видели изменения без перезагрузки:

- новые сообщения;
- изменение статуса обращения;
- прочтения и непрочитанные счетчики;
- события звонков;
- события набора текста и присутствия;
- потоковые ответы ИИ-агентов.

Клиентское приложение получает начальный снимок состояния через API, затем слушает WebSocket. Если соединение прерывается, клиент или среда выполнения перечитывает состояние и продолжает работу. Корректность не должна зависеть от того, к какому экземпляру Gateway подключен клиент.

Промышленное правило: события реального времени не должны жить только в памяти одного процесса. Экземпляры Gateway должны использовать общий контур распространения событий, а клиент должен уметь восстановиться через снимок состояния.

11. ИИ-агенты

ИИ-агент в Искре - это не внешняя сущность без контроля. Он представлен как виртуальный пользователь и подчиняется правилам платформы.

Архитектурно агент состоит из:

- **идентичность**: имя, аватар, видимость, участие в пространстве;
- **учетные данные**: токен бота, который хранится только на серверной стороне среды выполнения;
- **среда выполнения**: сервис, который слушает события и принимает решения;
- **контекст**: последние сообщения, служебные заметки, разрешенные данные и инструменты;
- **ответ**: потоковое сообщение, обычное сообщение, запрос на действие или передача оператору.

Для предотвращения дублей используется аренда обработки одного активного потребителя: один агент не должен отвечать из двух процессов среды выполнения одновременно. Если среда выполнения останавливается, новый процесс получает новую аренду обработки и продолжает работу.

Корпоративный контроль ИИ

ИИ-контур Искры является поддерживаемым модулем. Профиль внедрения фиксирует:

- какие пространства и очереди могут использовать ИИ-агентов;
- какие данные могут попадать в промпты и контекст;
- используется внешний поставщик LLM или внутренняя совместимая точка доступа;
- разрешена ли выгрузка из локального контура к поставщику LLM;
- какие действия инструментов требуют подтверждения человеком;
- какие действия агента должны быть доступны для расследования через состояние PostgreSQL, прикладные журналы и журналы подсистем;
- какие ограничения срока хранения действуют для промптов, контекста и служебных заметок.

Внешний поставщик LLM является отдельной исходящей интеграцией. Данные покидают контур заказчика только в рамках явно разрешенного пути интеграции. Если политика заказчика запрещает внешнюю выгрузку, ИИ-контур работает через внутреннюю LLM модель, внутреннего совместимого поставщика или отключается как функциональный модуль.

12. Инструменты и передача оператору

Не каждое действие агент должен выполнять самостоятельно. Для действий с последствиями используется модель подтверждения:

- агент создает запрос на действие;
- человек или доверенный обработчик подтверждает действие;
- результат возвращается агенту;
- агент продолжает разговор или передает обращение оператору.

Передача оператору нужен, когда:

- клиент просит человека;

- агент не уверен в ответе;
- действие требует полномочий оператора;
- сценарий выходит за пределы prompt и разрешенных инструментов;
- возникает ошибка внешнего поставщика.

Такой подход снижает риск автоматических действий без контроля и делает ИИ-агента частью команды, а не непрозрачным внешним ботом.

13. Медиа, звонки и телефония

Медиа-контур отделен от основного серверного контура. Это важно, потому что аудио/видео требует другой сетевой модели, UDP-доступности, SIP-интеграции и отдельных ресурсов.

Роли компонентов:

- Gateway создает бизнес-состояние звонка и выдает доступ;
- LiveKit обслуживает аудио/видео комнаты;
- выгрузка отвечает за запись и выгрузку;
- фоновый обработчик Transcription обрабатывает расшифровку аудио при включенной транскрибации;
- Assistant подготавливает сводку после расшифровки при включенном сценарии ИИ-сводки;
- Kamailio и LiveKit SIP подключают телефонные сценарии.

Для пользователя это выглядит как единый звонок в Искре, но технически медиа-контур масштабируется, мониторится и диагностируется отдельно.

Медиа-контур требует отдельного сетевого проектирования:

- публичная точка доступа для сигнализации;
- TCP fallback для сетей, где UDP ограничен;
- UDP-диапазон для WebRTC-медиа;
- корректная публикация публичного IP/NAT;
- отдельное планирование емкости для записей и выгрузки;
- отдельная диагностика SIP-трафика, если подключается PBX.

14. Внешние каналы

Искра поддерживает прием обращений и событий из внешних каналов:

- Webchat на сайте;
- Telegram;
- MAX;
- телефония и SIP/PBX;

- push/SMS события;
- внешние сервисы-коннекторы.

Каждый канал должен быть привязан к пространству, публичному профилю, очереди или другой точке маршрутизации. Внешний канал не должен напрямую менять внутреннее состояние без проверки: Gateway валидирует секреты, источник, права и бизнес-контекст.

Bridge-сервис должен рассматриваться как пограничный компонент. Он принимает полезную нагрузку внешнего поставщика, проверяет источник, нормализует событие, обеспечивает дедупликацию и передает нормализованное действие в Gateway.

15. Безопасность и границы доверия

Архитектура Искры исходит из нескольких правил:

- пользовательские клиенты работают по HTTPS и WebSocket;
- серверные секреты не передаются в Web/Mobile/Desktop клиенты;
- токен бота хранится только в среде выполнения агента;
- внутренние обратные вызовы защищаются общими секретами, mTLS или сетевой политикой;
- внешние каналы проходят через bridge-сервисы и слой коннекторов;
- ИИ действия инструментов должны иметь подтверждение там, где есть бизнес-риск;
- медиа-доступ выдается через Gateway, а не напрямую пользователем;
- состояние диалогов и прав хранится централизованно;
- базы данных и очереди доступны только из доверенных прикладных зон.

15.1 Аутентификация пользователей

В текущей реализации Искра использует аутентификацию пользователей в Gateway. Основная модель входа - подтверждение номера телефона одноразовым кодом; дополнительная модель - вход через приложение-аутентификатор по TOTP-коду для пользователей, у которых такой способ включен.

