

Serein: техническая документация

1. Обзор / Назначение

Serein - настольный клиент для подключения к серверам и выполнения операций с терминалами, файлами, контейнерами, базами данных и удалёнными рабочими столами. Операции выполняются с правами учётной записи, указанной для подключения.

Область документа: опубликованная версия 1.5.2, локальные изменения Compose от 2026-10-22 и миграция доставки обновлений в исходном дереве от 2026-10-23. Сохранение Compose-проектов после Down и регистрация по пути YAML относятся к локальной сборке. Их наличие в распространённом пакете с номером 1.5.2 не следует определять только по номеру версии.

Платформы клиента: Windows x64 и Linux x64. Лицензия локального исходного кода: Apache 2.0. По сообщению владельца от 2026-10-22, прежний GitHub-репозиторий недоступен, <https://serein-dev.xyz/> развёрнут на домашнем сервере через NPM.

Опубликованные пакеты 1.5.2 размещены в <https://serein-dev.xyz/releases/1.5.2/>. Манифест обновлений: <https://serein-dev.xyz/updates/latest.json>. На сайте доступны исходники тега v1.5.2 (коммит 00454cc14072f938b8659b282afe65ca27c7c9e7), portable ZIP, SHA256SUMS.txt и исходный список сумм прежнего выпуска. Архив serein-1.5.2-sbom-reconstructed.zip восстановлен 23.10.2026 по тегу: Rust приложения, RDP-помощник и npm; это не исходный GitHub-архив. Включённый PROVENANCE.txt описывает метод восстановления. Подписи опубликованных Windows/Linux updater-файлов проверены прежним публичным ключом приложения.

2. Архитектура и компоненты

flowchart LR

```
UI[React 18 / TypeScript / WebView] -->|invoke / events / channels| Core[Tauri 2 / Rust]
Core --> Profile[Локальный профиль / секреты / журнал]
Core -->|SSH| Server[Удалённый сервер]
Server --> Files[Файлы / процессы / Docker Compose]
Core -->|Каналы внутри SSH| Data[Базы данных / RDP / VNC]
Core --> Helper[Процесс serein-rdp]
```

Схема показывает логические зависимости. RDP-помощник участвует в передаче данных рабочего стола; схема не описывает побайтовый маршрут кадров.

Компонент	Ответственность и граница
src/renderer	Панели, вкладки, терминал xterm, редактор CodeMirror. Интерфейс обращается к Rust через window.api.
src/api/index.ts, src/shared/types.ts	Вызовы Tauri и типы сообщений. Это внутренний интерфейс приложения, не HTTP API.
src-tauri/src/commands	Обработчики IPC, поиск активной сессии, вызов предметных модулей, запись отдельных действий в журнал.

Компонент	Ответственность и граница
src-tauri/src	SSH, SFTP/SCP, процессы PTY, хранилище, Docker, БД, задачи, политики.
rdp-helper	Процесс RDP-помощника. В Windows-пакете требуется serein-rdp.exe из того же выпуска, что и основной исполняемый файл.
Программы на сервере	SSH-сервис обязателен для SSH-сценариев. Docker Compose, sqlite3, RDP/VNC и команды управления службами требуются соответствующим операциям.

Сетевые соединения пользователя направляются на выбранные им адреса. В текущем исходном дереве updater использует только `https://serein-dev.xyz/updates/latest.json`, а ручная загрузка ведёт в `https://serein-dev.xyz/#downloads`. Генератор манифеста по умолчанию формирует URL `https://serein-dev.xyz/releases/<version>/<asset>`; SEREIN_UPDATE_BASE допускает только HTTPS. Подписи и публичный ключ не менялись. Уже выпущенные бинарные файлы 1.5.2 содержат старые адреса проверки: эти встроенные адреса меняются только при новой сборке. Для них сохранён совместимый манифест по прежнему адресу `nehade.xyz/updates/terminal/latest.json`; его URL файлов уже указывают на `serein-dev.xyz`. Новая версия программы этой миграцией не выпускалась. Режим «Закрытый контур» отключает проверки обновлений, но не блокирует подключения пользователя.

Источники: API-мост, конфигурация Tauri, модель безопасности.

