

Техническое описание архитектуры — Grace CRM Assistant

Версия: 1.0 **Дата:** 4 мая 2026

Статус: Этап 0 -- согласование файл 00 [architecture.md](#)

Проект: Мультиагентная AI-система для отдела продаж I-TECH

1. Назначение системы

Grace CRM Assistant -- мультиагентная AI-система, развёртываемая поверх корпоративной CRM Grace ([grace.i-tech.su](#)). Система выполняет advisory-функцию: анализирует данные CRM, контролирует качество данных, формирует рекомендации менеджерам, обеспечивает управленческий контроль -- не нарушая принципа **Grace CRM = source of truth**.

Все изменения в CRM производятся только через API Grace. Система не имеет прямого доступа к базе данных MySQL.

2. Состав системы

Система включает два функциональных контура.

Контур 1 -- Sales AI (Grace CRM Assistant)

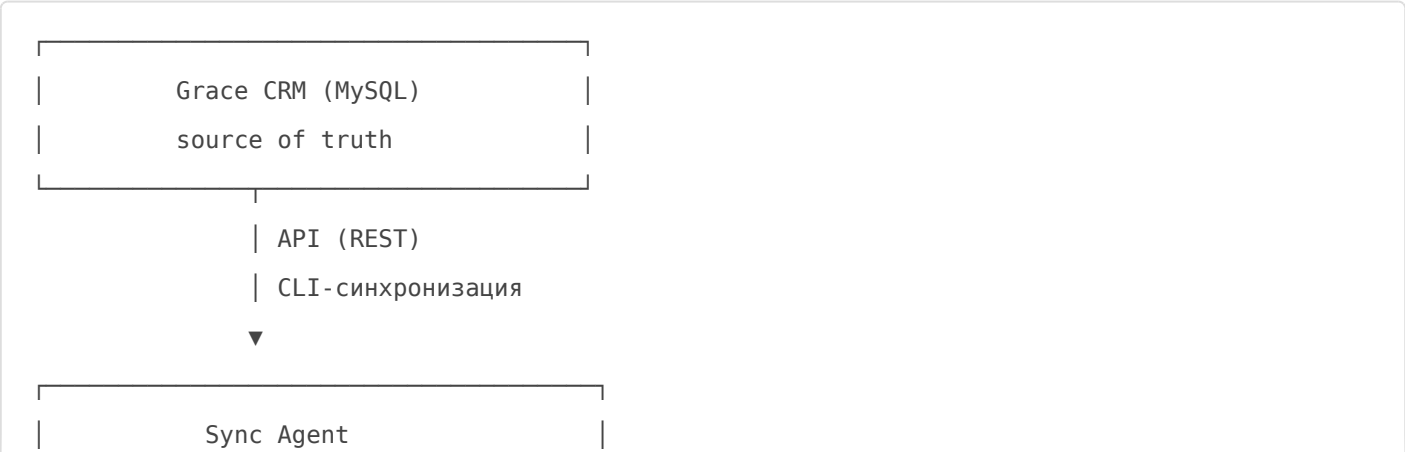
Агент	Наименование	Продукт агента
Sync Agent	Агент синхронизации	Актуальный поисковый образ данных CRM в OpenSearch
Quality Agent	Агент контроля качества	Реестр проектов с критическими пробелами данных
Recommendation Agent	Агент рекомендаций	Конкретный следующий шаг по каждому активному проекту
Notification Agent	Агент уведомлений	Доставленное уведомление с подтверждённой реакцией
Orchestrator	Оркестратор	Бесперебойная работа агентной системы как единого целого

