



OPERATOR / USER DOCUMENTATION

Information item: **User / Operator Documentation**. Structure and content are informed by **ISO/IEC 26514** and **ISO/IEC/IEEE 15289**. Life-cycle placement draws on operation and maintenance processes in **ISO/IEC 12207**. Security guidance is informed by **ISO/IEC 27001/27002**, **ISO/IEC 29100**, and potentially applicable Russian Federation law (152-FZ, 149-FZ). This is not certification or legal advice. Print layout targets A4.

на программное обеспечение

«CoronerChat»

Версия изделия: **2.7.10**

Аннотация

К программе (изделию)

CoronerChat 2.7.10 — прикладное ПО для стримеров и модераторов: единая лента Twitch, VK Video Live, Kick, YouTube Live, Rutube, DonationAlerts и MemeAlerts; веб-UI и OBS overlay; UX-пакет эфира; статистика и достижения; локализация ru/en/ro/de/fi; автообновление с GitHub Latest; desktop Electron (Windows). Настоящее руководство описывает установку, настройку, штатную эксплуатацию и первичную диагностику без доступа к исходному коду. Расширенный web User Guide: <docs/guide.html>.

К настоящему документу

Документ содержит требования к среде исполнения, порядок установки и запуска, описание операций по платформам, **полный перечень функций изделия, HTTP API и сохраняемых настроек интерфейса** (п. 6), типовые неисправности (п. 7), **информационную безопасность: международные стандарты и законодательство РФ** (п. 8), обслуживание и резервное копирование (п. 9). Архитектура и протоколы — в Software Architecture Description (<docs/coronerchat-architecture-print.html>). Публичная web-версия руководства: <docs/guide.html>.

Ключевые слова: user documentation; operator guide; CoronerChat; information security; personal data; 152-FZ; GDPR; ISO/IEC 27001; OAuth 2.0; Electron; OBS; Twitch; VK Video Live; Kick; YouTube Live; DonationAlerts; MemeAlerts.

Нормативные и справочные ссылки

- ISO/IEC 26514 — Requirements for designers and developers of user documentation.
- ISO/IEC/IEEE 15289 — Content of life-cycle information items.
- ISO/IEC 12207 — Software life cycle processes (operation, maintenance).
- ISO/IEC 27001 / 27002 — Information security management systems / controls.
- ISO/IEC 29100 — Privacy framework.
- GDPR (EU) Regulation (EU) 2016/679 — insofar as processing concerns EU data subjects.
- NIST Cybersecurity Framework; OWASP ASVS — application security reference.
- Федеральный закон от 27.07.2006 № 152-ФЗ «О персональных данных».
- Федеральный закон от 27.07.2006 № 149-ФЗ «Об информации, информационных технологиях и о защите информации».
- Постановление Правительства РФ от 01.11.2012 № 1119 — требования к защите ПДн в ИСПДн (корпоративный контур).
- Приказ ФСТЭК России от 11.02.2013 № 17 — модель угроз (справочно).
- ГОСТ Р ИСО/МЭК 27001 — национальная гармонизация ISO/IEC 27001 (рекомендательно).
- Связанный документ: Software Architecture Description — <docs/coronerchat-architecture-print.html>.

Термины и сокращения

Термин	Определение
Оператор	Пользователь изделия, выполняющий установку, настройку и эксплуатацию CoronerChat на своём ПК.
Изделие	Программное обеспечение CoronerChat в поставляемой сборке (desktop или web).
Оверлей	Режим веб-интерфейса для вывода чата в OBS Browser Source без панелей управления.
OAuth 2.0	Протокол авторизации на стороне платформ (вход по кнопкам в интерфейсе изделия).
ПДн	Персональные данные — любая информация, относящаяся к прямо или косвенно определённому физическому лицу (в т.ч. ник и сообщения в чате при отсутствии анонимизации).
ИСПДн	Информационная система персональных данных — совокупность ПДн и технических средств их обработки; в корпоративном сценарии CoronerChat может входить в состав ИСПДн.

CoronerChat — руководство оператора (v2.7.10)

Печатная форма A4. Версия изделия: **2.7.10** (`package.json` / GitHub Latest). ISO/IEC 26514.

Содержание

- 1. Введение
- 2. Требования к техническим средствам и программной среде
- 3. Состав поставки, установка и обновление
- 4. Подготовка к работе
- 5. Описание операций (платформы, оверлей, now playing, переменные окружения — п. 5.14–5.15)
- 6. Полный перечень функций изделия, API и настроек интерфейса
- 7. Сообщения оператору и устранение типовых неисправностей
- 8. Информационная безопасность (ISO/IEC · GDPR · законодательство РФ)
- 9. Техническое обслуживание, резервное копирование и контроль работоспособности

1. Введение

1.1. Назначение документа

Настоящее руководство оператора предназначено для обучения и повседневной эксплуатации ПО «CoronerChat» в штатных режимах. Документ не заменяет договор поставки, лицензионные соглашения сторонних сервисов и не описывает внутреннюю реализацию изделия.

1.2. Область применения изделия

Изделие применяется для приёма, нормализации и отображения сообщений чатов стриминговых и смежных сервисов (включая Rutube), локального поиска по буферу, отправки сообщений и модерации (в объёме API платформ), вывода оверлея в OBS, UX-пакета эфира, локальной статистики/достижений, смены языка UI, автообновления с GitHub, опциональной интеграции OBS WebSocket и сценариев «now playing».