3. Спецификация / API / Основные сущности

3.1. Профиль сервера и сессия

Поле / сущность	Допустимые значения и поведение
ServerConfig.id / ID сессии	ID профиля сохраняется между запусками. ID активной сессии выдаётся при подключении. В команды Docker и файлов передаётся ID сессии; наличие профиля не означает наличие соединения.
connection	ssh, serial, telnet, raw. Отсутствующее значение означает ssh. Telnet и raw TCP сами по себе не шифруют трафик.
authType	password, key, agent. Для ключа используется путь к приватному ключу. Пустой agentKey означает перебор ключей SSH-агента.
proxyJump / proxyCommand	proxyJump ссылается на другой сохранённый профиль и имеет приоритет над proxyCommand. proxyCommand запускает локальную программу; импорт такой записи требует проверки команды.
executeOnConnect	Команда отправляется после открытия сессии и может изменить состояние сервера.
tunnels	Сохранённые пробросы поднимаются после подключения. local открывает локальный порт к адресу со стороны сервера; remote открывает порт сервера с пересылкой на локальное назначение; dynamic создаёт SOCKS5-прокси.

При первом SSH-подключении пользователь подтверждает ключ сервера. Изменившийся отпечаток останавливает подключение. Фоновые подключения принимают только ранее подтверждённые ключи, включая ключи шлюзов. Ошибка чтения хранилища доверенных ключей не трактуется как первое подключение.

Источники: типы, SECURITY.md.

3.2. Профиль данных, секреты и политики

Каталог профиля: %APPDATA%\serein на Windows, каталог конфигурации serein по правилам XDG на Linux. SEREIN_CONFIG_DIR переопределяет каталог до запуска. Один профиль открывается одним процессом. Переносная сборка без этой переменной использует профиль по умолчанию.

Данные	Содержимое и побочный эффект
servers.json, secrets.json	Профили и защищённые секреты либо ссылки на системное хранилище. Копирование secrets.json не заменяет экспорт между компьютерами.
settings.json, known_hosts.json	Настройки и подтверждённые ключи серверов. Удаление доверенной записи приводит к повторной проверке сервера.
tasks.json, workspaces.json	Задачи и сохранённые рабочие пространства.
docker-compose-projects.json	Локальный реестр Compose по ID профиля сервера. Не содержит резервной копии YAML или данных контейнеров.
Журналы сеансов и действий	Могут содержать команды, пути и тексты запросов. При включённой отправке syslog запись передаётся на заданный адрес.

Защита секретов: DPAPI на Windows, Secret Service через системную связку на Linux. Мастер-пароль добавляет слой scrypt и AES-256-GCM. Экспорт .tbk защищается паролем и переносит данные профиля; файлы сервера, базы данных и игровые миры в него не входят.

Политики читаются из HKLM\SOFTWARE\Policies\Serein и защищённого %ProgramData%\Serein\policy.json на Windows, /etc/serein/policy.json на Linux. Проверяется защита источника, а не только его имя. Недопустимая или незащищённая политика не применяется; ошибка отображается и записывается в журнал. Политика не является системной изоляцией процесса.

Источники: хранилище, политики, SECURITY.md.

3.3. Файловая операция и сохранение редактора

SFTP используется при наличии подсистемы на сервере; для поддерживаемых сценариев предусмотрен SCP. Запись из редактора использует временный файл и замену целевого. Ошибка установки требуемых прав в SFTP прекращает сохранение.

У SFTP конфликт определяется по времени изменения: при ненулевых текущем и базовом mtime разница более 1000 мс возвращает {ok:false, conflict:true}. Это не сравнение содержимого. В реализации SCP base_mtime не проверяется, поэтому одинаковое обнаружение конфликта для двух транспортов не гарантируется.

Сравнение папок выбирает новые и изменённые файлы для загрузки, не удаляя лишние удалённые файлы. Отмена передачи не откатывает другие уже завершённые операции.

Источники: SFTP, SCP, файловые команды.

3.4. Compose: внутренний контракт локальной сборки

window.api.docker вызывает команды Rust. Параметр id означает активную SSH-сессию.

