

Приложения

Детализация и справочные материалы к регламенту: грамматика тем (А), карта потока (Б), сущностная модель (В), словарь (Г), архитектура связей (Д), структура базы знаний (Е). Не отдельные стандарты, а развёртка ядра — то, на что регламент опирается и ссылается.

- [Приложение А: Zulip АйТек — грамматика тем и индексация Опух](#)
- [Приложение Б: Zulip АйТек — каналы по потоку создания ценности \(крупная клетка\)](#)
- [Приложение В: Zulip АйТек — сущностная модель и жизненный цикл проекта \(заказа\)](#)
- [Приложение Г: Словарь — корпоративная база знаний \(KB\)](#)
- [Приложение Д: Архитектура и связи](#)
- [Приложение Е: Структура базы знаний \(BookStack\)](#)

Приложение А: Zulip АйТек

— грамматика тем и индексация Опух

Приложение А к РГ-СМК-Zulip-01.

Детализация к регламенту: правила именования тем и индексации Опух. Две оси: **канал** = кто участвует и кто это видит (стабильная ось, граница доступа и индексации Опух); **тема** = о чём конкретно (единица знания, то, что Опух достаёт и цитирует). Канонические описания каналов, словарь и стартовый контур — в регламенте.

Схема



Канал — медленная ось: каналов мало, они почти не меняются. Тема — быстрая: их много, они динамичны. Вся дисциплина сводится к тому, чтобы не путать две оси и держать

Описания каналов (для `description` в Zulip)

Грамматико-ориентированная версия; канонические формулировки — в регламенте, п. 3.

#продажи

“ Операционная работа отдела продаж: лиды, клиенты, сделки, проблемные кейсы. Изменения процессов и инициативы — НЕ сюда, а в `#развитие`. Вопросы качества и претензии клиентов — в `#качество`. Темы: `Клиент:`, `Сделка:`, `Проблемный кейс`.

#развитие

“ Внутренние инициативы и изменения: цифровизация, интеграция систем, изменения СМК. Одна тема = одна инициатива (`Инициатива: ...`). Итог фиксируем темой `Решение: ...`. Текущие: `Инициатива: Внедрение ERP`, `Инициатива: Внедрение Zulip`, `Инициатива: Интеграция GRACE`. Темы: `Инициатива:`.

#качество

“ Контроль (ОТК, лаборатория) и система менеджмента качества (ISO 9001): регламенты, аудиты, несоответствия и корректирующие действия, рекламации, анализ со стороны руководства. Изменение/пересмотр регламента как инициатива — в `#развитие`, сюда возвращаем ссылкой. Темы: `Регламент:`, `Аудит:`, `Несоответствие:`, `САРА:`, `Рекламация:`, `Анализ рука:`.

#проекты (сквозной — заказы)

“ Сквозные заказы: одна тема = один проект **Проект: <объект> / <№>**, первое сообщение — карточка, статус по шлюзам G0-G8. Детально — приложение В (сущностная модель).

Грамматика тем (закрытый список)

Префикс + стабильное имя. Префиксы — закрытый список, синонимы не плодим (**Заявка: / Лид: / Обращение:** → выбрано одно). Именно **префикс + имя** становится меткой документа в Опух.

Префикс	Канал	Что обозначает	Пример имени
Проект:	#проекты	заказ (сквозная тема)	Проект: ПС-Северная / 2026-312
Инициатива:	#развитие	внутренняя инициатива / изменение	Инициатива: Внедрение ERP
Клиент:	#продажи	контекст по клиенту	Клиент: 000 Ромашка
Сделка:	#продажи	конкретная сделка / пайплайн	Сделка: ВРУ для Ромашки
Проблемный кейс	#продажи	проблемная операционная ситуация	Проблемный кейс: срыв отгрузки
Регламент:	#качество	обсуждение регламента (канон — страница KB)	Регламент: Входной контроль
Аудит:	#качество	внутренний или внешний аудит	Аудит: Внутренний, Q2-2026
Несоответствие:	#качество	зафиксированное несоответствие	Несоответствие: партия №312
САРА:	#качество	корректирующее / предупреждающее действие	САРА: по несоответствию №312
Анализ рук-ва:	#качество	анализ со стороны руководства	Анализ рук-ва: 2026 H1