Поддерживаемые поверхности аутентификации:

- запрос, повторная отправка и проверка одноразового кода: `/v1/auth/request-otp`, `/v1/auth/resend-otp`, `/v1/auth/verify-otp`;
- запрос и проверка входа через приложение-аутентификатор: `/v1/auth/request-authenticator`, `/v1/auth/verify-authenticator`;
- обновление сессии: `/v1/auth/refresh`;
- выход и отзыв сессии: `/v1/auth/logout`;
- регистрация и обновление устройства пользователя: `/v1/auth/devices`, `/v1/auth/devices/me`.

Gateway хранит состояние аутентификации в PostgreSQL:

- `users`: пользователь, хэш телефона, зашифрованный номер телефона, профиль и статус;

- `otp_requests` и `otp_delivery_attempts`: запросы одноразовых кодов, попытки доставки, срок действия, остаток попыток;
- `authenticator_verification_requests`, `authenticator_enrollments`, `user_authenticator_credentials`, `authenticator_login_challenges`: проверка, подключение и использование TOTP-аутентификатора;
- `sessions` и `session_refresh_token_grace`: refresh-сессии, ротация refresh-токенов, grace-период предыдущего refresh-токена, отзыв сессии;
- `devices`: зарегистрированные устройства, платформа, push-токены, VoIP push-токены и время последней активности.

Access-токен является JWT, подписанным секретом Gateway. В токене фиксируются `sub`, `session_id`, `device_id`, `iat` и `exp`. Refresh-токен хранится у клиента как случайный секрет; в PostgreSQL хранится только его хэш. При обновлении сессии Gateway выпускает новую пару токенов и ротирует refresh-токен. При выходе сессия отзывается через запись `revoked_at`.

Для промышленного внедрения профиль внедрения должен зафиксировать:

- разрешенные способы входа: одноразовый код, TOTP-аутентификатор, корпоративный SSO при включении;
- поставщика доставки одноразовых кодов и разрешенные каналы доставки;
- срок действия access-токена, refresh-токена, одноразового кода и TOTP-запроса;
- лимиты запросов и проверок одноразовых кодов;
- требования к регистрации устройств и отзыву устройств;
- срок хранения записей `otp_requests`, `otp_delivery_attempts`, `sessions`, `devices` и данных аутентификатора;
- ответственного владельца со стороны заказчика за политику сессий и отзыв доступа.

Корпоративный SSO/IdP в этой спецификации описывается как поддерживаемый профиль внедрения, а не как обязательный способ входа по умолчанию. При включении SSO профиль внедрения должен отдельно определить IdP, протокол, сопоставление пользователей, групп и ролей, жизненный цикл пользователей, правила выхода, истечение сессии и порядок отзыва доступа. До утверждения этих параметров промышленный SSO не считается включенным.

15.2 Доступ и роли

Минимальная корпоративная модель доступа должна включать:

- персональные учетные записи пользователей;
- роли на уровне пространства и очередей;
- разделение администраторов платформы, администраторов пространства, операторов и обычных пользователей;
- отдельный доступ сервисных учетных записей;
- процедуру отзыва устройств, токенов, учетных данных ботов и секретов коннекторов.

15.3 Секреты и сертификаты

Промышленные секреты должны храниться в утвержденном хранилище секретов заказчика или в защищенном механизме развертывания. К таким секретам относятся:

- учетные данные БД;
- учетные данные Redis/NATS при включенной авторизации;
- ключи доступа объектного хранилища;
- API-ключ и секрет LiveKit;
- токены ботов;
- учетные данные поставщика LLM;
- секреты вебхуков и общие секреты внешних каналов;
- учетные данные push, SMS и биллинга;
- TLS закрытые ключи.

Секреты должны поддерживать ротацию без ручного изменения кода. Сертификаты публичных точек доступа принадлежат инфраструктурному контуру заказчика или согласованной зоне ответственности поставщика.

15.4 Шифрование и хранение данных

Трафик между пользователями и публичными точками доступа должен быть защищен TLS. Внутренние соединения должны быть защищены одним из поддерживаемых корпоративных подходов: сетевыми политиками в частной сети, mTLS или сервисной mesh-моделью. Выбранный подход фиксируется в профиле внедрения.

Шифрование данных на диске обеспечивается средствами PostgreSQL, системы хранения или инфраструктурной политики заказчика. Объектное хранилище должно поддерживать контроль доступа, жизненный цикл и защиту от случайного удаления в соответствии с требованиями заказчика.

15.5 Реализованные журналы событий

В текущей реализации Искры нет единого платформенного журнала аудита, покрывающего все действия пользователя, администратора, Gateway, ИИ-агента и медиа-контура.

Продукт сохраняет отдельные журналы событий в следующих подсистемах:

- `webchat_audit_events`: события Webchat-сессий, идентификации посетителя, WebSocket-подключения, реакций, звонков, сообщений и загрузок;
- `event_audit_log`: действия сценариев Event-organizer, включая запуск, изменение заметок, компиляцию workflow, импорт получателей, удаление получателей и другие действия внутри event-run.

Эти журналы описывают события своих подсистем и не являются единым корпоративным журналом аудита всей платформы. События Gateway, управления ролями, сессий пользователей, ИИ-агентов, медиа-комнат и внешних bridge-сервисов не должны считаться покрытыми единым аудитом, если это явно не реализовано в конкретной поставке и не указано в профиле внедрения.

Для расследований по событиям, не входящим в `webchat_audit_events` и `event_audit_log`, используются прикладные журналы сервисов, журналы инфраструктуры заказчика и данные PostgreSQL, доступные в рамках прав администратора БД и эксплуатационных процедур заказчика.

15.6 Матрица секретов Искры

Матрица секретов фиксируется в профиле внедрения по компонентам. Для каждого компонента указываются класс секрета, место использования, владелец, способ хранения и процедура ротации.

Gateway. Секреты: `JWT_SECRET`, учетные данные PostgreSQL, ключи объектного хранилища, API-ключ и секрет LiveKit, общие секреты коннекторов, ключ шифрования секретов аутентификатора. Использование: авторизация, сессии, выдача медиа-токенов, вложения, внешние каналы, одноразовые коды и аутентификатор. Ротация требует последовательного перезапуска Gateway и проверки активных сессий и токенов.

