

# Архитектура и взаимодействие компонентов

Тимон Ядро · Архитектор, интегратор, технический заказчик

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

Общая схема

Владение состоянием

Веб пилот

Прохождение задания

Изменения через Штурвал

Модели и внешние системы

Развёртывание и отказоустойчивость

Operator и корпоративный MCP

## Общая схема

Ядро объединяет пять пользовательских компонентов. Двигатель ведёт серверных агентов; Штат предоставляет сотруднику веб-интерфейс; Штурвал обслуживает оператора компании; Штрих представляет личный слой сотрудника; Шлюз связывает разрешённую вкладку с выбранным исполнителем.

Node и Native Messaging host — поддерживающие процессы существующей связи с устройством. OpenCode — исполнитель модельного цикла, а выбранная модель — отдельная зависимость. Для систем компании используются настроенные коннекторы. Эти части требуют согласованной версии и собственной проверки работоспособности.

### Пять компонентов Тимон Ядро

Ответственность и состояние связей · 15 сентября 2026 года



#### Связанная личная беседа Штрих — Шлюз через Node

Проверенный интеграционный кандидат. Установка согласованного комплекта и приёмка отдельно.

#### Отдельные зависимости и поверхности

OpenCode и модель · Operator MCP · встроенный TeamON MCP · системы компании

Публичный консультант сайта использует отдельный профиль OpenCode без корпоративных инструментов.

Схема не объявляет общую корпоративную историю всех интерфейсов реализованной.

#### Пять компонентов Ядра и границы текущей интеграции

Линии на схеме показывают владельцев и назначения связей. Подписанный статус имеет приоритет: личный мост Штрих — Шлюз проверен как кандидат, а общее корпоративное продолжение через все поверхности пока не принято.

## Владение состоянием

Core хранит состояние серверной работы и её историю в рамках существующих идентичностей каналов. OpenCode ведёт технический цикл исполнения. Веб-интерфейс хранит временное представление и ссылки на операции. NotchChatStore в Штрихе остаётся владельцем связанной личной беседы; расширение хранит локальный черновик и её восстанавливаемую проекцию.

Три маршрута Шлюза — личный Codex, Штрих и корпоративный TeamON — не являются одним хранилищем. Смена исполнителя не объединяет историю, не копирует личную память в компанию и не создаёт полномочия. Существующее серверное продолжение связано с каналом и его edge-сессией. Универсальный идентификатор работы для свободного перехода между всеми интерфейсами не считается реализованным.

Человек может работать с серверным агентом через подключённый канал либо веб-Штат. Для продолжения точно той же корпоративной работы в Штрихе необходимы отдельный контракт идентичности, выбора беседы, прав и синхронизации. В личном локальном мосте такой выбор уже ограничен конкретной связанной беседой и сайтом.

## Веб пилот

В срезе от 11 сентября 2026 года один backend обслуживает отдельные пути Штата и Штурвала. Он использует локальные учётные записи с индивидуальными длинными кодами и серверными привязками к Core. Это пилотная схема входа, не реализация корпоративного OIDC или произвольных проектных ролей.

Сеансы двух приложений разделены. Секреты Core не выдаются браузеру. Backend разрешает ограниченный список маршрутов; он не является универсальным прокси к Core. Состояние и история поступают через Bridge. Подробный контракт опубликован в справочнике API пилота.

## Прохождение задания

После принятия поручения необходимо сохранить его идентичность и связь с историей. Затем выполняются модельный цикл и разрешённые инструменты. Пользователь получает состояние и результат. При перезагрузке страницы интерфейс должен восстанавливать сведения о работе, а не считать потерю соединения доказательством отсутствия задания.

Для операций с побочными эффектами важны исходное намерение, идентификатор, подтверждение и фактический результат. Возможная повторная доставка события не должна превращаться в повтор бизнес-действия. При неизвестном результате требуется проверка состояния, а не безусловный повтор на другой модели или с новым идентификатором.

## Изменения через Штурвал

Пилот поддерживает чтение состояния агента и изменение имени, описания и языка через подготовку и отдельное подтверждение. Core возвращает предложение, связанное с ревизией и контрольным представлением изменения. Оператор проверяет содержание перед применением.

При конфликте ревизии нужна новая подготовка. При таймауте после подтверждения сначала читают статус первоначальной операции. Остальные возможности полного Штурвала, включая расширенные роли и управление подключениями, описаны в паспорте как целевой состав и не следуют из наличия трёх редактируемых настроек.

## Модели и внешние системы

Модель подключается отдельно от двигателя. Совместимость включает не только возможность отправить текст, но и вызовы инструментов, ошибки, ограничения, отмену и фактическое качество задач. Сходство API не является достаточным доказательством совместимости.

Инструменты обращаются к системам компании с согласованными полномочиями. Если модель использует внешний API, передаваемые данные и сетевой маршрут определяются политикой организации. Для автономной конфигурации модель и все необходимые сервисы должны быть доступны внутри контура; незаявленное переключение на внешний сервис недопустимо.

## Развёртывание и отказоустойчивость

Совместное размещение компонентов на одном сервере допустимо для выбранного профиля, но не означает высокой доступности. Отказ узла, заполнение диска, потеря модели и недоступность коннектора имеют разные последствия. Архитектура восстановления должна учитывать как данные Core, так и файлы, конфигурацию и необходимые секреты.

До промышленной эксплуатации фиксируют версии компонентов, зависимости, расположение данных, резервное копирование, миграции и проверку восстановления. Значения RPO, RTO, допустимой нагрузки и перечень поддерживаемых платформ определяются испытаниями и соглашением, а не этой обзорной схемой.

## Operator и корпоративный MCP

Operator MCP предоставляет разрешённые административные чтения и действия существующего Core. Отдельное рабочее пространство Operator показывает фиксированный интерфейс и разрешённые чтения через тот же контур авторизации. Оно не является заменой пользовательского Штата и не расширяет само по себе веб-пилот Штурвала.

Встроенный TeamON MCP — ещё один слой: агент получает через него сведения о компании, агенте, пользователе и состоянии системы. В проверенном срезе Core доступны 46 инструментов, 21 ресурс и 9 prompts. Число описывает конкретную ревизию исходников, а не гарантированную доступность каждого инструмента в любом экземпляре. Эффективный набор определяется ролью, конфигурацией и версией сервера.

Коннектор MCP корпоративной системы, административный Operator MCP и встроенный TeamON MCP имеют разные задачи. Установка MCP-клиента не предоставляет доступ к компании без действующей серверной привязки и полномочий.

## Основание документа

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