Префикс	Канал	Что обозначает	Пример имени
Решение:	любой	зафиксированное решение + обоснование	Решение: выбор корп. мессенджера
Рекламация:	#качество (ссылка из #продажи)	претензия клиента + разбор	Рекламация: Ромашка, дефект сборки

Правила

Канон vs обсуждение. Документ — источник истины (регламент, стандарт КД, методика) живёт страницей в KB (BookStack). В Zulip под `Регламент:` идёт его обсуждение и изменение, со ссылкой на страницу KB. `Решение:`, ставшее нормой, повышается до страницы KB. Не держим канонические документы «вечными» темами в чате — иначе истина расплзается между чатом и базой.

Изменение vs эксплуатация — куда писать «изменение СМК». *Изменение* с началом, концом и проектным управлением → `#развитие` (`Инициатива: Пересмотр регламентов под новую редакцию ISO`). *Установившийся артефакт и его эксплуатация* (сам регламент в KB, его аудит, несоответствия по нему) → `#качество`. СМК ведёт steady-state в своём канале, а изменение заводит инициативой в `#развитие` и оттуда ссылается назад.

Решение: — **институциональная память.** Как только тред дошёл до решения, заводим (или переименовываем тему в) `Решение: ...` с самим решением и кратким обоснованием. Это самый ценный для Опух тип: на запрос «что мы решили по X» движок отдаёт чистую цитату, а не разбирает переписку.

Рекламация: — **один источник истины на инцидент.** Факт услышан в `#продажи`, но авторитетная запись и корректирующее действие — в `#качество` (`Рекламация: ...` + связанная `САРА: ...`). Из `#продажи` ставим ссылку через `#качество>тему`. Не дублируем обсуждение в двух каналах — иначе в индексе окажется три полуверсии одного инцидента.

Стабильность тем. Тему не переиспользуют под новый предмет — заводят новую. Один «документ» не должен противоречить сам себе во времени.

Один словарь. Типы тем и сущности должны совпадать с лексикой словаря и разделов KB, чтобы Опух, индексируя все источники, видел согласованную онтологию, а не несколько языков для одних сущностей.

Связь с Опух

- **Канал/полка = граница индексации и доступа.** Коннектор (Zulip, BookStack) забирает то, что видит его токен. Подписка/доступ = осознанное решение «идёт в RAG: да/нет».
- **Источники и приоритет.** KB (BookStack) — канон, отличные RAG-документы (полка/книга/глава = готовая метадата, версии). Zulip — обсуждение и свежий статус. В ассистентах приоритет KB для «что является истиной», Zulip — для «что нового».
- **Тема/страница = единица retrieval.** Префикс + стабильное имя работают как заголовок документа и метка цитаты; по ним строятся ассистенты (например, только по `Решение:` -темам или по document set раздела СМК в KB).
- **Чувствительное защищаем не правами Zulip, а тем, что коннектор не подписан** — либо отдельным document set с ограниченным доступом. Пер-юзерные права источников в Опух автоматически не переносятся.

Масштабирование (разделы KB)

Текущие области — проекция будущей карты разделов базы знаний; при расширении ничего не переделывается.

Сейчас	Раздел KB	Потом
<code>#продажи</code>	Коммерческий	+ <code>#снабжение</code> , <code>#финансы</code> и т.д.
<code>#качество</code>	СМК / ISO 9001	—
<code>#развитие</code>	<i>ортогонален</i>	слой изменений для всех разделов
—	Инженерный	<code>#проектирование</code>
—	Производственный	<code>#производство</code>

Чек-лист

При создании канала:

- Решить: индексирует ли его Опух-бот (да/нет), записать в `description`.
- Чувствительное (цены, переговоры, претензии с юр. риском) → бот не подписан или отдельный document set.
- В `description` указать «что сюда / что не сюда» и список префиксов тем.

При создании темы:

- Префикс — из закрытого списка выше.
- Имя стабильное; новый предмет → новая тема.
- Дошли до решения в треде → завести/переименовать в `Решение: ...`; норма → страница KB.
- Рекламация → запись в `#качество`, ссылка из `#продажи`.