2. Требования к техническим средствам и программной среде

- **Desktop (рекомендуемая поставка для Windows):** 64-разрядная ОС Windows 10 или новее; доступ в Интернет для работы с внешними API и чатами; свободный TCP-порт на интерфейсе обратной связи (по умолчанию 3210, настраивается). Отдельная установка Node.js для конечного пользователя не требуется — среда исполнения входит в сборку Electron.
- **Web / разработка из источников:** установленный **Node.js** версии, совместимой с зависимостями проекта (см. `package.json` / документацию Node.js); ОС по усмотрению разработчика; сетевой доступ аналогично desktop.
- **OBS Studio** (при использовании Browser Source): актуальная стабильная сборка; в источнике указывается URL локального сервера изделия (см. п. 5.6).
- Для опционального компонента фильтрации трафика (zapret / winws), вызываемого из desktop-версии: по умолчанию доступны только **пользовательские команды** запуска внешнего скрипта; встроенная загрузка bundle и запуск winws включаются переменной `CORONERCHAT_ZAPRET_BUILTIN=1` (Windows, UAC). Выполнять только при осознанной необходимости и в рамках законодательства.

3. Состав поставки, установка и обновление

3.1. Установщик Windows (NSIS)

Запустите файл вида `CoronerChat Setup <версия>.exe`. Следуйте шагам мастера; при необходимости измените каталог установки. Ярлыки создаются по выбранным в мастере опциям. Деинсталляция — стандартными средствами Windows («Приложения и возможности»).

3.2. Portable-сборка

Файл `CoronerChat <версия>.exe` или каталог `win-unpacked` с `CoronerChat.exe`. Сохраняйте рядом с исполняемым файлом каталог данных `CoronerChat-data` (создаётся автоматически) — в нём хранятся настройки, состояние приложения, журналы и кэш.

3.3. Обновление

Установите новую версию поверх (NSIS) или замените portable-файлы, сохранив каталог `CoronerChat-data`. После обновления при необходимости повторите OAuth на площадках.

Автообновление (2.7.5). Канал зашит: `dorimeryt-alt/CoronerChat`. При запуске изделие запрашивает GitHub **Latest** (не архивные теги 2.25–2.35). Вкладки «Обновления»: проверить / скачать и установить Setup / пропустить версию / напомнить позже. API: `/api/update/check|install|dismiss`. Подробнее: `docs/updates-and-releases.md`, страница сайта `docs/update.html`.

4. Подготовка к работе

1. Убедитесь в наличии сетевого доступа к используемым платформам.
2. Запустите desktop-ярлык или `CoronerChat.exe` / выполните `npm run dev:web` (для режима разработки из источников).
3. Дождитесь загрузки интерфейса. В консоли web-режима отображается фактический локальный URL и порт.
4. Включите нужные источники в настройках и выполните авторизацию для каждой платформы, с которой изделие будет работать под вашей учётной записью (OAuth 2.0 / вход через браузер — по подсказкам интерфейса).
5. Для **Twitch** при использовании отправки сообщений, модерации Helix и slash-команд задайте Client ID в настройках или через переменные окружения (см. `.env.example`, `README.md`) и пройдите авторизацию (в т.ч. Device Flow).
6. Для предварительной настройки OAuth-приложений (разработка, корпоративная поставка) допускается задание идентификаторов и секретов через переменные окружения — полный перечень см. п. 5.14 и файл `src/config.js`.
7. Укажите канал, трансляцию или идентификатор для каждого включённого источника (Twitch, VK Video Live, Kick, YouTube и др.) в соответствии с подсказками интерфейса.

Порт по умолчанию — **3210** (переменные `CORONERCHAT_PORT` или устаревшее имя `CHATSHOW_PORT`). В **web-режиме** при занятости порта изделие может выбрать другой свободный порт автоматически. В **desktop-режиме** используется заданный порт; при повторном запуске ярлыка изделие переводит фокус на уже открытое окно и может подключаться к уже работающему локальному серверу на том же порту. Если порт занят **другой независимой копией** CoronerChat (например, установленная версия и сборка для разработки одновременно), интерфейс может соответствовать не той сборке — закройте лишнюю копию или задайте для второй другой `CORONERCHAT_PORT`. Если порт занят сторонним процессом, освободите порт или смените порт через переменную.

5. Описание операций

5.1. Выбор канала и просмотр чата

Для каждого подключённого источника задайте канал, URL трансляции или иной идентификатор в формате, ожидаемом платформой (подсказки — в интерфейсе изделия). Сообщения с разных платформ сводятся в одну ленту; обновление — по WebSocket. При прокрутке вверх автоскролл приостанавливается до возврата вниз (кнопка в интерфейсе).

5.2. Сводная таблица возможностей по платформам

Ниже — типовой состав функций по внешним источникам; после обновления изделия сверяйте фактические элементы интерфейса и `README.md`.