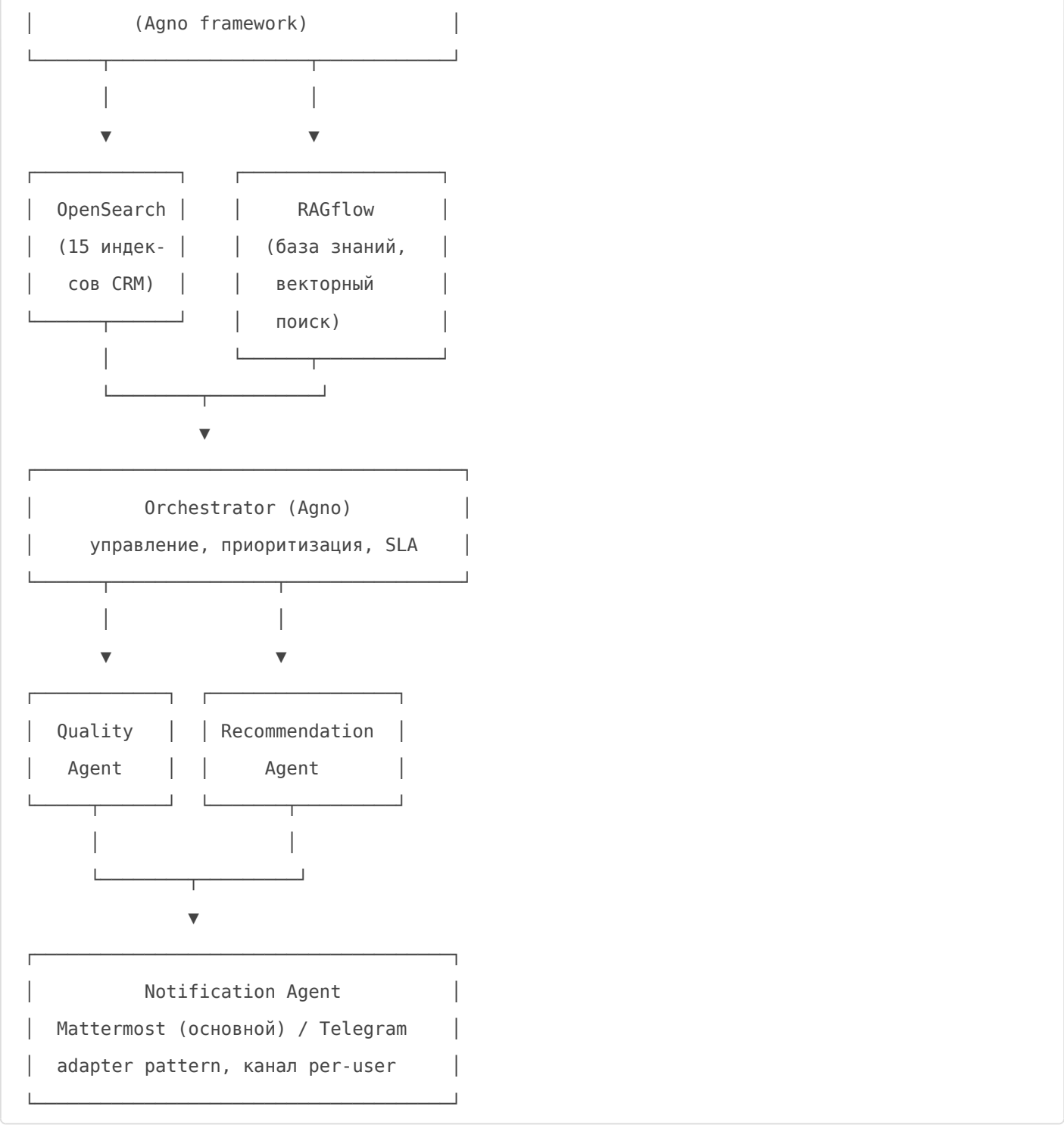
Контур 2 -- Knowledge AI (база знаний)

Агент	Наименование	Продукт агента
Ingest Agent	Агент загрузки знаний	Проиндексированный документ в RAGflow с метаданными
Query Agent	Агент поиска знаний	Контекстный ответ с цитатой из источника
Lint Agent	Агент аудита знаний	Отчёт о противоречиях и устаревших данных в базе знаний

3. Архитектура решения

3.1 Общая схема





3.2 Слой данных

Слой	Система	Роль
Источник	Grace CRM (MySQL)	Source of truth для операционных данных
Поисковый	OpenSearch (15 индексов)	Быстрый доступ, аналитика, downstream для агентов