Assistant. Секреты: API-ключи поставщиков LLM, идентификатор и секрет OAuth-клиента для Bitrix, HeadHunter, календарей Google/Yandex/Apple, ключи веб-поиска. Использование: ИИ-ответы, сводки, инструменты агентов, интеграции календаря и поиска. Ротация выполняется по каждому поставщику; сценарии агентов проверяются дымовым тестом.

Webchat service. Секреты: `WEBCHAT_SESSION_SECRET`, `WEBCHAT_IDENTITY_HS256_SECRET`, секрет внешнего коннектора. Использование: сессии Webchat, идентификация посетителя, обратные вызовы доставки. Ротация требует стратегии для активных сессий Webchat.

Telegram bridge. Секреты: `TELEGRAM_BOT_TOKEN`, `EXTERNAL_CONNECTOR_SECRET`. Использование: проверка вебхука Telegram, вызовы API поставщика, доставка. Ротация требует обновления вебхука и конфигурации поставщика.

MAX bridge. Секреты: учетные данные поставщика MAX, `EXTERNAL_CONNECTOR_SECRET`. Использование: проверка вебхука MAX, вызовы API поставщика, доставка. Ротация требует обновления настроек интеграции MAX.

LiveKit / выгрузка. Секреты: `LIVEKIT_API_KEY`, `LIVEKIT_API_SECRET`, ключи объектного хранилища. Использование: проверка токенов комнат, записи, выгрузка записей. Ротация синхронизируется между Gateway, LiveKit и контуром выгрузки.

Объектное хранилище. Секреты: ключ доступа, секрет и учетные данные политики бакета. Использование: вложения, записи, импорт, артефакты, OTA-файлы и файлы загрузки. Ротация проверяется дымовым тестом загрузки и выгрузки.

PostgreSQL. Секреты: учетные данные прикладной БД, учетные данные миграций, учетные данные резервного копирования. Использование: бизнес-состояние, сессии, устройства, журналы событий подсистем, конфигурации агентов, аренды обработки. Ротация согласуется с пулами соединений и инструментами резервного копирования.

Redis / NATS. Секреты: сервисные учетные данные при включенной авторизации. Использование: состояние среды выполнения, распространение событий реального времени, очереди. Ротация требует совместимого перезапуска зависимых сервисов.

Мониторинг / отчетность об ошибках. Секреты: учетные данные администратора Grafana, секрет GlitchTip, учетные данные экспортеров. Использование: операционный доступ, отчетность об ошибках, метрики. Ротация выполняется в согласованное окно изменений.

SIP/Kamailio. Секреты: SIP-транки, учетные данные PBX, секрет телефонного bridge-сервиса. Использование: телефонные сценарии, интеграция с PBX. Ротация согласуется с владельцем PBX и проверкой входящих/исходящих звонков.

Секреты Искры запрещено размещать в сборочных артефактах Web, Mobile и Desktop, клиентских конфигурациях и пользовательских журналах. Значения секретов в диагностических пакетах, выгрузках поддержки и экспортах инцидента маскируются до передачи за пределы доверенного контура заказчика.

15.7 Контроли безопасности и источники расследования

Этот раздел не описывает единый журнал аудита. Он фиксирует контроли безопасности Искры и источники данных, которые используются при эксплуатации текущей реализации.

Вход, выход, обновление токена, регистрация устройства. Контроль: проверка идентичности, срок действия сессии и отзыв устройства. Источники расследования: таблицы `sessions`, `session_refresh_token_grace`, `devices`, прикладные журналы Gateway и журналы балансировщика.

Изменение пространства, роли, очереди или участника. Контроль: ролевая модель доступа. Источники расследования: состояние PostgreSQL по пользователям, пространствам, ролям и очередям; прикладные журналы Gateway; журнал изменений БД, если он включен инфраструктурой заказчика.

Создание или изменение канала. Контроль: административное право и политика секретов коннектора. Источники расследования: конфигурация канала в PostgreSQL, прикладные журналы Gateway и журналы соответствующего bridge-сервиса.

Входящий вебхук Telegram/MAX/Webchat. Контроль: проверка общего секрета и источника, ключ идемпотентности. Источники расследования: журналы bridge-сервисов, прикладные журналы Gateway, журналы балансировщика; для Webchat также используется `webchat_audit_events`.

Создание, включение, выключение ИИ-агента. Контроль: административное право и жизненный цикл учетных данных бота. Источники расследования: конфигурация ИИ-агента в PostgreSQL, прикладные журналы Gateway и Assistant.

Выдача или отзыв учетных данных бота. Контроль: минимально необходимые права и отзыв учетных данных. Источники расследования: серверная конфигурация учетных данных, прикладные журналы Gateway и состояние соответствующего агента.

Запуск, завершение или ошибка потока агента. Контроль: аренда обработки одного активного исполнителя, политика повтора и передачи оператору. Источники расследования: аренды обработки и состояние диалога в PostgreSQL, журналы Assistant, события реального времени и журналы очередей.

Запрос инструмента и согласование. Контроль: подтверждение человеком для действий с бизнес-риском. Источники расследования: состояние диалога, сообщения и служебные записи в PostgreSQL, журналы Assistant и Gateway.

Передача оператору от ИИ-агента. Контроль: право на очередь и владение диалогом. Источники расследования: состояние очереди, назначения оператора и диалога в PostgreSQL, прикладные журналы Gateway и Assistant.

Выдача токена медиа-комнаты. Контроль: авторизация Gateway перед доступом к LiveKit. Источники расследования: прикладные журналы Gateway, журналы LiveKit и данные звонка в PostgreSQL при наличии соответствующей записи.

Загрузка и выгрузка вложения. Контроль: политика доступа к объектам, ограничения размера, политика проверки вредоносного содержимого при включении. Источники расследования: метаданные вложения в PostgreSQL, журналы Gateway, журналы объектного хранилища и события Webchat в `webchat_audit_events`, если вложение связано с Webchat.

Административное изменение конфигурации. Контроль: административное право и фиксация текущего состояния конфигурации. Источники расследования: актуальная конфигурация в PostgreSQL, прикладные журналы Gateway и журнал изменений БД, если он включен инфраструктурой заказчика.