Источник (в ленте)	Приём сообщений в общую ленту	Учётные данные	Отправка текста в чат платформы из UI	Примечание
Twitch	Да	Настройки Twitch: Client ID/Secret при необходимости; Device Flow; опционально env TWITCH_USERNAME , TWITCH_OAUTH_TOKEN для чтения IRC без Helix	Да	Модерация Helix, slash-команды, смена категории/названия стрима, события чата — п. 5.8
VK Video Live	Да	OAuth VK ID: Client ID/Secret; redirect /api/auth/vk/callback относительно URL открытого UI	Да	Отдельное поле ввода; ответ на сообщение; часть команд «/» — п. 5.9
Kick	Да	OAuth Kick (PKCE): Client ID/Secret; redirect /api/auth/kick/callback ; scope с chat:write	Да	Slug канала; дождаться подключения сокета — п. 5.10
YouTube Live	Да	Google OAuth: в desktop задайте GOOGLE_OAUTH_CLIENT_ID , GOOGLE_OAUTH_CLIENT_SECRET , redirect в консоли Google вида http://127.0.0.1:<порт>/ (GOOGLE_OAUTH_REDIRECT_PORT , по умолчанию 45280); включить YouTube Data API v3	Да	Идентификатор трансляции / ссылка — п. 5.11
Rutube	Да	Cookie-сессия через встроенное окно входа (desktop); чтение чата — без сессии	Да (после входа)	URL эфира rutube.ru/live/... — п. 5.13а
DonationAlerts	Да (донаты и связанные события)	OAuth: Client ID/Secret; redirect /api/auth/donation-alerts/callback	Нет	Поток событий в ленту и центр уведомлений — п. 5.12
MemeAlerts	Да	Токен из ссылки OBS (?obsToken=...) или JWT; без OAuth того же типа, что у DonationAlerts	Нет	Socket.IO; фильтр мин. суммы; фильтр ленты «Донаты» включает DA+MA — п. 5.13

5.3. Общие правила авторизации

Для каждой платформы используйте свою вкладку или блок в настройках. При запросе изделия откройте ссылку в браузере, подтвердите доступ, вернитесь в UI. Redirect URI в кабинете разработчика внешнего сервиса должен совпадать с подсказкой изделия (для локального UI обычно http://127.0.0.1:<порт>CoronerChat>/...). Не показывайте посторонним экраны с кодами входа и секретами.

5.4. Отправка сообщений и модерация

Отправка в чат доступна для **Twitch**, **VK Video Live**, **Kick**, **YouTube Live** и **Rutube** (после cookie-входа). **DonationAlerts** и **MemeAlerts** поставляют в ленту уведомления о донатах, а не двусторонний текстовый чат. Модерация Helix — преимущественно **Twitch**.

5.5. Смена названия и категории трансляции

Для **Twitch** — через интерфейс изделия при наличии прав (Helix). Для остальных платформ используйте кабинеты самих сервисов или возможности API, если они добавлены в вашей версии UI.

5.6. Режим оверлея для OBS

В Browser Source укажите URL вида `http://127.0.0.1:3210/?overlay=1` (подставьте фактический порт). Дополнительно: `?musicOverlay=1`, `?alerts=1`, `?goal=1`, `?vote=1`, `?recap=1` (итоги эфира), `?modview=1` (LAN для модераторов). В UX-пакете есть «Копировать все OBS URL».

5.7. Now playing и медиа-сессия Windows

В настройках укажите HTTP bridge URL опроса «текущего трека» или используйте локальный режим считывания системной медиа-сессии Windows (если доступен в вашей сборке). Формат ответа bridge описан в `README.md` репозитория.

5.8. Twitch — порядок работы

- **Подключение канала:** укажите имя канала (логин) в поле смены канала; для публичного чата допускается работа без полной авторизации (ограниченный функционал).
- **Авторизация Helix:** сохраните Client ID (и при необходимости Secret), нажмите вход, завершите Device Flow в браузере; дождитесь статуса «авторизован».
- **Отправка и slash:** поле ввода Twitch; команды «/» обрабатываются через Helix или нативный IRC по правилам изделия.
- **Модерация:** действия у сообщения при наличии прав модератора/стримера на выбранном канале.
- **Метаданные стрима:** смена названия и категории в блоке настроек Twitch.
- **7TV и эмодзи:** настройки Twitch / общие параметры кэша 7TV — см. `README.md` (переменные `CORONERCHAT_7TV_*` в п. 5.14).

5.9. VK Video Live — порядок работы

- **Приложение VK ID:** создайте приложение типа, допускающего OAuth для Live; в настройках CoronerChat укажите Client ID и Secret, затем запустите авторизацию — redirect должен указывать на тот же хост и порт, с которого открыт UI.
- **Канал / трансляция:** задайте идентификатор канала VK Video или трансляции в соответствии с полями интерфейса; дождитесь статуса подключения чата.
- **Отправка:** используйте поле ввода VK; доступен ответ на сообщение (reply) через UI.
- **Команды:** строки, начинающиеся с «/», могут обрабатываться как slash-команды VK (см. сообщения об ошибках при неподдерживаемой команде).

5.10. Kick — порядок работы

- **Приложение Kick:** зарегистрируйте OAuth-клиент; укажите Client ID и Secret; в настройках Kick на портале разрешите redirect URI вида `http://127.0.0.1:<порт>/api/auth/kick/callback` (или ваш фактический origin).
- **Права:** для отправки сообщений нужны области в т.ч. `chat:write` и успешный обмен кода по PKCE.
- **Канал:** укажите slug канала Kick; дождитесь зелёного статуса сокета чата перед отправкой.
- **Отправка:** отдельное поле ввода Kick; лимиты длины сообщения действуют со стороны API Kick.