Слой	Система	Роль
Семантический	RAGflow	База знаний, векторный поиск, RAG

3.3 Ключевые принципы архитектуры

- Grace CRM -- единственный источник операционных данных
 - AI-система выполняет advisory-функцию, не операционную
 - Все записи в CRM -- только через API Grace
 - Слабая связанность: агенты независимы, новые роли добавляются без изменения существующих
 - Событийная модель: cron-polling + ручной запуск (webhooks -- при наличии поддержки в API Grace)
 - On-premise развёртывание
-

4. Описание агентов -- продуктовый подход

Каждый агент описывается через продуктовую формулу:

Продукт × Внутренний клиент × Триггер × Критерий выполнения

4.1 Sync Agent -- Агент синхронизации

Продукт: Актуальный поисковый образ данных Grace CRM в OpenSearch -- гарантирующий, что любой запрос от downstream-агентов отражает реальное состояние CRM на момент последнего запуска.

Параметр	Содержание
Внутренний клиент	Quality Agent, Recommendation Agent
Механизм	CLI -> API Grace CRM -> трансформация -> bulk index в OpenSearch
Триггер	Cron по расписанию + ручной запуск администратором
Входные данные	API Grace CRM: projects, activities, clients, tasks, comments, calculations, orders, users, objects
Выходной продукт	Обновлённые индексы OpenSearch (15 индексов)

Параметр	Содержание
Критерий выполнения	Все индексы обновлены без ошибок; расхождение Grace CRM / OpenSearch = 0 по завершении синхронизации

Режимы работы:

Режим	Триггер	Что синхронизируется
Инкрементальный	Cron каждые N минут	Изменения с последнего запуска (параметр <code>updated_since</code>)
Полный	Ночной cron / ручной запуск администратором	Все сущности, полная пересборка индексов

Входные данные (API endpoints):

```
GET /api/projects?updated_since=timestamp
GET /api/projects/{id}
GET /api/activities?updated_since=timestamp
GET /api/clients?updated_since=timestamp
GET /api/clients/{id}
GET /api/tasks?updated_since=timestamp
GET /api/comments?entity_type=&entity_id=
GET /api/calculations?updated_since=timestamp
GET /api/orders?updated_since=timestamp
GET /api/users
GET /api/objects?updated_since=timestamp
```

Красный флаг: ошибка синхронизации -> downstream-агенты работают на устаревших данных -> все рекомендации невалидны.

4.2 Quality Agent -- Агент контроля качества данных

Продукт: Ежедневный реестр проектов с критическими пробелами данных -- для менеджеров и РОПа -- с указанием конкретного поля, ответственного и срока устранения.

Параметр	Содержание
Внутренний клиент 1	Менеджер -- задача на заполнение по своим проектам
Внутренний клиент 2	РОП -- сводка по команде, эскалация при просрочке

Параметр	Содержание
Триггер	Ежедневно в 09:00 + при изменении статуса проекта
Входные данные	OpenSearch: <code>itech_projects</code> , <code>itech_calculations</code> , <code>itech_activities</code> ; правила проверки из конфига
Выходной продукт	Структурированный реестр: проект -> пробел -> ответственный -> срок; сигнал в Notification Agent
Критерий выполнения	Доля проектов с полными данными $\geq 85\%$; каждый пробел имеет назначенного ответственного

Правила проверки (настраиваются в конфиге):

Проверка	Поле	Условие
Обязательные поля	<code>account_id</code> , <code>property_id</code> , <code>project_status_id</code> , <code>probability_new</code> , <code>forecast_date</code>	Не пусто
Следующий шаг	<code>next_action</code> , <code>next_action_date</code>	Заполнено и дата не в прошлом
Логика статусов	<code>calculation_status</code>	При статусе «КП выставлено» -- существует расчёт
Активность	<code>last_activity_at</code>	Не более 30 дней назад для активных проектов

4.3 Recommendation Agent -- Агент рекомендаций

Продукт: Конкретный следующий шаг по каждому активному проекту -- для менеджера -- сформулированный на основе истории CRM и базы знаний RAGflow, доставляемый не позднее 2 часов после изменения статуса или по запросу.