Приложение Б: Zulip АйТек — каналы по потоку создания ценности (крупная клетка)

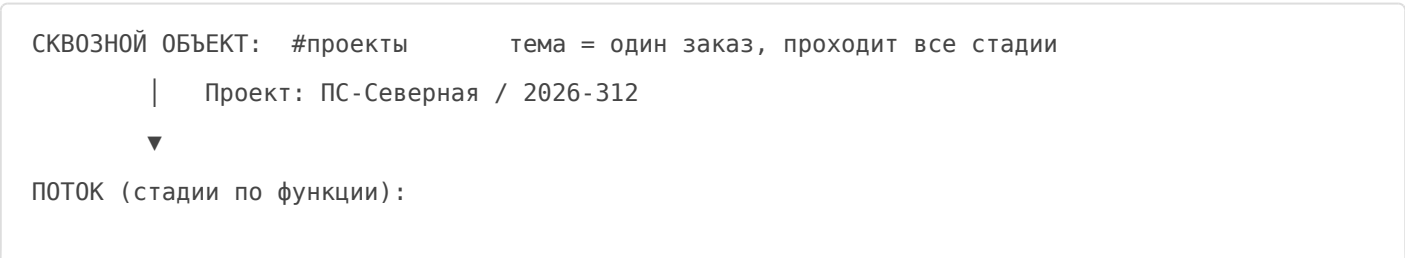
Приложение Б к РГ-СМК-Zulip-01.

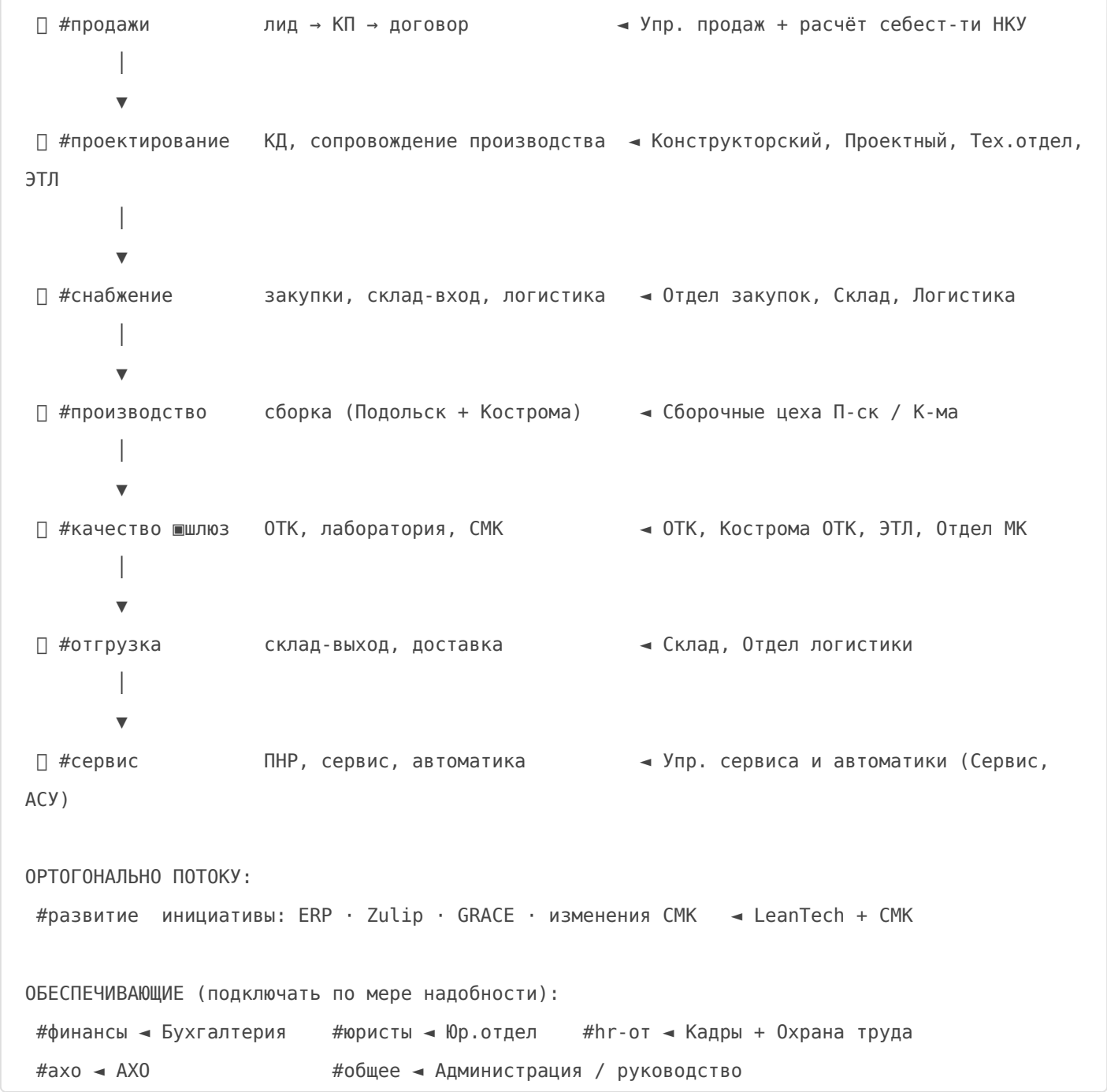
Верхнеуровневая (целевая) структура каналов для всей компании. Принцип: **каналы кладём на стадии потока создания ценности, а не на оргструктуру**. Отделы — участники (подписчики) стадийных каналов, а не отдельные каналы. Сквозной объект потока — заказ (**#проекты**) — вынесен в отдельный слой.