Отказ доступа или нарушение политики. Контроль: мониторинг безопасности и корреляция оповещений. Источники расследования: прикладные журналы Gateway и bridge-сервисов, журналы балансировщика, журналы инфраструктуры заказчика и метрики мониторинга.

Сырые токены, секреты ботов, ключи LLM, SMS-секреты, закрытые ключи и полные учетные данные поставщика не должны попадать в прикладные журналы, диагностические пакеты и выгрузки инцидента. При необходимости единого юридически значимого аудита он оформляется как отдельная возможность поставки и должен быть явно указан в профиле внедрения.

16. Надежность, HA и DR

Система рассчитана на то, что отдельные контуры могут перезапускаться или временно быть недоступными.

Ситуация	Ожидаемое поведение
WebSocket клиента оборвался	Клиент переподключается и перечитывает состояние
Среда выполнения агента перезапустилась	Получает новую аренду обработки и продолжает слушать события
Поставщик LLM недоступен	Поток помечается ошибочным, возможен повтор или передача оператору
Медиа-контур недоступен	Сообщения продолжают работать, звонки диагностируются отдельно
Объектное хранилище недоступно	Текстовые сообщения работают, вложения требуют восстановления доступа
Внешний канал прислал повтор	Коннектор поддерживает дедупликацию и идемпотентность

16.1 Цели восстановления

Для промышленного внедрения Искры до запуска пользователей оформляется и утверждается раздел **'Цели восстановления'** в профиле внедрения. Этот раздел является обязательным: без утвержденных RPO/RTO эксплуатационная готовность Искры не считается подтвержденной.

Поставщик Искры передает заказчику перечень контуров, для которых требуются цели восстановления, и технические зависимости каждого контура. Архитектор заказчика совместно с владельцем эксплуатации и службой информационной безопасности указывает целевые значения, проверяет их соответствие внутренним требованиям и утверждает профиль внедрения. Команда эксплуатации заказчика реализует резервное копирование, репликацию, мониторинг и процедуры восстановления в соответствии с утвержденными значениями.

В разделе "Цели восстановления" обязательно фиксируются:

- RPO для PostgreSQL;
- RPO для объектного хранилища;
- допустимая потеря временных событий реального времени;
- RTO для Gateway/API;
- RTO для контура реального времени;
- RTO для медиа-контура;
- RTO для внешних каналов;
- порядок восстановления после ошибочной миграции;
- ответственный владелец восстановления со стороны заказчика;
- дата последней проверки восстановления и ссылка на протокол проверки.

16.2 Резервное копирование и восстановление

Политика резервного копирования должна покрывать:

- глобальные объекты PostgreSQL, схему и данные;
- выделенные базы вспомогательных сервисов для включенных опциональных модулей;
- объектное хранилище с вложениями, записями и артефактами;
- конфигурацию публичных точек доступа, LiveKit, SIP и коннекторов;
- секреты или процедуру их восстановления через утвержденное хранилище секретов.

Резервное копирование без проверки восстановления не считается доказанным механизмом восстановления. Для промышленного внедрения требуется регулярная проверка восстановления в изолированном контуре.

16.3 Домены отказа

Gateway, Web, Webchat, Assistant, bridge-сервисы и процессы фоновой обработки являются компонентами без локального состояния или с минимальным локальным состоянием. Их пересоздание допускается при доступности PostgreSQL, Redis/NATS и объектного хранилища.

PostgreSQL, объектное хранилище и слой событий являются доменами отказа с состоянием. Их отказ влияет на корректность данных и требует отдельной стратегии: репликации, резервного копирования, восстановления, мониторинга, планирования емкости и процедур переключения при отказе.

16.4 Матрица данных и порядок восстановления

Бизнес-состояние. Хранилище: PostgreSQL. Содержит пользователей, сессии, устройства, пространства, роли, очереди, диалоги, сообщения, доставку, состояние непрочитанных сообщений, конфигурации ИИ-агентов, аренды обработки и журналы событий реализованных подсистем. Требование: обязательное резервное копирование глобальных объектов, схемы и данных; восстановление проверяется регулярной тренировкой.

Вложения и медиа-артефакты. Хранилище: объектное хранилище / S3 / MinIO. Содержит вложения сообщений, записи, импорт, экспорт, OTA-файлы и файлы загрузки. Требование: резервное копирование или репликация по политике заказчика; жизненный цикл защищает от случайного удаления.

События реального времени и очередей. Хранилище: NATS. Содержит распространение событий, события фоновых обработчиков и доставку потребителям. Требование: срок хранения и состояние потребителей мониторятся; критичное бизнес-состояние восстанавливается из PostgreSQL.

Состояние среды выполнения. Хранилище: Redis. Содержит присутствие, лимиты частоты запросов, кэши и выбранные данные координации медиа. Требование: восстанавливается из текущего состояния системы; политика отказоустойчивости фиксируется для профиля внедрения.

Медиа-сессии. Хранилище: LiveKit. Содержит активные комнаты, участников и состояние медиа. Требование: активные звонки могут быть потеряны при отказе медиа-контура; записи сохраняются через выгрузку в объектное хранилище.

Состояние исполнения агента. Хранилище: PostgreSQL, аренды обработки Gateway, события NATS. Содержит статус потока агента, запросы на действия, передачу оператору и аренду обработки одного активного исполнителя. Требование: после восстановления среда выполнения агентов перечитывает состояние через Gateway и не должна создавать дублирующий ответ.

Курсор и идемпотентность внешнего канала. Хранилище: PostgreSQL. Содержит конфигурацию канала, внешние идентификаторы событий и состояние доставки. Требование: после восстановления bridge-сервисы используют ключи идемпотентности и состояние поставщика для безопасного продолжения.

Секреты и конфигурация точек доступа. Хранилище: хранилище секретов / конфигурация развертывания. Содержит учетные данные, публичные URL, внутренние URL и лимиты поставщика. Требование: восстанавливается до запуска приложений; значения секретов не берутся из резервных копий журналов.

Мониторинг и отчетность об ошибках. Хранилище: Prometheus/Grafana/GlitchTip или инструменты заказчика. Содержит метрики, панели мониторинга, оповещения и события ошибок. Требование: не является источником бизнес-состояния; восстанавливается для операционной видимости.