Операция	Контракт, изменения состояния и ошибки
composeList(id)	Опрос Compose и при необходимости меток контейнеров, объединение с сохранённым реестром. Таймаут каждого опроса - 15 с. Ошибка опроса не заменяется

Операция	Контракт, изменения состояния и ошибки
	статусом «все остановлены». Ошибка чтения или записи реестра возвращается вызывающему коду.
<code>composeRegister(id, composeFile)</code>	Принимает абсолютный POSIX-путь после удаления крайних пробелов. Управляющие символы запрещены. Выполняет <code>docker compose -f <path> config --format json</code> , предел ожидания 20 с. При успехе сохраняет имя и путь. Развёрнутая конфигурация с подставленными секретами не возвращается в интерфейс. Возвращаемое «Не запущен» является значением регистрации, а не отдельной проверкой состояния контейнеров.
<code>composeAction(id, composeFile, project, action, service?)</code>	<code>action</code> : up, down, start, stop, restart. up и down работают с проектом; остальным нужен <code>service</code> . Имя проекта/службы соответствует <code>[A-Za-z0-9][A-Za-z0-9_.-]*</code> . up требует известного абсолютного POSIX-пути и вызывает <code>up -d</code> . Остальные действия могут использовать имя проекта без пути.
<code>composeRead(id, composeFile)</code>	Читает первые 400 строк файла; это просмотр, не получение полного содержимого без ограничения. Неизвестный путь возвращает ошибку.
<code>composeLogs(...)</code>	Передаёт журнал выбранной службы, начиная с последних 200 строк, затем следует за новыми записями. При закрытии панели отмените поток через <code>cancelComposeLogs</code> и снимите подписку на события.

Перед изменяющим действием проект сохраняется в локальный реестр. Если сохранить реестр не удалось, удалённое действие не запускается. Предел ожидания действия - 60 с. Таймаут клиента не подтверждает отсутствие изменений на сервере: перед повтором необходимо запросить состояние.

`down` вызывает `Compose` без `-v`: удаляет контейнеры проекта и управляемую им сеть, не запрашивает удаление томов. Данные в удаляемом записываемом слое контейнера теряются. Сохранность подключённых томов и каталогов зависит от конфигурации проекта и внешних действий. `stop` сохраняет контейнер. Сохранённый реестр позволяет снова показать проект после `down`, но не восстанавливает удалённый YAML.

Два канала ошибки обязательны к обработке: отклонённый `Promise` и успешный вызов с `{ok:false, error}`. Отсутствие сессии возвращает Сессия не подключена. Ненулевой код `Compose`, неверный JSON, имя или путь не считаются успехом.

Источники: API, обработчики, валидация и реестр.

3.5. Задачи и подключения к БД

Шаги задачи на одном сервере выполняются по порядку; разные серверы могут обрабатываться параллельно. Виды шага: `command`, `upload`, `download`, `sync`, `service`, `docker`, `healthcheck`. Условия: `success`, `failure`, `always`. Ошибка прекращает обычную последовательность на этом сервере, если не задано `continueOnError`; шаги обработки ошибки и завершения определяются условиями.

Предел шага по умолчанию - 300 с, верхний предел - 21600 с. Повторы ограничены 10; повтор изменяющего шага может повторить уже выполненный побочный эффект. Пробный прогон подключается и выполняет проверки, но не запускает пользовательские команды. Он не является исполнением или откатом. История содержит до 50 запусков; вывод шага ограничен 64 КиБ в текущем окне и 8 КиБ в истории.

Виды подключения БД: `postgres`, `mysql` (включая `MariaDB`), `mssql`, `sqlite`, `mongo`, `redis`. Адрес интерпретируется со стороны SSH-сервера; при отсутствии используется `127.0.0.1`. Порты по умолчанию: 5432, 3306, 1433, 27017, 6379 соответственно для сетевых протоколов; `SQLite` использует файл и программу `sqlite3` на сервере.

В `db.rs` заданы пределы результатов: 5000 строк, суммарно 16 МиБ учитываемых значений, 64 КиБ на ячейку, 100 наборов результатов. Это не предел всей памяти процесса. Клиентский срок запроса - 30 с; при его истечении соединение БД закрывается. Серверный предел 28 с применяется там, где поддержан протоколом. Отмена PostgreSQL пытается сохранить соединение; для остальных движков соединение закрывается. Заккрытие не доказывает откат выполненного запроса.

Источники: задачи, БД, команды БД.

TODO(doc): [определить гарантии совместимости внутреннего IPC между версиями и опубликовать полный перечень контрактов с кодами ошибок. Текущие обработчики используют текстовые ошибки, а не единый каталог кодов.]