Почему не по оргструктуре

В компании ~30 активных подразделений. Если каждое = канал, поток (лид → расчёт → договор → КД → закупка → сборка → ОТК → отгрузка → ПНР), который пересекает почти все из них, окажется размазан по силосам и потеряет единое русло. Процессный взгляд = поток создания ценности = сквозные процессы, поэтому верхний уровень каналов повторяет **стадии потока**, а оргструктура проявляется ниже — через список участников каждого канала.

Схема





Стадийные каналы и участники

Канал (стадия)	Что в потоке	Отделы-участники
#продажи	лид → КП → договор	Управление продаж (Отдел продаж, по работе с клиентами, сопровождение); Отдел перспективных расчётов и оценки себестоимости НКУ

Канал (стадия)	Что в потоке	Отделы-участники
#проектирование	КД, разработка и сопровождение производства	Конструкторский отдел; Проектный отдел (НКУ / РУСН и подстанции); Технический отдел; Электротехническая лаборатория
#снабжение	закупки, входной склад, входная логистика	Отдел закупок; Склад; Отдел логистики
#производство	сборка изделий	Сборочный цех (Подольск); Кострома Сборочный цех; Кострома Технический отдел
#качество	контроль (шлюз потока) + система менеджмента	ОТК; Кострома ОТК; Электrolаборатория; Отдел менеджмента качества
#отгрузка	выходной склад, доставка клиенту	Склад; Отдел логистики
#сервис	пусконаладка, сервис, автоматизация	Управление сервиса и автоматики (Отдел сервиса, Отдел АСУ)

Один отдел может быть участником нескольких стадий — это нормально (Склад и Логистика входят и в #снабжение, и в #отгрузку). Отдел — участник, а не владелец канала.

Сквозной объект — #проекты

Это сердце процессного взгляда. В позаказном производстве ценность создаёт не отдел, а **заказ, который течёт через все стадии**. Поэтому заказ — первоклассная сущность:

- #проекты — одна тема = один заказ: Проект: <объект> / <№>. Здесь сквозная координация по конкретному заказу от расчёта до отгрузки и ПНР, со всеми задействованными отделами в одном месте.
- Стадийные каналы (#снабжение, #производство...) — для **повторяющейся, стандартной работы стадии**: мощности, нормы, типовые проблемы, улучшения стадии — то, что не привязано к одному заказу.

Это тот же принцип «поток (по заказу) vs функция (по стадии)», что и изменение-vs-эксплуатация, только масштабированный на всю компанию. Связка: из стадийного канала на заказ ссылаемся через #проекты>Проект: ...; итог по заказу фиксируем Решение: внутри его темы; рекламацию — Рекламация: в #качество со ссылкой на заказ. Сущностная модель и шлюзы — приложение В.