Параметр	Содержание
Внутренний клиент	Менеджер (первично); РОП (при просрочке реакции)
Триггер	Изменение статуса проекта / запрос менеджера / просрочка контакта
Входные данные	OpenSearch (<code>itech_projects</code> , <code>itech_calculations</code> , <code>itech_comments</code> , <code>itech_accounts</code>); RAGflow (кейсы, скрипты, возражения)
Выходной продукт	Рекомендация: действие + срок + обоснование + ссылка на кейс из базы знаний

Параметр	Содержание
Критерий выполнения	Менеджер знает следующий шаг по каждому активному проекту; нет проектов без зафиксированного следующего действия

Формат рекомендации:

Проект: [название] Клиент: [название] Стадия: [статус]Последний контакт: N дней назад
Рекомендация: [конкретное действие]Срок: [дата]Обоснование: [краткое обоснование]Похожий кейс: [ссылка из базы знаний]

4.4 Notification Agent -- Агент уведомлений

Продукт: Адресное уведомление нужному человеку в нужный момент через нужный канал -- без информационного шума -- как условие того, что сигналы системы реально доходят и вызывают реакцию.

Параметр	Содержание
Внутренний клиент	Менеджер / РОП / Зам. ГД -- в зависимости от типа события
Триггер	Сигнал от любого агента системы
Входные данные	Сигнал от агента + профиль получателя (предпочтительный канал)
Выходной продукт	Доставленное уведомление с подтверждённой реакцией
Механизм доставки	Adapter pattern: Mattermost (основной) / Telegram (резерв), канал настраивается per-user
Критерий выполнения	Реакция в течение N часов; при отсутствии реакции -- эскалация в Orchestrator

Матрица маршрутизации уведомлений:

Событие	Получатель	Приоритет
Пробел в данных по своему проекту	Менеджер	Нормальный
Просрочка реакции менеджера > SLA	РОП	Высокий
Рекомендация по проекту	Менеджер	Нормальный

Событие	Получатель	Приоритет
Критическое событие (крупная сделка, риск ухода)	РОП, Зам. ГД	Высокий
Сводка по команде	РОП	Ежедневный

4.5 Orchestrator -- Оркестратор

Продукт: Бесперебойная работа агентной системы как единого целого -- обеспечивающая, что ни одно критическое событие не теряется и каждый сигнал доходит до нужного человека в нужное время.

Параметр	Содержание
Внутренний клиент	РОП, Зам. ГД
Триггер	Постоянно -- реагирует на сигналы всех агентов
Входные данные	Сигналы от всех агентов; SLA-параметры из конфига
Выходной продукт	Гарантия непрерывности потока сигналов; каждое событие имеет назначенного ответственного
Критерий выполнения	Отсутствие потерянных эскалаций; система работает без ручного вмешательства

4.6 Ingest Agent -- Агент загрузки знаний

Продукт: Проиндексированный документ в RAGflow с корректными метаданными -- готовый к поиску Query Agent.

Параметр	Содержание
Внутренний клиент	Query Agent, Recommendation Agent
Триггер	Загрузка нового документа ответственным за базу знаний
Входные данные	Документ (PDF, DOCX, MD) + метаданные: отрасль, тип клиента, стадия сделки, продукт, теги
Выходной продукт	Проиндексированная запись в RAGflow; обновлённый граф связей
Критерий выполнения	Документ доступен для поиска; метаданные заполнены полностью

4.7 Query Agent -- Агент поиска знаний

Продукт: Контекстный ответ с цитатой из источника -- для менеджера -- за ≤ 5 секунд.

Параметр	Содержание
Внутренний клиент	Менеджер, Recommendation Agent
Триггер	Запрос менеджера / вызов от Recommendation Agent
Входные данные	Текстовый запрос + контекст проекта
Выходной продукт	Ответ + цитата + ссылка на источник
Критерий выполнения	Релевантный ответ ≤ 5 сек; источник всегда указан

4.8 Lint Agent -- Агент аудита знаний

Продукт: Еженедельный отчёт о противоречиях, устаревших данных и логических разрывах в базе знаний.