4. Инструкция по разворачиванию / использованию

4.1. Установка клиента

1. Скачайте опубликованный пакет с <https://serein-dev.xyz/#downloads> и проверьте SHA-256 по списку соответствующего выпуска. Порядок проверки указан в разделе 4.4.
2. На Windows установите WebView2, если он отсутствует. Запустите установщик либо распакуйте переносной пакет целиком, включая RDP-помощник.
3. На Linux выберите DEB или AppImage, предусмотренные `tauri.linux.conf.json`. Для DEB требуются `libwebkit2gtk-4.1-0` и `libayatana-appindicator3-1`. Наличие пакета не подтверждает совместимость со всеми дистрибутивами.
4. До первого запуска задайте `SEREIN_CONFIG_DIR`, если нужен отдельный профиль. Иначе используется каталог по умолчанию.
5. Откройте клиент. Для SSH-сценария обеспечьте сетевой доступ к SSH-сервису и права учётной записи.

4.2. Подключение и операции

1. Создайте профиль: адрес, порт, пользователь, способ входа. При использовании шлюза проверьте выбранный профиль шлюза.
2. Сверьте отпечаток SSH-ключа с администратором сервера до подтверждения.
3. Откройте терминал и проверьте имя сервера. Для проверки без изменения состояния можно выполнить `hostname`.
4. Передайте тестовый текстовый файл в каталог с правом записи. Откройте файл, сохраните изменение, перечитайте результат с сервера.
5. Для Compose откройте Docker → Compose. Если проект не найден, в локальной сборке зарегистрируйте абсолютный путь к его YAML. Перед `up` проверьте путь и имя проекта.
6. Перед `Down` определите, где хранятся данные приложения: том, подключённый каталог или слой контейнера. После действия обновите список и проверьте фактическое состояние.
7. Перед задачей проверьте серверы, параметры, условия шагов и план пробного прогона. После запуска оцените результат каждого сервера; общий запуск не означает успех всех шагов.
8. Для резервной копии профиля выполните зашифрованный экспорт `.tbk`. Данные сервера резервируйте отдельно. При переносе сначала просмотрите состав импорта и разрешаемые `ProxyCommand`.

4.3. Сборка из исходного дерева

1. Получите исходное дерево и установите Node ^20.19.0 || >=22.12.0, Rust 1.96.0 с rustfmt и clippy.
2. На Windows установите средства сборки MSVC C++ и WebView2. На Linux установите зависимости из CONTRIBUTING.md и docs/LINUX_MIGRATION.md.
3. Из корня проекта выполните:

```
npm ci

npm run typecheck

npm run lint

npm test

cargo fmt --manifest-path src-tauri/Cargo.toml --check

cargo clippy --manifest-path src-tauri/Cargo.toml --all-targets -- -D warnings

cargo test --manifest-path src-tauri/Cargo.toml --lib
```

4. Для разработки выполните `npm run tauri -- dev`. Адрес dev-сервера: 127.0.0.1:1420.
5. Для Windows выполните `npm run tauri -- build`; для Linux - `npm run build:linux`. Подпись обновлений требует настроенного ключа сборочной среды. Сборка не публикует пакет автоматически.

Источники: package.json, тулчейн, Linux-конфигурация, CONTRIBUTING.md.

4.4. Сайт продукта, документы и проверка загрузок

Контакт сайта: hadeworks@gmail.com. Раздел #license содержит описание Apache License 2.0 и полный текст из LICENSE; скачать его можно по /documents/LICENSE.txt. Раздел #start описывает установку, импорт SSH config (без паролей), ручное добавление сервера и проверку отпечатка при подключении. Инструкция AppImage разделяет Debian 13+/Ubuntu 24.04+ (libfuse2t64), Debian 12/Ubuntu 22.04 (libfuse2) и Arch (pacman -Syu fuse2). Команды установки системных пакетов изменяют ОС; Arch-команда выполняет полное обновление. Приложение запускается без sudo. При недоступности FUSE указан --appimage-extract-and-run. Поддержка каждого производного дистрибутива отдельно не подтверждена.

Сайт содержит обзор из четырёх скриншотов приложения с ручным переключением и клавишами стрелок. Загрузки выделяют Windows/Linux по данным браузера; определение ОС не скрывает другие пакеты. Добавлены установка, история шести выпусков и контакт автора. Поиск работает локально по выбранному языку документации, от двух символов, показывает до 12 фрагментов. Команды копируются через Clipboard API; при отказе предлагается ручное копирование. Оглавление расположено в начале документа и прокручивается вместе с ним, не перекрывая текст. Панель выбора языка и PDF остаётся закреплённой; анимации учитывают паузу и reduced-motion.