Две площадки — это измерение, а не вторая ветка

В оргструктуре есть неактивное «зеркало» Подольска (обособленное подразделение со своими отделами Закупок, Логистики, Склада, Бухгалтерии, Цифровизации, МК, Сборочным цехом...). **Не воспроизводить это как каналы.** Подольск и Кострома — одни и те же стадии потока на двух площадках, поэтому площадка = измерение темы (Площадка: Подольск / Площадка: Кострома) внутри #производство и #качество, а не отдельное дерево каналов. Иначе получишь дублирование всей структуры x2 и поток снова рассыпется.

Владельцы и старт

Владельцы:

- Zulip-админ, техплатформа и интеграция систем — LeanTech (партнёр по внедрению).
- Владелец #развитие — LeanTech + СМК; команда GRACE (отдел цифровизации АйТека, Калдарбеков) — продуктовая команда приложения GRACE, участник по своим инициативам.
- СМК-часть #качество — Отдел менеджмента качества.
- Каждый стадийный канал — руководитель ведущего отдела стадии.

Порядок ввода (по регламенту, фазами):

1. **Старт (фаза 1):** #общее, #проекты (заказы), #продажи, #качество, #развитие.
2. **Фаза 2 — стадии потока:** #проектирование → #снабжение → #производство → #отгрузка → #сервис.
3. **Фаза 3 — обеспечивающие:** #финансы, #юристы, #hr-от, #ахо — по мере реальной потребности.

“ Грамматика тем (закрытый список префиксов, правила Решение: / Рекламация: , индексация Опух) — приложение А (itech-zulip-ontology.md). Сквозной префикс — Проект: .

Приложение В: Zulip АйТек — сущностная модель и жизненный цикл проекта (заказа)

Приложение В к РГ-СМК-Zulip-01.

Дополнение к карте каналов. Здесь — скелет, на котором держится вся логика: знание о **контрагентах** и **объектах**, и проходящий через стадии **проект** (= заказ).

Терминология (важно)

Слово	Значение в АйТек	Где
Проект	заказ клиента — поставка НКУ/РУСН на конкретный объект, со своим жизненным циклом	сквозной объект потока, канал <code>#проекты</code> , тема <code>Проект: ...</code>
Развитие	внутренние инициативы и изменения: цифровизация, интеграция, изменения СМК	канал <code>#развитие</code> (<code>Инициатива: </code>), владелец — LeanTech + СМК

«Проект» закрепляем за заказом (язык бизнеса). Внутренние инициативы переименовываем в `#развитие`, чтобы одно слово не означало две разные вещи.

Сущностный спайн

Логика навешана не на стадии потока, а на сущности и их связи:

Контрагент —1:N→	Объект —1:N→	Проект (=заказ) —1:N→	Изделия
заказчик,	площадка/	конкретная	НКУ 0,4 кВ,
генподрядчик,	сооружение	поставка на объект	РУСН-10,
поставщик,	(ПС, завод,	со своим ЖЦ	подстанции
субподряд	ЖК, цех)	(G0...G8)	

Это граф, а не строгое дерево: у одного объекта может быть несколько контрагентов (заказчик, генподряд, проектировщик), а проект тянет ещё и поставщиков. Поэтому модель живёт как связанные записи, а не как вложенные папки.

Где что живёт (четыре слоя):

- **CRM/ERP** — источник истины по сущностям: карточки Контрагентов, Объектов, Проектов и связи между ними (граф сущностей).
- **KB (BookStack)** — источник истины по документам: регламенты, стандарты КД, методики; коммерческий раздел держит профили контрагентов и объектов как знание.
- **Zulip** — обсуждение, навешанное на сущности через карточку темы и нейминг. Чат не хранит сущности, он на них ссылается.
- **Onyx** — индексирует все три источника; сущности становятся **якорями retrieval**, поэтому «знание об объекте/контрагенте» отвечается поверх записей, документов и обсуждения.

Тема проекта в Zulip

Чтобы тема несла объект и контрагента (а не только номер), первое сообщение темы = карточка проекта со ссылками в структурный слой.