Поддерживаемый порядок восстановления Искры:

1. Остановить публичный трафик или перевести входной контур в режим обслуживания.

2. Восстановить конфигурацию развертывания: DNS, сертификаты, внутренние URL сервисов, публичные URL, функциональные флаги и ссылки на секреты.
3. Восстановить секреты из утвержденного хранилища секретов.
4. Восстановить глобальные объекты PostgreSQL, схему базы данных и данные на версию, совместимую с выбранным релизом Искры.
5. Восстановить бакеты объектного хранилища с вложениями, записями, импортом, экспортом, OTA-файлами и файлами загрузки.
6. Запустить PostgreSQL, объектное хранилище, Redis и NATS; проверить готовность и базовую задержку.
7. Выполнить миграции только если точка восстановления и план релиза требуют прямой миграции; откат после миграции выполняется через восстановление, а не через ручное изменение данных.
8. Запустить Gateway и Web, проверить `/healthz`, вход пользователя, чтение пространств, диалогов и сообщений.
9. Запустить Webchat, Assistant, Event-organizer, bridge-сервисы Telegram/MAX, фоновые обработчики расшифровки и нормализации стикеров.
10. Запустить LiveKit, контур выгрузки и SIP/Kamailio при включенном медиа- или телефонном контуре.
11. Выполнить дымовые проверки: вход пользователя, отправка сообщения, событие реального времени, загрузка/выгрузка вложения, обращение Webchat, передача оператору от ИИ-агента, выдача медиа-токена, внешний вебхук канала.
12. Включить публичный трафик и продолжить мониторинг по доле ошибок, задержке очередей, дублям внешних событий, ошибочным потокам ИИ и качеству медиа.

17. Горизонтальное масштабирование

Искра масштабируется не как один большой сервер, а как набор контуров с разной природой нагрузки. Сервисы без локального состояния поддерживают горизонтальное масштабирование экземплярами; состояние, очереди, аренда обработки и распространение событий должны оставаться общими и наблюдаемыми.

17.1 Gateway

Gateway поддерживает горизонтальное масштабирование несколькими экземплярами при условии, что все экземпляры используют общие PostgreSQL, Redis/NATS и объектное хранилище. Экземпляр Gateway не должен быть единственным владельцем бизнес-состояния: пользователи, сессии, устройства, права, диалоги, сообщения, доставка, аренда обработки и реализованные журналы событий хранятся в общем слое состояния.

Для HTTP-запросов используется балансировка на уровне входного контура или балансировщика нагрузки. Для WebSocket-соединений каждый экземпляр Gateway должен получать события из общего контура реального времени/распространения событий и доставлять клиенту актуальное состояние. При обрыве соединения клиент перечитывает снимок состояния через API и продолжает работу с последней сохраненной позиции.

Практически это означает:

- новые экземпляры Gateway добавляют пропускную способность API и WebSocket;
- база данных и пул соединений должны масштабироваться вместе с Gateway;
- события реального времени не должны жить только в памяти одного экземпляра;
- идемпотентность операций важнее привязки пользователя к конкретному экземпляру;
- привязка сессии к экземпляру допустима как оптимизация, но не должна быть обязательным условием корректности.

17.2 Реальное время и доставка событий

Контур реального времени должен выдерживать распространение событий на Web, Desktop, Mobile, Webchat и среду выполнения агента. Масштабирование строится вокруг общей шины событий и понятного правила восстановления: если событие потеряно на соединении, клиент перечитывает состояние из Gateway.

Критичные метрики:

- количество активных WebSocket-соединений на экземпляр;
- задержка доставки события до клиента;
- размер очередей и задержка потребителей;
- частота переподключений;
- доля клиентов, которым приходится перечитывать снимок состояния.

17.3 ИИ Агенты

ИИ-агенты масштабируются через фоновые обработчики среды выполнения агентов. Несколько процессов среды выполнения могут обслуживать разные пространства, очереди или агентов, но один конкретный агент в конкретном контексте не должен отвечать дважды.

Масштабирование среды выполнения агента должно сохранять три правила:

- одна активная аренда обработки на обработку агентского потока;
- повторный запуск среды выполнения не создает дубль ответа;
- все сообщения, запросы на действие, состояние потоков и передача оператору фиксируются через Gateway.

При росте нагрузки добавляются фоновые обработчики среды выполнения, увеличиваются лимиты поставщиков LLM и отдельно контролируются очереди событий агентов. Если поставщик LLM замедляется, система должна деградировать через ошибочный поток, политику повтора или передачу оператору, а не блокировать основной контур сообщений.

17.4 Фоновые обработчики

Расшифровка, нормализация стикеров, обратные вызовы выгрузки, OTA/биллинга и другие фоновые задачи масштабируются как пулы фоновых обработчиков. Количество фоновых обработчиков определяется длиной очереди, временем обработки и долей ошибок.

Для таких фоновых обработчиков важны:

- идемпотентная обработка повторных задач;

- ограничение параллелизма для внешних поставщиков;
- отдельные правила повтора и dead-letter;
- отсутствие локального состояния, которое теряется при перезапуске;
- сохранение результата в PostgreSQL или объектное хранилище.

17.5 Внешние каналы

Bridge-сервисы Telegram, MAX, Webchat, телефонии и других каналов поддерживают горизонтальное масштабирование при обязательной дедупликации входящих событий. Слой коннекторов должен рассматривать повторный вебхук от поставщика как штатный сценарий и использовать внешний идентификатор события, идентификатор канала или другой стабильный ключ идемпотентности.

17.6 Медиа-контур

Звонки масштабируются отдельно от сообщений. Узлы LiveKit/медиа увеличиваются по числу одновременных комнат, участников, битрейту, записям и SIP-сессиям. Gateway остается управляющим контуром: он выдает права и токены, но не передает через себя аудио/видео поток.

Для медиа-контура особенно важны:

- публичная доступность нужных TCP/UDP портов;
- корректная маршрутизация клиентов к узлам медиа;
- отдельный мониторинг потерь пакетов, jitter, битрейта и переподключений;
- ресурсы для выгрузки/записи, потому что запись потребляет CPU, память и пропускную способность хранилища;
- отдельная диагностика SIP/Kamailio для опционального модуля телефонии.