5.11. YouTube Live — порядок работы

- **Google Cloud:** OAuth 2.0 Client (тип, допускающий loopback), включённый **YouTube Data API v3**; в `.env` для desktop — `GOOGLE_OAUTH_CLIENT_ID`, `GOOGLE_OAUTH_CLIENT_SECRET`; redirect URI в консоли Google — `http://127.0.0.1:<GOOGLE_OAUTH_REDIRECT_PORT>/` (по умолчанию порт 45280, см. подсказку в UI).
- **Запуск входа:** в desktop изделие откроет браузер и локальный приём redirect; при занятости порта задайте другой `GOOGLE_OAUTH_REDIRECT_PORT` и тот же URI в консоли Google.
- **Трансляция:** укажите идентификатор трансляции YouTube или ссылку в полях вкладки YouTube.
- **Отправка:** поле ввода YouTube после успешной авторизации и подключения live chat.

5.12. DonationAlerts — порядок работы

- **Приложение DonationAlerts:** Client ID и Secret; redirect `/api/auth/donation-alerts/callback` на origin вашего UI.
- **Авторизация:** кнопка входа во вкладке «Донаты»; после успеха события донатов попадают в ленту и в центр событий.
- **Фильтры:** при необходимости настройте отображение и связку с другими источниками во вкладке «Донаты».

5.13. MemeAlerts — порядок работы

- **Токен:** в личном кабинете MemeAlerts откройте «Ссылку для OBS»; скопируйте URL целиком — изделие извлечёт `obsToken` из query-параметра, либо вставьте JWT вручную.
- **Подключение:** кнопка сохранения токена во вкладке «Донаты»; статус опирается на наличие токена (`tokenConfigured`) и состояние сокета.
- **Фильтрация:** минимальная сумма; скрытие строк МА; фильтр ленты «Донаты» показывает события и DonationAlerts, и MemeAlerts.

5.13a. Rutube — порядок работы

- **Чтение чата:** вкладка Rutube → URL эфира вида `rutube.ru/live/video/...` → «Подключить». Сообщения поступают через публичный poll API.
- **Отправка:** «Встроенное окно входа» (desktop) для cookie-сессии; затем статус сессии и поле ввода. API: `POST /api/settings/rutube, /api/auth/rutube/browser|refresh`.

5.13b. Язык интерфейса (i18n)

- **Настройки** → **Общие:** выбор `ru / en / ro / de / fi`.
- Сохранение: `POST /api/settings/locale`, персист `uiLocale`.
- С версии **2.7.5** смена языка не сбрасывает чекбоксы и вложенную разметку контролов.

5.14. Переменные окружения (полный перечень по изделию)

Значения задаются в системной среде процесса или в файле `.env` при запуске из исходников. Имена `CHATSHOW_*` поддерживаются для совместимости рядом с `CORONERCHAT_*`.

Переменная	Назначение
CORONERCHAT_HOST / CHATSHOW_HOST	Интерфейс привязки HTTP-сервера (по умолчанию 127.0.0.1 — только локальная машина; локальный API отдаёт debug-логи, экспорт настроек и управление OBS без отдельной аутентификации, поэтому LAN-доступ не включён по умолчанию. Для доступа с другого устройства в сети явно задайте 0.0.0.0 или конкретный LAN-адрес и по возможности настройте CORONERCHAT_DECK_TOKEN, см. §8.6).
CORONERCHAT_PORT / CHATSHOW_PORT	Порт HTTP UI (по умолчанию 3210).
CORONERCHAT_DEFAULT_CHANNEL	Имя канала Twitch по умолчанию (или см. брендинг).
TWITCH_USERNAME, TWITCH_OAUTH_TOKEN	Резервная пара для чтения Twitch IRC без Device Flow.
TWITCH_CLIENT_ID, TWITCH_CLIENT_SECRET	Переопределение встроенного публичного OAuth-приложения Twitch (Device Flow / Helix). Необязательно — без переменной используется значение по умолчанию из src/config.js; тот же Client ID можно также ввести прямо в настройках интерфейса.
VKVIDEO_CLIENT_ID, VKVIDEO_CLIENT_SECRET	OAuth-приложение VK Video / VK ID для серверной части.
DONATIONALERTS_CLIENT_ID, DONATIONALERTS_CLIENT_SECRET	OAuth DonationAlerts.
GOOGLE_OAUTH_CLIENT_ID, GOOGLE_OAUTH_CLIENT_SECRET	OAuth Google для YouTube Data API.
GOOGLE_OAUTH_REDIRECT_PORT	Локальный порт loopback для redirect Google (по умолчанию 45280).
KICK_CLIENT_ID, KICK_CLIENT_SECRET	OAuth Kick (PKCE).
CORONERCHAT_7TV_API_ORIGINS, CORONERCHAT_7TV_ASSET_ORIGINS, CORONERCHAT_7TV_DOH_ENDPOINTS	Дополнительные origin / DoH для 7TV (см. README.md).
CORONERCHAT_DEMO_ACTIVE, CORONERCHAT_DEMO_PORT	Демо-режим (разработка), при необходимости.
CORONERCHAT_ZAPRET_BUILTIN	При значении 1 на Windows включается встроенный путь запуска winws (скачивание zapret-win-bundle при первом запуске, возможен UAC). По умолчанию встроенный режим выключен ; без явного 1 в переменной доступны только собственные команды запуска/останова в блоке Zapret (п. 5.15).
CORONERCHAT_DECK_TOKEN / CHATSHOW_DECK_TOKEN	Общий секрет для чувствительных HTTP-эндпоинтов (debug-логи, экспорт/импорт настроек, управление OBS, /api/deck/* для Stream Deck/Ulanzi) при обращении не с локальной машины — заголовок X-CoronerChat-Deck-Token. С 127.0.0.1 эти эндпоинты всегда доступны без токена; пусто — доступ с других устройств сети к ним запрещён (403).

