

Подробный паспорт Штата

Тимон Ядро · Архитектор и разработчик интеграции

Редакция 2026-09-15 · Редакция 15 сентября 2026 года; состояние функций уточняется документом готовности

Область применимости и актуальные ограничения:
<https://team-on.ru/yadro/docs/readiness.html>

Оглавление

Область применения источника

1. Цель и смысл
2. Пользователи и проверяемая ценность
3. Системная карта
4. Информационная архитектура
5. Архитектура и источники
6. Данные и состояния
7. Основной сценарий
8. Безопасность и доступность
9. Ошибки и эксплуатация
10. Границы и интеграции
11. Приёмка
12. Открытые решения

Область применения источника

В составе Ядра реализован собственный ограниченный веб-пилот Штата поверх Core Bridge. Он предоставляет вход, список разрешённых агентов, собственную историю, текст, ограниченные вложения, состояние и остановку. Публичная страница входа доступна.

Полный исходный интерфейс Staff развивается отдельно и использует другой runtime. Наличие его репозитория и версии не означает перенос всех его экранов в текущий Core. Далее сохранены подробные обязанности целевого компонента; расширенный жизненный цикл задач и артефактов требует отдельной реализации и приёмки.

1. Цель и смысл

Дать сотруднику место, где понятно, кому поручить задачу, какие материалы передать, что сейчас происходит и где получить результат. Штат объединяет агента, разговор, задание и документы в связный рабочий процесс, а не ограничивается текстовым чатом.

Переход с ноутбука или из мессенджера не должен терять ранее начатую работу. Для серверных заданий установка Шлюза не обязательна; устройство подключают, когда задача действительно требует локального контекста или действия.

2. Пользователи и проверяемая ценность

Основной пользователь — сотрудник с назначенными агентами и областями. Руководитель может работать с предоставленной областью как обычный участник; отдельный продукт Штаб сюда не переносится. Ценность измеряется завершёнными задачами, доступностью результата после возврата, временем поиска нужного файла и понятностью ошибок. Метрики не подменяют факт созданного артефакта.

3. Системная карта

```
flowchart LR
    Person[Сотрудник] --> UI[Штат: навигация / диалог / результат]
    UI --> Web[Web backend: сессия и права]
    Web --> Engine[Двигатель: данные и выполнение]
    Engine --> Web
    Web --> UI
    UI --> Pair[Подключение Шлюза по необходимости]
```

Ключ LLM и технический Core token не выдаются браузеру. Web backend — доверенный серверный адаптер, но не второй владелец истории/задач.

4. Информационная архитектура

Раздел	Объекты и действия	Важное состояние
Моя работа	Активные задачи, продолжение, результаты	Требуется ответ/подтверждение, актуальность события
Агенты	Назначение, доступные функции, выбор	Нет доступа / функция не настроена
Диалог	Текст, вложения, tools, отмена	Загрузка, выполнение, reconnect, неизвестный исход

Раздел	Объекты и действия	Важное состояние
Результаты	Файл, задача, источник, версия	Подготовка, готово, ошибка, удалено
Знания	Доступные материалы, область, поиск	Извлечение/индексация и дата обновления
Устройства	Pairing, разрешения, отзыв	Подключено, сон, недоступно, отозвано

Знания не означают автоматически готовую поисковую/embedding систему. Для каждой функции интерфейс опирается на capabilities принятой серверной конфигурации.

5. Архитектура и источники

Изученный snapshot nglain/teamon-staff: 90e0abf5e95ae5e12b32a9c960bfdded0a5e3e5ff, product/ui/package.json 0.3.15. Presentation: product/ui/assets/staff.js и staff.css, локальные fonts/icons. Transport источника: openclaw-gateway.js и openclaw-conversation.js. Его нельзя механически направить на Core URL: нужны совместимые контракты.