17.7 Слой данных

Горизонтальное масштабирование сервисов упирается в общий слой состояния. PostgreSQL хранит источник истины, Redis/NATS поддерживают состояние среды выполнения и события, объектное хранилище хранит вложения и записи. Эти компоненты масштабируются собственной эксплуатационной моделью: репликацией, резервным копированием, лимитами соединений, мониторингом, задержкой и сроком хранения.

Обязательные правила для роста нагрузки:

- вынести пул соединений перед PostgreSQL;
- разделять нагрузки чтения и записи только там, где это не нарушает консистентность продукта;
- контролировать размер таблиц сообщений, событий, журналов подсистем и вложений;
- держать объектное хранилище внешним общим ресурсом для всех экземпляров;
- не использовать локальный диск приложения как источник истины;
- фиксировать срок хранения для событий реального времени, журналов, записей и временных артефактов в профиле внедрения.

17.8 Порядок масштабирования

Поддерживаемая последовательность горизонтального масштабирования для смешанной нагрузки:

1. Емкость Gateway и WebSocket.
2. Соединения PostgreSQL, индексы и пул соединений.
3. Контур реального времени/распространения событий и очереди.
4. Фоновые обработчики среды выполнения агентов и лимиты LLM.
5. Узлы медиа, выгрузка и SIP-контур.
6. Пропускная способность объектного хранилища и политики жизненного цикла.
7. Bridge-сервисы внешних каналов.

Профиль внедрения задает порядок горизонтального масштабирования под профиль нагрузки: при доминирующей медиа-нагрузке первым масштабируется медиа-контур; при доминирующих ИИ-сценариях первыми масштабируются среда выполнения агента и LLM-интеграции. Каждый контур масштабируется по своей продуктовой метрике, а не только по CPU.

18. Расчет емкости и планирование емкости

Расчет емкости выполняется как часть профиля внедрения. Для Искры фиксируются не только CPU и память, но и количество соединений, скорость сообщений, распространение событий, битрейт медиа, объем вложений и задержки ИИ.

18.1 Входные параметры расчет емкости

Профиль внедрения содержит:

- число зарегистрированных пользователей;
- число активных пользователей в рабочий день;
- пиковое число одновременных WebSocket-соединений;
- сообщений в секунду и сообщений в день;
- число пространств, очередей и операторов;
- число внешних каналов и вебхуков в минуту;
- число ИИ-агентов и одновременных диалогов агентов;
- среднюю и пиковую длительность LLM-запросов;
- число одновременных звонков и участников;
- долю звонков с записью;
- средний размер вложений и дневной прирост объектного хранилища;
- срок хранения сообщений, вложений, записей, журналов подсистем и прикладных журналов.

18.2 Что масштабировать по какой метрике

Контур	Основная метрика	Что увеличивать
Gateway/API	RPS, p95/p99 задержка, доля 5xx	Экземпляры Gateway, пул соединений, индексы
Реальное время	Активные соединения, задержка распространения событий, частота переподключений	Емкость Gateway, ресурсы NATS/Redis
PostgreSQL	Соединения, задержка запросов, блокировки, рост таблиц	Пул соединений, индексы, ресурсы БД, реплики чтения при сохранении консистентности
ИИ	Задержка очереди, задержка LLM, ошибочные потоки, доля передач оператору	Фоновые обработчики среды выполнения, лимиты поставщика, емкость точки доступа модели
Медиа	Комнаты, участники, битрейт, потери пакетов, записи	Узлы LiveKit, фоновые обработчики выгрузки, пропускная способность сети
Фоновые обработчики	Длина очереди, длительность задания, доля повторов	Экземпляры обработчиков, лимиты поставщика
Объектное хранилище	Дневной прирост, пропускная способность чтения/записи, ошибочные загрузки	Емкость хранилища, пропускная способность, политики жизненного цикла

Результат расчета емкости включается в профиль внедрения Искры и привязывается к одному из поддерживаемых профилей: Small, Standard, High Availability или Индивидуальный корпоративный. Для выбранного профиля заказчик утверждает ресурсы, лимиты, репликацию, политики резервного копирования и ожидаемые SLO; поставщик Искры проверяет, что выбранные значения соответствуют поддерживаемой архитектуре продукта.

18.3 Профиль внедрения Искры

Профиль внедрения Искры - обязательное приложение к архитектурному решению локального внедрения. Это не справочный раздел и не набор рекомендаций. Профиль внедрения должен быть заполнен для конкретного заказчика, согласован до промышленного запуска и использоваться как основание для проектирования инфраструктуры, приемки эксплуатации и последующих изменений.

Поставщик Искры отвечает за продуктовую часть профиля: перечень поддерживаемых контуров, обязательные зависимости, допустимые варианты масштабирования, требования к состоянию, API, событиям, медиа, агентам и внешним каналам. Заказчик отвечает за инфраструктурную и эксплуатационную часть: сетевые зоны, вычислительные ресурсы, БД, объектное хранилище, секреты, резервное копирование, мониторинг, SIEM, IdP, RPO/RTO, сроки хранения и внутренние владельцы сервисов.

Профиль внедрения должен содержать:

- выбранный профиль: Small, Standard, High Availability или Индивидуальный корпоративный;
- список включенных модулей Искры и отключенных модулей;
- целевую сетевую топологию и опубликованные публичные точки доступа;
- модель состояния: PostgreSQL, Redis, NATS, объектное хранилище, LiveKit и опциональные компоненты;
- правила горизонтального масштабирования по контурам;
- SLO, RPO/RTO, сроки хранения и требования к проверке восстановления;
- перечень внешних поставщиков и разрешенных исходящих потоков данных;
- матрицу ответственности поставщика Искры и заказчика;
- лиц, утвердивших профиль со стороны архитектуры, эксплуатации и информационной безопасности заказчика.

Изменение профиля внедрения после промышленного запуска выполняется как архитектурное изменение: фиксируется новая версия профиля, проверяется влияние на масштабирование, безопасность, RPO/RTO и интеграции, затем изменение утверждается владельцами заказчика.