5.15. Обход DPI (Zapret / winws)

- **Где в интерфейсе:** настройки (иконка шестерёнки) → вкладка «Общие» → блок «Zapret / обход DPI» (расположен после секции «Платформы»). Отдельной вкладки «Zapret» в текущей версии изделия нет.
- **Режим по умолчанию:** встроенный компонент winws из поставки CoronerChat **выключен**. Доступны сохранение **своих команд** запуска и остановки внешнего zapret (PowerShell-скрипт в кавычках или строка для cmd), кнопки «Запустить сейчас» и «Остановить», чекбокс автозапуска при старте приложения. Автозапуск выполняется только если есть что запускать (заполнена своя команда или включён встроенный режим см. ниже).
- **Включение встроенного winws:** перед запуском CoronerChat задайте переменную окружения CORONERCHAT_ZAPRET_BUILTIN=1 (только Windows); возможна загрузка zapret-win-bundle при первом запуске и

запрос повышения прав (UAC). Чекбоксы групп доменов в блоке Zapret и файл дополнительных хостов учитываются во встроенном режиме.

- **Файл дополнительных доменов** (комплект исходников поставки): `resources/zapret/extra-hosts.txt` — по одному FQDN в строке; строки, начинающиеся с `#`, игнорируются. В собранном приложении используется копия рядом с модулями (`zapret-builtin/extra-hosts.txt`).
- Справочно по проекту zapret: репозиторий **bol-van/zapret** и смежные материалы; соблюдайте применимое законодательство при фильтрации трафика.

6. Полный перечень функций изделия, программных интерфейсов и настроек

Ниже зафиксированы пользовательские возможности и программные интерфейсы по коду сервера (`src/server/chat-app-server.js`) и UI (`src/server/web-ui.js`). Точные тела запросов и побочные эффекты см. в исходных текстах и в техническом описании архитектуры.

6.1. Вкладки панели настроек (Drawer) — 2.7.10

- **Общие:** язык UI (i18n); подключение каналов; блок Zapret; правила подсветки; авто-ответы; уведомления; экспорт/импорт настроек.
- **Twitch, VK, YouTube, Kick, Rutube:** OAuth/сессии, каналы/URL эфира, статусы.
- **Донаты:** DonationAlerts (OAuth); MemeAlerts (OBS JWT); мин. суммы; скрытие; общий фильтр «Донаты» в ленте.
- **Эфир:** UX-пакет — чеклист, пресеты, алерты/goal/vote, favorites, keywords, resend, song request, soft-confirm, Safe mode.
- **Статистика / Достижения:** история эфиров, категории/игры, топы чата и донатов, полка ачивок (только в приложении).
- **Вид:** пресеты оформления (в т.ч. Arcade), шрифты, оверлеи, плотность, планшет, скрытые авторы.
- **Интеграции:** OBS WebSocket, now playing, 7TV relay/prefetch, LAN/QR планшет, deck.
- **Обновления:** проверка GitHub Latest, установка Setup, skip-версия.
- **О приложении:** версия, User Guide (long-read), помощник настройки, поддержка автора.

6.2. Главное окно: чат, шапка, вспомогательные панели

- Единая лента с цветовой маркировкой источника; фильтр по платформе; поиск; экспорт и локальная очистка буфера.
- Сводка зрителей, карточки статусов Twitch/VK/YouTube, DonationAlerts/MemeAlerts, права Helix.
- Центр событий (event bus); телеметрия OBS в шапке; кухня / now playing / музыкальный оверлей.
- Композеры Twitch, VK, Kick, YouTube, Rutube; совместная отправка строки в Twitch и VK; ответы VK; подсказки slash и @.
- Модерация у сообщения (Twitch), локальное скрытие, закреп (с проверкой прав), контекст пользователя (ПКМ/история), 7TV picker.
- Карточки событий Twitch (channel points, hype train и др.) согласно фильтрам категорий в настройках вида.

6.3. Инструменты стримера (Twitch / OBS)

- Опросы и предикты; рейд; клип; смена категории и названия стрима; поиск категории.
- Управление эфиром/записью OBS, сценами, выходами, превью через WebSocket.
- Центр модерации (настраиваемый набор быстрых действий через API изделия).

6.4. Now playing, кухня, 7TV, relay, обновления, статистика

- Опрос внешнего bridge, ручное обновление трека, локальная media session Windows.
- Таймер «кухня» с оверлеем; каталоги 7TV; relay и prefetch.
- Автообновление с GitHub Latest (check / install / dismiss) — п. 3.3.
- Локальная статистика эфиров и достижения; публичный recap `?recap=1`.

6.5. Диагностика и перенос

- Экспорт/импорт настроек — файл содержит токены, обращаться как с конфиденциальным носителем.
- Серверный и клиентский журналы, открытие каталога логов (в desktop при поддержке ОС).
- Контекст пользователя для разборов модерации.