Параметр	Содержание
Внутренний клиент	Ответственный за базу знаний
Триггер	Еженедельно
Входные данные	Все документы в RAGflow
Выходной продукт	Отчёт: список противоречий, устаревших фактов, незаполненных разделов
Критерий выполнения	Выявлены все документы старше 6 месяцев и документы с конфликтующими утверждениями

5. Технологический стек

Слой	Технология	Роль
Агентный фреймворк	Agno	Агенты, Teams, оркестрация
LLM (основной)	OpenAI GPT-4o / GPT-4o-mini	Рассуждение, генерация рекомендаций
LLM (резерв)	GigaChat / YandexGPT	Fallback при недоступности OpenAI

Слой	Технология	Роль
Поисковый слой	OpenSearch 2.13	15 индексов CRM, аналитика
База знаний	RAGflow	Семантический поиск, RAG, граф знаний
Коммуникации	Mattermost (основной) + Telegram (резерв)	Доставка уведомлений, adapter pattern
Синхронизация	CLI + Agno Agent	Grace CRM API -> OpenSearch
Планировщик	Cron / systemd timer	Расписание синхронизации и агентов
Деплой	Docker Compose	On-premise, I-TECH инфраструктура

Выбор фреймворка агентов базы знаний

Для контура Knowledge AI выбран **RAGflow** как платформа базы знаний по следующим основаниям:

Критерий	RAGflow
Развёртывание	Self-hosted, on-premise -- соответствует требованиям I-TECH
Поиск	Гибридный: semantic + full-text
Граф знаний	Поддерживается нативно
Интеграция с Agno	Через REST API -- Agno вызывает RAGflow как tool
Мультиформатность	PDF, DOCX, MD, TXT

Агенты контура Knowledge AI реализованы на **Agno** и взаимодействуют с RAGflow через его REST API как с внешним инструментом (tool call). Прямой встройки RAGflow в агентный фреймворк не требуется.

6. Цепочка данных (основной сценарий)

Grace CRM (MySQL) ↓ CLI → APISync Agent ↓ bulk indexOpenSearch (15 индексов) ↓
запрос агентовQuality Agent → реестр пробелов → Notification Agent → Mattermost/Telegram →
МенеджерRecommendation Agent → рекомендация → Notification Agent → Mattermost/Telegram →
Менеджер ↑RAGflow (база знаний) – контекст для рекомендаций

7. Что остаётся за рамками Этапа 0

Тема	Этап
Калькулятор премии (Кк)	Этап 1
Telegram-бот (UI)	Этап 2
Полная интеграция RAGflow + Ingest/Query/Lint агентов	Этап 2
Интеграция с 1С ЕРП	Этап 3

8. Программа и методика приёмо-сдаточных испытаний (ПМИ)

8.1 Общие принципы

ПМИ фиксирует проверку каждого функционального блока системы на соответствие требованиям Этапа 0.

Формат тест-кейса:

- Идентификатор
- Описание
- Входные данные
- Шаги выполнения
- Ожидаемый результат (эталон)

- Допустимое отклонение
 - Критерий: пройдено / не пройдено
-

8.2 ПМИ -- Sync Agent

ТС-SA-01 -- Инкрементальная синхронизация

- Вход: изменения в CRM (≥ 1 запись)
- Шаги: запуск `grace-sync run --mode incremental`
- Ожидаемо: записи появляются в OpenSearch
- Отклонение: $\leq 1\%$ записей может быть с задержкой
- Критерий: все записи синхронизированы

ТС-SA-02 -- Полная синхронизация

- Вход: полный запуск
 - Ожидаемо: алиас переключён, данные консистентны
 - Критерий: расхождение с CRM = 0
-

8.3 ПМИ -- Quality Agent

ТС-QA-01 -- Обнаружение пробелов

- Вход: проект без forecast_date
- Ожидаемо: проект попадает в реестр
- Критерий: найден 100% таких проектов

ТС-QA-02 -- SLA-эскалация

- Вход: нет реакции > SLA
 - Ожидаемо: уведомление РОП
 - Критерий: эскалация сработала
-