Повторная проверка 10.09.2026: получен HEAD 0e4687f10f149eb5f165e57d1f4116220bc076f0, UI 0.3.17. Прочитаны package.json, импорты staff.js и таблица маршрутов index.js: транспорт по-прежнему OpenClaw, добавлен conversation-search.js. Это кандидат для переноса представления; полный функциональный diff и интеграционная приёмка с Двигателем ещё не выполнены.

Адаптер Ядра получает профиль/agent list, читает историю с pagination/cursor, создаёт и продолжает task, загружает/выдаёт файлы, показывает события и отмену. Точные public routes появятся при реализации; существующий Core bridge — источник форматов, но web session не выдаётся за токен устройства. Ревизии Staff и права на assets/dependencies сверяются до переноса. Лицензионные уведомления сохраняются.

6. Данные и состояния

Долговременные conversation/run/files/context принадлежат Двигателю. Штат хранит только временное состояние представления: выбранный агент, раскрытые панели, неотправленный ввод по согласованной политике. Содержание рабочих документов не кэшируется бессрочно в общедоступном browser storage. Выход пользователя очищает локальные приватные данные и закрывает подписки.

После отправки сохраняется связь с request/run ID. Reload восстанавливает task; сетевая ошибка не означает, что сервер не принял сообщение. Повторная отправка использует исходную идентичность запроса или проверяет status, а не создаёт дубль. Поток модели и поток статусов инструментов визуально различаются.

7. Основной сценарий

Войти → выбрать назначенного агента → добавить документ → сформулировать задачу → увидеть чтение/обработку → получить файл → открыть источник → вернуться после reload. Если нужны действия на компьютере, пользователь подключает Шлюз и видит запрошенные разрешения. Сценарий заканчивается доступным результатом, а не только сообщением «готово».

8. Безопасность и доступность

UI не задаёт роли; backend проверяет каждый объект и скачивание. Доступ по прямой ссылке равен доступу через навигацию. Markdown/имена файлов считаются недоверенными. Активные HTML/документы не исполняются в origin с авторизованной cookie. Не передаём модельные ключи браузеру; используем серверные сессии и CSRF-защиту.

Проверяем клавиатуру, фокус, подписи, контраст, масштаб 200%, размер экрана ноутбука, длинные имена и ошибки загрузки. Русские названия едины с Штурвалом и Шлюзом. «Нет доступа», «не настроено» и «временно недоступно» — разные сообщения.

9. Ошибки и эксплуатация

После reconnect показывается реальный status; при истечении сессии требуется вход, но уже запущенная задача не исчезает. При удалении/отзыве документа ссылку нельзя продолжать выдавать из cache. При несовместимом backend UI сообщает о версии. Сборка сайта содержит локальные assets без обязательного CDN; отдельный HTTPS origin. Метрики: загрузка, ошибки API/upload, задержка событий и фактическое скачивание; содержание документов в telemetry не отправляется.

10. Границы и интеграции

Штат не управляет системными секретами/ролями и не выполняет shell сам по себе. Он не копирует runtime Штаба или Master. Результаты и проектные поверхности из Staff adapters входят только после реализации Core ACL и принятого BOM.

11. Приёмка

ENT-05/06/07/08/13/14/29; DEMO-01/02/03/05/08. Нужны работа с реальной моделью, правильный артефакт, reload/resume, отрицательный тест чужого документа, истечение сессии, доступность интерфейса и проверка установленной версии assets. Скриншот макета или работа исходного UI с OpenClaw этот результат не заменяет.

12. Открытые решения

Версия переносимого UI после diff, web API/schema, IdP, хранение drafts, объём файлов, preview isolation и обработка больших результатов. Все решения отражаются в UX и контракте, затем подтверждаются на одном сквозном документном сценарии.

Общий контекст: системная карта (см. оглавление документации), критерии приёмки (см. оглавление документации), план демо (см. оглавление документации).

Основание документа

Сверка продуктовых документов и закреплённых источников 15 сентября 2026 года. Проверка документации не заменяет промышленную приёмку. Полная применимость определяется документом о состоянии компонентов.