6.6. Сводная таблица групп HTTP API (обработчик `handleApiRequest`)

Группа	Маршруты (метод)
Служебные страницы	GET /, /health, /assets/logo.jpg; GET /ws возвращает подсказку (основной канал чата UI — WebSocket upgrade на путь /ws того же хоста и порта)
Состояние и QR	GET /api/state; GET /api/qr?text=...
Авторизация Twitch	POST /api/auth/twitch/client-id start poll logout
Авторизация VK	POST .../vk/config start poll logout browser implicit-token start-implicit; GET .../vk/callback
YouTube (Google)	POST /api/auth/youtube/google logout
DonationAlerts	POST .../donation-alerts/config start poll logout; GET .../callback
Kick	POST .../kick/config start poll logout browser; GET .../kick/callback
Rutube	POST /api/settings/rutube; POST /api/auth/rutube/browser refresh
MemeAlerts	POST /api/meme-alerts/connect disconnect
Каталоги и ассеты	GET /api/7tv/catalog health global-emote asset prefetched-asset, GET/POST prefetch; GET /api/twitch/badges chatters asset emotes; GET /api/vk/catalog asset
Настройки (чтение)	GET /api/settings/obs relay zapret update
Настройки (запись)	POST /api/settings/ui locale obs now-playing youtube kick rutube donation-filters relay zapret update; POST transfer export import
OBS WebSocket	POST /api/obs/launch connect disconnect stream/start stop record/start stop output scene/program; GET /api/obs/outputs scenes preview (+ mixer/mute/studio-mode по сборке)
Now playing / кухня	GET/POST /api/now-playing, POST .../manual refresh; POST /api/kitchen
Модерация и чат	POST /api/channel chat/send chat/clear-local platforms/enable; POST /api/moderation/delete-message timeout-user ban-user unban-user; GET/POST /api/moderation/center; POST /api/admin/pin-message
Twitch-расширения	POST /api/poll/create end, /api/prediction/lock create resolve cancel, /api/raid/start cancel, /api/twitch/clip, /api/stream/category title; GET /api/stream/category-search
Zapret	POST /api/zapret/start stop
Обновления	POST /api/update/check install dismiss
Диагностика	GET /api/debug/logs; POST .../clear open; POST /api/debug/client-log; GET /api/user-context

6.7. Параметры, сохраняемые через POST /api/settings/ui (основной профиль desktop)

Часть полей дублируется в планшетном профиле (tabletUi). Названия ниже — как в теле JSON запроса.

chatFontScale, appUiScale, overlayFontScale, musicOverlayFontScale, chatFontWeight, overlayFontWeight, musicOverlayFontWeight, overlayFontFamily, overlayVariant, overlayFrameStyle, musicOverlayLayout, musicOverlayFontFamily, musicOverlayVariant, musicOverlayFrameStyle, appStylePreset, appFontFamily, appTheme, appAccent, appMessageDensity, appColorVisionMode, appChatBackgroundColor, overlayBackgroundColor, musicOverlayBackgroundColor, showLinkedMediaInApp, showLinkedMediaInOverlay, showLinksInOverlay,

`overlayLinkPlaceholder`, `boldChatText`, `forceEconomyMode`, `highLoadAppCpuPercent`, `highLoadSystemCpuPercent`, `highLoadAppMemoryMb`, `highLoadSystemMemoryPercent`, `animatedInlineBudgetNormal`, `animatedInlineBudgetEconomy`, `overlayAdaptiveAnimationsEnabled`, `overlayAnimatedInlineBudget`, `showDigitalClock`, `showTopStreamSummary`, `showObsTelemetryBar`, `showVkComposerNote`, `showPinnedPanel`, `showRoleActions`, `showModerationCenter`, `showMessagePlatformIcons`, `messagePlatformMarkerMode`, `chatScrollMode`, `showComposerPlatformIcons`, `showSummaryPlatformIcons`, `onboardingCompleted`, `hiddenTwitchAuthors`, `hiddenVkAuthors`, `mentionGlossary`, `dismissedUpdateNoticeVersions`, `headerAlertRedChatError`, `headerAlertRedPollReauth`, `headerAlertRedWsDown`, `headerAlertYellowTwitchScope`, `headerAlertYellowPollScope`, `headerAlertYellowObs`, `headerAlertYellowVkStream`, а также вложенный объект `tabletUi` с аналогами для планшетного режима.

7. Сообщения оператору и устранение типовых неисправностей

Симптом	Возможные действия
Страница не открывается по ожидаемому порту	Проверьте консоль запуска на фактический порт; задайте <code>CORONERCHAT_PORT</code> ; закройте конфликтующий процесс.
Нет сообщений чата	Проверьте сеть, имя канала, статус стрима; для закрытых режимов — авторизацию.
Ошибки OAuth / истёк токен	Выполните выход и повторную авторизацию в интерфейсе изделия.
Оверлей OBS пустой	Убедитесь, что URL содержит актуальный порт и параметр <code>?overlay=1</code> ; обновите источник после перезапуска изделия.
Kick: «авторизуйся OAuth» или чат не подключён	Проверьте Client ID/Secret и redirect; scopes; slug канала; сеть; дождитесь статуса сокета.
YouTube: ошибка Google OAuth / порт занят	Согласуйте <code>GOOGLE_OAUTH_REDIRECT_PORT</code> и redirect URI в Google Cloud; включите YouTube Data API v3.
MemeAlerts не подключается	Проверьте полный URL OBS с <code>obsToken</code> или корректность JWT; статус во вкладке «Донаты». Фильтр ленты «Донаты» показывает и DA, и MA.
DonationAlerts: ошибка OAuth	Redirect URI в кабинете DA должен совпадать с <code>.../api/auth/donation-alerts/callback</code> на вашем origin.
VK: ошибка redirect или сессии	Origin UI и зарегистрированный redirect VK ID должны совпадать по схеме, хосту и порту.
Rutube: чат читается, отправка не работает	Выполните «Встроенное окно входа» во вкладке Rutube (cookie-сессия). Чтение работает без сессии; отправка — только после входа.
После смены языка «отвалились» чекбоксы / UI	Исправлено в 2.7.5 (apply18nDom). Установите Latest с GitHub; при 2.7.4 и ниже — обновитесь вручную.
Предлагают обновиться на 2.35.x вместо 2.7.x	Архивные теги. С 2.7.5 канал читает GitHub Latest . Один раз поставьте 2.7.5+ с Releases.
Интерфейс «не той» версии / пропали новые пункты настроек	На порту <code>3210</code> может отвечать уже запущенная другая копия CoronerChat. Закройте лишний процесс или запустите вторую копию с другим <code>CORONERCHAT_PORT</code> .
Zapret: кнопка запуска неактивна, встроенный режим недоступен	По умолчанию нужны свои команды в блоке Zapret (Общие) или включите встроенный путь: <code>CORONERCHAT_ZAPRET_BUILTIN=1</code> перед запуском изделия (Windows).