Имя темы (коротко, стабильно): Проект: <код объекта> / <№> Пример: Проект: ПС-Северная / 2026-312

Карточка (первое сообщение):

Контрагент:	000 «Ромашка» (заказчик) · ГК «Стройсервис» (генподряд)
Объект:	ПС 110/10 кВ «Северная», г. Подольск
Проект №:	2026-312 · Договор: №10/02
Состав:	НКУ 0,4 кВ — 12 шкафов · РУСН-10 — 3 ячейки
Срок:	отгрузка до 2026-09-15
Записи:	CRM/ERP ↔ <ссылка> · KB ↔ <ссылка>
Статус:	G2 — договор подписан, в проектировании

Эта карточка даёт и людям, и Опух три якоря сразу — контрагент, объект, проект — по любому из которых ищется контекст.

Жизненный цикл проекта (шлюзы)

Тема **Проект:** — сквозная, идёт через стадийные каналы. На каждом шлюзе статус обновляется в карточке; стандартная работа стадии остаётся в стадийном канале.

Проект: <Объект> / <№>		
G0	лид / запрос на объект	#продажи → квалификация
▼		
G1	расчёт себестоимости + КП	#продажи → отправлено клиенту
▼		
G2	договор подписан	#продажи → проект «в работу»
▼		
G3	КД готова	#проектирование → запуск в производство
▼		
G4	материалы обеспечены	#снабжение
▼		
G5	изделие собрано	#производство (Подольск / Кострома)
▼		
G6	ОТК / ПСИ пройдены ■шлюз	#качество → готово к отгрузке
▼		
G7	отгружено на объект	#отгрузка
▼		
G8	ПНР / сдача	#сервис → проект закрыт

Стадийный канал ведёт повторяющуюся работу стадии (мощности, нормы, типовые проблемы), а конкретный проект — сквозную тему; из стадии на проект ссылаемся через **#проекты>Проект: ...**. Решение по проекту фиксируем **Решение:** внутри его темы; рекламацию — **Рекламация:** в **#качество** со ссылкой на объект и проект.

Что это открывает (запросы к Onyx)

Сущности-якоря превращают разрозненную переписку в наведируемое знание:

- **По контрагенту:** «все проекты ООО Ромашка», «история рекламаций по этому заказчику».
- **По объекту:** «что у нас по ПС Северная за всё время», «какие изделия туда уходили».
- **По проекту:** «статус 2026-312», «что решили по составу щитов».
- **Срез по шлюзу:** «какие проекты застряли на G4 (снабжение)».

Без сущностного спайна эти запросы недоступны — есть только тысячи сообщений. Со спайном — это и есть «знание об объектах и контрагентах», ради которого всё строится.

Связь с базой знаний (KB)

Контрагент / Объект / Проект — это коммерческий раздел KB в действии: именно эти сущности делают базу знаний наведируемой, а не складом документов. Один словарь сущностей между CRM/ERP, KB и Zulip-темами — обязательное условие, чтобы Onyx видел единый граф, а не три разных языка для одного объекта.

“ Нейминг един во всём комплекте: Проект: = заказ (#проекты), Инициатива: = изменение (#развитие). Это приложение В; грамматика тем — приложение А (itech-zulip-ontology.md), карта потока — приложение Б (itech-zulip-channels-valuestream.md); ядро — регламент РГ-СМК-Zulip-01 (itech-zulip-reglament-start.md).

Приложение Г: Словарь — корпоративная база знаний (KB)

Каноническая страница базы знаний (KB). Приложение Г к РГ-СМК-Zulip-01.

Единая контролируемая лексика компании — один источник истины по терминам, на который ссылаются стандарт, приложения, темы Zulip и Onyx-ассистенты. Если термин встречается где-либо в работе, его определение — здесь.

Стек одним предложением: сущности — в CRM/ERP, документы — в KB (BookStack), обсуждение — в Zulip, поиск поверх всех — Onyx.