Small. Назначение: пилот, промышленная эксплуатация подразделения, ограниченный поток обращений. Топология: по одному экземпляру Gateway, Web, Webchat, Assistant и bridge-сервисов/сервисов фоновой обработки; медиа и ИИ включаются по необходимости. Модель состояния: PostgreSQL, Redis, NATS и объектное хранилище могут быть отдельными управляемыми или локально установленными сервисами с обязательным резервным копированием. Логика масштабирования: рост начинается с Gateway/WebSocket и пула PostgreSQL; отказоустойчивость не заявляется без отдельного профиля.

Standard. Назначение: базовая корпоративная промышленная эксплуатация. Топология: 2+ экземпляра Gateway/Web/Webchat за балансировщиком нагрузки; отдельные пулы фоновых обработчиков; Assistant и bridge-сервисы масштабируются по нагрузке. Модель состояния: выделенный PostgreSQL, общее объектное хранилище, контролируемые Redis/NATS, регулярная проверка восстановления. Логика масштабирования: добавляются экземпляры сервисов без локального состояния; отдельно контролируются соединения БД, задержка NATS, задержка очереди ИИ и пропускная способность объектного хранилища.

High Availability. Назначение: промышленная эксплуатация с требованием отказоустойчивости прикладного слоя. Топология: 2+ экземпляра всех критичных компонентов без локального состояния; узлы медиа и емкость выгрузки отделены от контура сообщений. Модель состояния: отказоустойчивые PostgreSQL/объектное хранилище/Redis/NATS по стандарту заказчика; резервное копирование, переключение при отказе и проверка восстановления обязательны. Логика масштабирования: все экземпляры работают без постоянного локального состояния; отказ одного прикладного узла не останавливает сообщения, реальное время, Webchat и пулы фоновых обработчиков ИИ.

Индивидуальный корпоративный. Назначение: высокая нагрузка, несколько сетевых зон, строгие требования соответствия, SIEM и IdP. Топология: зоны DMZ, приложений, данных, медиа и эксплуатации; отдельные bridge-сервисы каналов, фоновые обработчики ИИ, пулы емкости медиа/SIP. Модель состояния: слой данных проектируется вместе с заказчиком: репликация, срок

хранения, шифрование, экспорт доступных журналов подсистем, DR-площадка или согласованная альтернатива. Логика масштабирования: каждый контур масштабируется по своей метрике: RPS, активные соединения, сообщений в день, диалоги агентов, битрейт медиа, частота вебхуков, рост хранилища.

Профиль внедрения должен явно указывать, какие модули Искры включены: сообщения, Webchat, Telegram/MAX, ИИ-агенты, медиазвонки, запись/выгрузка, расшифровка, SIP/Kamailio, мониторинг/отчетность об ошибках и корпоративный SSO. Невключенный модуль не должен оставлять опубликованные публичные точки доступа или активные учетные данные поставщика.

19. Наблюдаемость и эксплуатация

Промышленная эксплуатация должна иметь видимость по всем критичным контурам.

19.1 Проверки работоспособности

Минимальный набор проверки работоспособности:

- Gateway `/healthz`;
- Assistant `/healthz`;
- Event-organizer `/healthz`;
- Webchat `/healthz`;
- bridge-сервисы `/healthz`;
- фоновый обработчик расшифровки `/healthz`;
- нормализатор стикеров `/healthz`;
- PostgreSQL готовность;
- Redis готовность;
- NATS работоспособность;
- объектное хранилище готовность;
- готовность сигнализации и медиа LiveKit.

19.2 Метрики

Минимальный набор метрик:

- HTTP RPS, задержка, класс статуса по Gateway;
- активные соединения WebSocket, переподключения, задержка распространения событий;
- соединения PostgreSQL, блокировки, медленные запросы, рост таблиц;
- память Redis, вытеснения, задержка;
- задержка потребителей NATS, ожидающие сообщения, переподключения;
- потоки ИИ, ошибочные потоки, задержка LLM, запросы на действия, доля передач оператору;
- длина очереди фоновой обработки, длительность задания, повторы и dead-letter события;

- комнаты LiveKit, участники, потери пакетов, jitter, битрейт, задания выгрузки;
- ошибки загрузки/выгрузки и емкость объектного хранилища;
- ошибки вебхуков внешних каналов и доля дублей событий.

19.3 Эксплуатационные журналы

Прикладные журналы должны помогать расследовать инциденты, но не должны содержать секреты, токены ботов, ключи LLM, SMS-секреты, токены доступа и персональные данные сверх необходимого минимума.

В текущей реализации для диагностики используются прикладные журналы Gateway, Assistant, bridge-сервисов, Webchat service, Event-organizer, LiveKit, фоновых обработчиков и инфраструктурных компонентов заказчика. Отдельные журналы событий подсистем доступны в `webchat_audit_events` и `event_audit_log`; их покрытие описано в разделе 15.5.

Экспорт журналов в систему централизованного сбора журналов заказчика настраивается как эксплуатационная интеграция. Единый экспорт юридически значимых событий не относится к базовой текущей реализации и должен быть отдельно указан в профиле внедрения, если он входит в конкретную поставку.

19.4 Оповещения

Оповещения должны покрывать:

- недоступность публичных точек доступа;
- рост 5xx и p95/p99 задержки;
- исчерпание соединений БД;
- рост задержки NATS и фоновых обработчиков;
- массовые WebSocket переподключения;
- рост ошибочных потоков ИИ;
- отказ поставщика LLM или превышение лимитов поставщика;
- отказ объектного хранилища;
- деградацию качества сети LiveKit;
- ошибки вебхуков внешних каналов;
- неуспешные задачи резервного копирования или устаревшие резервные копии.

20. Данные, срок хранения и конфиденциальность

Профиль внедрения должен содержать политику хранения:

- сообщения и системные события;
- вложения и медиа-файлы;
- записи звонков;
- транскрипции и сводки;

- prompts, контекст и служебные заметки ИИ-агентов;
- журналы подсистем и прикладные журналы;
- прикладные журналы;
- события отчетности об ошибках;
- временные артефакты, OTA-файлы и файлы загрузки.