Расширенная диагностика — в файле `runtime.log` в каталоге данных изделия; чувствительные поля в журнале маскируются.

8. Информационная безопасность и защита данных

Security posture: CoronerChat спроектирован как **local-first** операторский клиент с безопасными значениями по умолчанию. Архитектура и эксплуатационная модель сформированы с учётом контролей **ISO/IEC 27001/27002**, privacy framework **ISO/IEC 29100**, практик **NIST CSF / OWASP ASVS**, а также потенциально применимых требований **GDPR** и законодательства РФ. Оператор самостоятельно определяет применимость требований, роли controller/processor, правовые основания и сроки хранения. Публичная выжимка: [docs/security.html](#). Раздел не является сертификацией и не заменяет юридическую консультацию.

8.0. Итог соответствия (summary)

- **Конфиденциальность:** HTTP UI по умолчанию только на `127.0.0.1`; внешние API — исключительно по TLS; секреты не уходят в публичный overlay.
- **Целостность:** официальный канал обновлений GitHub Releases (Latest); чувствительные LAN-эндпоинты требуют deck-токен.
- **Доступность:** локальное хранение состояния; резервное копирование `CoronerChat-data`; восстановление авторизаций через повторный OAuth.
- **Privacy by design:** минимизация подключённых площадок, удаление данных при очистке каталога, маскирование секретов в журнале.

8.1. Нормативная база (международная + РФ)

Эксплуатация сопряжена с обработкой **секретов доступа** (OAuth-токены, client secrets, ключи MemeAlerts) и, как правило, с обработкой **персональных данных** участников чата (ники, тексты, идентификаторы платформ). Ориентиры:

- **ISO/IEC 27001 / 27002** — системные контроли ИБ (доступ, криптография на транспорте, инциденты, поставщики).
- **ISO/IEC 29100** и **GDPR** — принципы privacy: законность, минимизация, ограничение хранения, права субъекта.
- **NIST CSF / OWASP ASVS** — Identify→Protect→Detect→Respond→Recover; безопасные дефолты приложения.
- **152-ФЗ / 149-ФЗ** — ПДн и защита информации на территории РФ.
- **ПП РФ № 1119, Приказ ФСТЭК № 17** — корпоративный контур ИСПДн и модель угроз (справочно).
- **ГОСТ Р ИСО/МЭК 27001** — национальная гармонизация ISO/IEC 27001.

8.2. Роли и зона ответственности

- **Владелец ПК / локальный администратор:** установка ОС и обновлений, антивирус, учётные записи Windows, межсетевой экран, физический доступ к машине, резервное копирование каталога данных.
- **Оператор-стример (пользователь изделия):** настройка CoronerChat, авторизации на внешних платформах, решение о трансляции/записи чата, передача экспорта настроек, реакция на инциденты утечки токена.
- **Организация-работодатель (при корпоративном использовании):** локальный акт о ПДн, назначение ответственного, учёт ИСПДн при необходимости, согласования с ИБ и юридической службой.

8.3. Активы и носители информации в изделии

Актив	Содержание / риск	Рекомендуемые меры
app-state и смежные JSON	Токены OAuth, настройки UI, списки скрытых авторов, глоссарий, закрепы	Шифрование тома ОС (BitLocker), ограничение доступа к профилю, запрет копирования на съёмные носители без шифрования
runtime.log	Диагностика; секреты маскируются, но остаётся контекст событий	Ротация и выборочная выгрузка; удаление перед передачей ПК в ремонт
Экспорт настроек	Полный дамп конфигурации, включая секреты	Хранить только в зашифрованном архиве; не отправлять в открытом виде
Кэш 7TV и медиа	Изображения эмодзи, метаданные каналов	Очищать при утилизации ПК; не считать «анонимным» при привязке к аккаунту
Оперативная память и сеть	Сообщения чата, ответы API	Не оставлять сессию без присмотра; блокировка рабочей станции по Win+L

8.4. Персональные данные (152-ФЗ): состав и минимизация

Тексты сообщений чата, отображаемые ники и сопутствующие идентификаторы пользователей платформ могут квалифицироваться как ПДн при отсутствии безусловной анонимизации. Рекомендуется:

- Зафиксировать **цель обработки** (ведение эфира, модерация, архив стрима) и **правовое основание** (согласие, договор, законный интерес — по решению юриста организации).
- Применять **минимизацию**: не включать лишние платформы; не экспортировать полные логи публично; ограничить время хранения резервных копий CoronerChat-data .
- Для трансграничной передачи (Twitch, Google, Kick и др.) — оценить требования **ст. 12 152-ФЗ** и политики платформ; документировать факт передачи в локальном регламенте.
- Обеспечить возможность **блокировки** обработки и удаления локальных данных при прекращении использования (удаление каталога данных и деинсталляция).