8.4 ПМИ -- Recommendation Agent

ТС-RA-01 -- Генерация рекомендации

- Вход: проект без активности >14 дней
- Ожидаемо: рекомендация сформирована ≤ 2 часа
- Критерий: рекомендация содержит действие, срок, обоснование

8.5 ПМИ -- Notification Agent

ТС-NA-01 -- Доставка уведомления

- Вход: событие от агента
 - Ожидаемо: сообщение доставлено
 - Критерий: получено пользователем
-

8.6 ПМИ -- API

ТС-API-01 -- Получение проектов

- Вход: GET /projects
 - Ожидаемо: список проектов
 - Критерий: HTTP 200, корректный JSON
-

8.7 ПМИ -- Knowledge AI

ТС-КВ-01 -- Поиск

- Вход: запрос
 - Ожидаемо: ответ с источником
 - Критерий: время ≤ 5 сек
-

9. Прототипы дашбордов OpenSearch Dashboards

9.1 Общие требования

- Фильтры: период, тип клиента
 - Экспорт: Excel / PDF
-

9.2 Дашборд «Менеджер»

Фильтры: период, стадия, тип клиента

Виджеты:

- Воронка продаж по стадиям
 - Мои проекты (таблица)
 - Просроченные действия
 - Топ «спящих» клиентов
 - Рекомендации (список)
-

9.3 Дашборд «РОП»

Фильтры: менеджер, период, стадия, тип клиента

Виджеты:

- Воронка по команде (динамика)
 - Распределение заказов и проектов по менеджерам
 - Топ-10 спящих клиентов
 - Средний Кк по отделу
Кк = 1 - (количество проектов с ошибками / общее количество проектов)
 - Эскалации:
 - проект
 - менеджер
 - время просрочки
 - статус
-

9.4 Дашборд «Зам. ГД»

Виджеты:

- Created / Shipped / Backlog (динамика) -- график, показывающий изменение объёма созданных заказов (Created), фактически отгруженной выручки (Shipped) и накопленного незавершённого портфеля (Backlog) во времени; позволяет оценить баланс между продажами, отгрузками и ростом незавершёнки
- Growth YoY (%) -- показатель год-к-году, отражающий темп роста ключевых метрик (Created, Shipped, Backlog) относительно аналогичного периода прошлого года; используется для оценки динамики бизнеса

- Backlog vs Shipped -- сравнительный график объёма незавершённого портфеля (Backlog) и фактической отгрузки (Shipped); позволяет выявить перегрев воронки и риски накопления незавершёнки
- Производственная нагрузка (items, hours) -- метрики загрузки производства: количество активных позиций (items) и суммарные трудозатраты (hours); позволяет оценить соответствие объёма продаж возможностям производства
- Monthly trend -- помесечная динамика ключевых показателей (Created, Shipped); используется для выявления сезонности, пиков и провалов продаж

Дополнительно:

- Прогноз выручки

10. Развёртывание и инфраструктура

10.1 Сервер

- ОС: Ubuntu 24.04 LTS
- CPU: Intel Xeon E5-2620 v4 (8 ядер, 2.1 GHz)
- RAM: 32 GB
- Диск: 2×240 GB SSD

10.2 Размещение компонентов

Компонент	Размещение
OpenSearch	Docker container
RAGflow	Docker container
Agno Agents	Docker container
API Layer	Docker container
Mattermost	Внешний / контейнер

10.3 Требования к ресурсам

- OpenSearch: ≥ 16 GB RAM
 - RAGflow: ≥ 8 GB RAM
 - Agents/API: ≥ 4 GB RAM
-

10.4 Сеть и доступ

- Внутренний контур (VPN / LAN)
 - Доступ по HTTP(S)
 - Ограничение по IP
-

10.5 Запуск (базовый)

```
docker-compose up -d
```

10.6 Резервное копирование

- OpenSearch snapshot (ежедневно)
 - RAGflow storage backup
-

10.7 Мониторинг

- Health endpoint `/api/v1/health`
 - Логи Docker
 - Метрики OpenSearch
-

Revision #1

Created 2026-06-22 03:22:39 UTC by Claude Ops

Updated 2026-06-22 03:22:39 UTC by Claude Ops