Для каждого класса данных фиксируются:

- владелец данных;
- место хранения;
- срок хранения;
- правила удаления;
- правила экспорта;
- доступ администраторов;
- требования к резервному копированию;
- требования к обезличиванию или минимизации.

Для внешних поставщиков ИИ, SMS, push-уведомлений, биллинга и каналов профиль внедрения содержит исходящий поток данных: какие данные уходят, кому, для какой цели и на каком основании это разрешено политикой заказчика.

21. Инфраструктурные требования

21.1 Сеть

Промышленное внедрение должно предоставлять:

- публичный HTTPS входной контур;
- поддержку upgrade для WebSocket;
- внутреннюю сеть между сервисами;
- доступ Gateway к PostgreSQL, Redis, NATS, объектному хранилищу, Assistant, фоновым обработчикам и bridge-сервисам;
- исходящий доступ к поставщикам включенных внешних интеграций;
- LiveKit TCP/UDP маршрутизация;
- SIP TCP/UDP маршрутизация для опционального модуля телефонии;
- правила межсетевого экрана между зонами DMZ, приложений, данных и эксплуатации.

21.2 Хранилище

PostgreSQL требует надежного постоянного хранилища с мониторингом задержки, IOPS, свободного места и резервного копирования. Объектное хранилище должно быть общим для всех прикладных экземпляров. Запись медиа и крупный импорт требуют отдельного планирования емкости по пропускной способности и объему.

21.3 Секреты и конфигурация

Конфигурация должна разделять:

- публичные URL;
- внутренние URL сервисов;
- учетные данные;
- функциональные флаги;
- лимиты поставщика;
- параметры хранения;
- SLO и пороги оповещений.

Секреты не должны храниться в клиентских приложениях или попадать в сборочные артефакты.

21.4 Версии и совместимость

Профиль внедрения фиксирует базовую линию версий для PostgreSQL, Redis, NATS, объектного хранилища, LiveKit, браузеров/сред выполнения, Mobile- и Desktop-клиентов. Обновления компонентов с состоянием выполняются по отдельному плану изменения с процедурами отката и восстановления.

22. Модель внедрения

Архитектура Искры поддерживает поэтапное включение контуров: сначала запускается базовый контур сообщений, затем подключаются ИИ-агенты, медиа, внешние каналы, телефония и дополнительные процессы фоновой обработки.

Поддерживаемая последовательность внедрения:

1. **Базовые сообщения**: Gateway, Web, PostgreSQL, Redis/NATS, объектное хранилище, базовые роли и пространства.
2. **Реальное время и клиенты**: WebSocket, Desktop/Mobile, push-уведомления.
3. **Внешние каналы**: webchat, Telegram/MAX или другие bridge-сервисы.
4. **ИИ-агенты**: Assistant, среда выполнения агента, поставщик LLM, согласование действий инструментов, передача оператору.
5. **Медиа**: LiveKit, звонки, выгрузка/запись, расшифровка.
6. **Телефония**: SIP/Kamailio/PBX как опциональный модуль.
7. **Корпоративное усиление защиты**: SIEM, проверки восстановления, DR-тест, проверка безопасности, приемка SLO.

Каждый новый контур добавляется без обхода единой модели прав, состояния и реализованных журналов событий.

23. Граница ответственности

Область	Поставщик Искры	Заказчик / эксплуатация
Архитектура продукта	Описывает компоненты, контуры, интеграции и требования	Проверяет соответствие внутренним стандартам
Прикладной код	Поставляет Gateway, клиенты, фоновые обработчики, bridge-сервисы, Assistant	Разворачивает в утвержденной инфраструктуре
Инфраструктура	Формулирует требования и эталонную топологию	Предоставляет вычислительные ресурсы, сеть, хранилище, DNS, сертификаты
Секреты	Описывает необходимые типы секретов	Хранит, ротирует и контролирует доступ
База данных	Описывает требования к схемам и миграциям	Обеспечивает резервное копирование, восстановление, мониторинг, емкость
Объектное хранилище	Описывает типы объектов и требования доступа	Обеспечивает надежность хранения, жизненный цикл, резервное копирование/репликацию
Мониторинг	Поставляет точки доступа и набор ключевых метрик	Настраивает сбор, панели мониторинга, оповещения, SIEM
Проверка безопасности	Предоставляет архитектурные данные и рекомендации по усилению защиты	Проводит согласование, тестирование на проникновение, применение политик
DR	Описывает зависимые контуры и предпосылки восстановления	Определяет RPO/RTO, проводит тренировки восстановления

Настоящая спецификация фиксирует базовую техническую границу ответственности. Договор поставки может расширять ее, но не отменяет эти обязанности без явного приложения к проекту внедрения.

24. Контрольный список готовности к промышленной эксплуатации

- Определена целевая топология с зонами DMZ, приложений, данных, медиа и эксплуатации.
- Публичные точки доступа опубликованы только там, где это требуется.
- TLS, сертификаты и DNS настроены и принадлежат согласованной стороне.

- Пользователи входят и получают правильные роли.
- Пространства, очереди и каналы настроены.
- Web, desktop и mobile клиенты видят одно состояние.
- Webchat или внешний канал создает обращение в нужной очереди.
- ИИ-агент отвечает, не дублирует ответы и умеет передавать диалог оператору.
- Действия инструментов имеют подтверждение там, где есть бизнес-риск.
- Звонок создается, участники подключаются, медиа работает.
- Вложения загружаются, открываются и хранятся в общем объектном хранилище.
- Push/SMS/OTP и внешние обратные вызовы защищены секретами.
- Логи не содержат токен бота, SMS секреты, ключи LLM и персональные данные сверх необходимости.
- Есть резервное копирование базы, объектного хранилища и конфигурации.
- Тренировка восстановления выполнена и задокументирована.
- Настроена наблюдаемость для Gateway, Assistant, медиа, хранилища, реального времени и фоновых обработчиков.
- Настроены оповещения по доступности, задержке, 5xx, соединениям БД, очередям, ошибкам ИИ, качеству медиа и возрасту резервных копий.
- Зафиксированы RPO/RTO, срок хранения и матрица ответственности.
- Проверка безопасности завершена до промышленного доступа пользователей.