Сайт <https://serein-dev.xyz/> содержит разделы #home, #about, #downloads, #docs. Раздача пакетов и исходников находится в /releases/1.5.2/, манифест - /updates/latest.json. Документация читается в браузере на русском и английском; скачиваемые экспорты предоставляются в PDF. Markdown остаётся источником документов. Карточки загрузок содержат краткое назначение файлов; происхождение SBOM описано в технической документации и в PROVENANCE.txt внутри архива. Относительные ссылки на код требуют исходного дерева соответствующей версии; исходники скачиваются отдельным ZIP.

1. Выберите файл своей платформы в разделе загрузок.
2. Получите <https://serein-dev.xyz/releases/1.5.2/SHA256SUMS.txt> и сравните SHA-256 скачанного файла.
3. При несовпадении суммы не запускайте файл; повторно получите пакет и уточните выпуск.

4. Для состава зависимостей используйте восстановленный SBOM и прочитайте `PROVENANCE.txt`. Историческая сумма старого SBOM не является суммой пересозданного архива.

Windows setup и Linux ApplImage не пересобирались: их байты и подписи совпадают с опубликованным выпуском. Исходники из тега не содержат локальных исправлений от 22.10.2026 и миграции endpoint от 23.10.2026. Сайт не изменяет конфигурацию установленного клиента. Настройки темы и паузы анимаций сохраняются в `localStorage`; недоступное хранилище не блокирует чтение и скачивание.

Файлы раздаются nginx из каталога `/mnt/material/serein` через `read-only mount /srv/artifacts`. NPM завершает TLS и проксирует `serein-dev.xyz` на `http://serein-site:80`. Для выпуска сначала размещаются подписанные файлы и суммы, затем атомарно заменяется `/mnt/material/serein/updates/latest.json`. Несуществующие файлы возвращают HTTP 404; `latest.json` отдаётся с `Cache-Control: no-cache`.

5. Известные ограничения и обработка ошибок

Состояние	Обработка и граница гарантии
SSH-сессия отсутствует или разорвана	Восстановить соединение и повторно проверить состояние перед изменяющей операцией. Наличие ID профиля не заменяет активную сессию.
Отпечаток SSH изменился	Установить причину с администратором. Не удалять доверенную запись только ради обхода отказа.
Compose завершился по таймауту	Запросить состояние контейнеров и журнал. Не повторять изменяющую команду автоматически.
Реестр Compose повреждён	Сохранить копию файла для диагностики. Ошибка не означает отсутствие проектов на сервере. Не создавать пустой реестр поверх данных без восстановления состава.
Конфликт редактора	Сравнить локальный текст и файл сервера. При SCP отсутствует проверка <code>base_mtime</code> , действующая в SFTP.
Запрос БД обрезан или прерван	Проверить признак обрезки, ограничить запрос, при закрытом соединении подключиться заново. Не трактовать результат клиента как подтверждение отката транзакции.
Политика не применена	Проверить сообщение об ошибке, формат и права источника. Не считать приложение ограниченным этой политикой до подтверждения применения.
Отправка syslog	Поддерживаются UDP/TCP без TLS. Для недоверенного участка требуется отдельно защищённый канал или ретранслятор.
RDP	Доверие основано на SSH-сервере; RDP-сертификат не проверяется по системному списку доверия. Совместимость и задержки зависят от сервера и сети.
Компрометация учётной записи ПК	Шифрование профиля не защищает от процесса, контролирующего эту учётную запись или память запущенного клиента.

Отдельная готовая версия macOS не подтверждена материалами этой сборки. Telnet/raw TCP не предоставляют защиту SSH. Экспорт профиля и сравнение папок не являются резервной копией сервера.

TODO(doc): [зафиксировать проверенную матрицу версий Windows, Linux-дистрибутивов, SSH-серверов, Docker Compose, СУБД и RDP/VNC.]

TODO(doc): [определить поддерживаемое число одновременных сессий и измерения производительности с указанием оборудования, сети и версии сборки.]

TODO(doc): [опубликовать проверенный порядок восстановления повреждённого профиля и реестра Compose с критериями сохранности данных.]