8.5. Модель угроз (сжато, по духу документов ФСТЭК)

Угроза	Проявление для CoronerChat	Меры
НСД локально	Доступ постороннего к открытому UI на ПК стримера	Блокировка сессии ОС; отдельная учётная запись Windows; не хранить пароль OBS в открытом виде рядом с ПК
НСД по сети	CORONERCHAT_HOST=0.0.0.0 в ЛВС без фильтрации	По умолчанию привязка уже к 127.0.0.1 ; при осознанном расширении до 0.0.0.0 — чувствительные эндпоинты (debug, экспорт/импорт настроек, OBS) с недоверенных адресов требуют CORONERCHAT_DECK_TOKEN (403 без него, п. 5.14, 8.6); дополнительно — сегментация VLAN + ACL, HTTPS-терминирование только при доверенной обратной прокси-конфигурации (не входит в изделие)
Вредоносное ПО	Кража app-state , перехват буфера обмена с токенами	Антивирус с актуальными базами; контроль автозагрузки; запрет неподписанных инсталляторов
Утечка по каналу поддержки	Отправка runtime.log или экспорта без редактирования	Ручная вырезка секретов; отдельный защищённый канал передачи
Ошибка прав модерации	Случайный бан/таймаут зрителя	Разграничить учётные записи «стример» и «модератор»; использовать тестовый канал
Компрометация OAuth	Утечка refresh-токена	Немедленный «Выйти» на всех сервисах изделия; отзыв приложения в кабинетах Twitch/VK/Google и смена секретов

8.6. Технические меры на стороне изделия и среды

- Изделие использует **TLS при обращении к внешним HTTPS API** платформ; локальное хранилище состояния на диске не шифрует изделие самостоятельно — защита обеспечивается средствами ОС и организационными мерами.
- Заголовки ответов HTML/JSON для API настроены с Cache-Control: no-store — снижение риска кэширования секретов промежуточными прокси (при корректной конфигурации браузера).
- Журнал runtime.log применяет **маскирование** секретов — не полагаться на маскирование как на единственную защиту при публикации логов.
- По умолчанию HTTP-сервер привязан к 127.0.0.1 (см. п. 5.14). Чувствительные и разрушающие маршруты (debug-логи, экспорт/импорт настроек, управление OBS, /api/deck/*) дополнительно проверяют источник запроса: с localhost — всегда разрешено, с другого адреса сети — только с корректным заголовком X-CoronerChat-Deck-Token , иначе 403/503.
- Компонент zapret/winws при включении затрагивает сетевой стек ОС целиком; ошибочная конфигурация может нарушить доступность других приложений и создать новые векторы атаки — включать только уполномоченным персоналом и с отдельным локальным регламентом.

8.7. Организационные меры

- Утвердить **политику паролей** для OBS WebSocket и для учётных записей платформ.
- Вести **учёт носителей** с резервными копиями `CoronerChat-data`.
- При увольнении сотрудника — **очистка** авторизаций в изделии и смена паролей связанных сервисов.
- Привязать использование CoronerChat к **инструкции по ИБ** организации (если применимо).

8.8. Инциденты информационной безопасности

При подозрении на компрометацию токенов или несанкционированный доступ: немедленно выполнить выход из всех интеграций в UI, удалить/заменить скомпрометированные OAuth-приложения в кабинетах платформ, сменить пароль OBS, проверить целостность исполняемых файлов изделия, при необходимости уведомить пользователей чата и **уполномоченный орган по 152-ФЗ** (для юридических лиц — в порядке, установленном локальными актами и законом).

8.9. Соответствие требованиям при обработке ПДн в ИСПДн

При корпоративном развёртывании CoronerChat как элемента ИСПДн оператор-организация получает **готовый прикладной контур с безопасными дефолтами** (localhost, TLS к площадкам, маскирование логов, gated LAN API). Дополнительно организация классифицирует ИСПДн, ведёт модель угроз и локальные акты. Изделие не является средством сертификации уровня защищённости само по себе — оно **обеспечивает техническую базу**, на которой достигается полное соответствие при соблюдении чеклиста оператора (см. также `docs/security.html`).

9. Техническое обслуживание, резервное копирование и контроль работоспособности

- **Резервное копирование:** периодически копируйте каталог `CoronerChat-data` (portable / win-unpacked) или пользовательский профиль, указанный в документации к сборке, для сохранения настроек и истории состояния.
- **Журналы:** при обращении в поддержку приложите фрагмент `runtime.log` за период воспроизведения проблемы, удалив из скриншотов и вложений чувствительные данные вручную.
- **Контроль на поставляемой сборке:** в составе изделия может быть доступна встроенная smoke-проверка (см. `README.md`, ключ `--run-release-tests` для исполняемого файла).
- **Сопровождение:** изменения в программе сопровождаются обновлением комплекта документации; актуальность настоящего руководства проверяйте по версии в `package.json` и сопутствующим файлам в каталоге `docs/`.

Печать: формат A4. Architecture description: `docs/coronerchat-architecture-print.html`. Security statement: `docs/security.html`.