Слои и системы

- **KB (knowledge base) / корпоративная база знаний** — канонический слой документированного знания: регламенты, стандарты, методики. Источник истины по документам. Четыре раздела (контура): СМК/ISO 9001, инженерный, производственный, коммерческий.
 - **BookStack** — вики-платформа, на которой ведётся KB: полки → книги → главы → страницы; версии, поиск, перелинковка.
 - **Zulip** — корпоративный мессенджер; слой обсуждения и координации. Источник «что происходит и как мы к этому пришли».
 - **CRM / ERP** — слой сущностей и транзакций. Источник истины по контрагентам, объектам, проектам.
 - **ERP** — внедряемая система управления ресурсами; целевой носитель сущностной модели (текущая внутренняя инициатива).
 - **Onyx** — слой RAG-поиска поверх трёх источников (Zulip, KB/BookStack, CRM/ERP); даёт цитируемые ответы.
 - **GRACE** — собственное приложение АйТека (разрабатывает отдел цифровизации АйТека); интеграция со смежными системами — на стороне LeanTech.
-

Коммуникация (Zulip)

- **Канал (stream)** — устойчивая единица: кто участвует и кто видит; граница доступа и индексации.
 - **Тема (topic)** — единица работы/знания внутри канала; единица поиска в Опух.
 - **Сквозной канал** — следует за объектом потока (заказом) через все стадии (`#проекты`).
 - **Стадийный канал** — закреплён за стадией потока; ведёт повторяющуюся работу стадии.
 - **Префикс темы** — метка типа темы из закрытого списка: `Проект:` , `Инициатива:` , `Клиент:` , `Сделка:` , `Проблемный кейс` , `Регламент:` , `Аудит:` , `Несоответствие:` , `САРА:` , `Анализ рук-ва:` , `Решение:` , `Рекламация:` . Правила применения — приложение А.
-

Сущности

- **Контрагент** — заказчик, генподрядчик, проектировщик, поставщик или субподрядчик.
 - **Объект** — площадка/сооружение, для которого изготавливается оборудование (ПС, завод, ЖК, цех). Живёт долго, может собрать несколько проектов.
 - **Проект** (= заказ) — поставка оборудования на конкретный объект со своим жизненным циклом (G0-G8). В бизнес-языке АйТека «проект» = заказ.
 - **Изделие** — единица продукции в составе проекта: НКУ, РУСН, подстанция.
 - **Инициатива** — внутреннее изменение/улучшение (цифровизация, интеграция, изменения СМК). Не путать с проектом-заказом.
-

Поток и процесс

- **Поток создания ценности** — сквозной маршрут заказа от запроса до сдачи; основа архитектуры каналов.
- **Стадия потока** — этап маршрута: продажи → проектирование → снабжение → производство → качество → отгрузка → сервис.
- **Шлюз (G0-G8)** — контрольная точка перехода проекта между стадиями: G0 запрос, G1 расчёт+КП, G2 договор, G3 КД, G4 снабжение, G5 сборка, G6 ОТК/ПСИ, G7 отгрузка, G8 ПНР/сдача.
- **Карточка проекта** — первое сообщение темы `Проект:` с контрагентом, объектом, составом, сроком, статусом и ссылками в CRM/ERP и KB.
- **Площадка** — производственная локация (Подольск / Кострома). Измерение темы (`Площадка:`), а не отдельная ветка каналов.

Доменные термины (НКУ-производство)

- **НКУ** — низковольтное комплектное устройство.
 - **РУСН** — распределительное устройство собственных нужд. *[Уточнить каноническую расшифровку — в оргструктуре фигурирует и «РУСН», и «распределительные устройства среднего напряжения».]*
 - **ПС** — подстанция.
 - **КД** — конструкторская документация.
 - **КП** — коммерческое предложение.
 - **ПНР** — пусконаладочные работы.
 - **ПСИ** — приёмо-сдаточные испытания.
 - **ОТК** — отдел технического контроля.
 - **ЭТЛ** — электротехническая лаборатория.
 - **СМК** — система менеджмента качества.
 - **ISO 9001** — международный стандарт системы менеджмента качества.
 - **САРА** — корректирующие и предупреждающие действия (corrective and preventive actions).
-

Как пользоваться

- Это единственный источник истины по терминам. Любое расхождение в значении решается правкой здесь, а не локальной трактовкой.
- Изменение термина проводится как **Инициатива:** в **#развитие**; утверждённая формулировка обновляется на этой странице.
- На словарь ссылаются стандарт (РГ-СМК-Zulip-01), приложения А-В, описания каналов и Опух-ассистенты.
- Владелец — руководитель СМК; ведение — совместно с LeanTech.

Приложение Д:

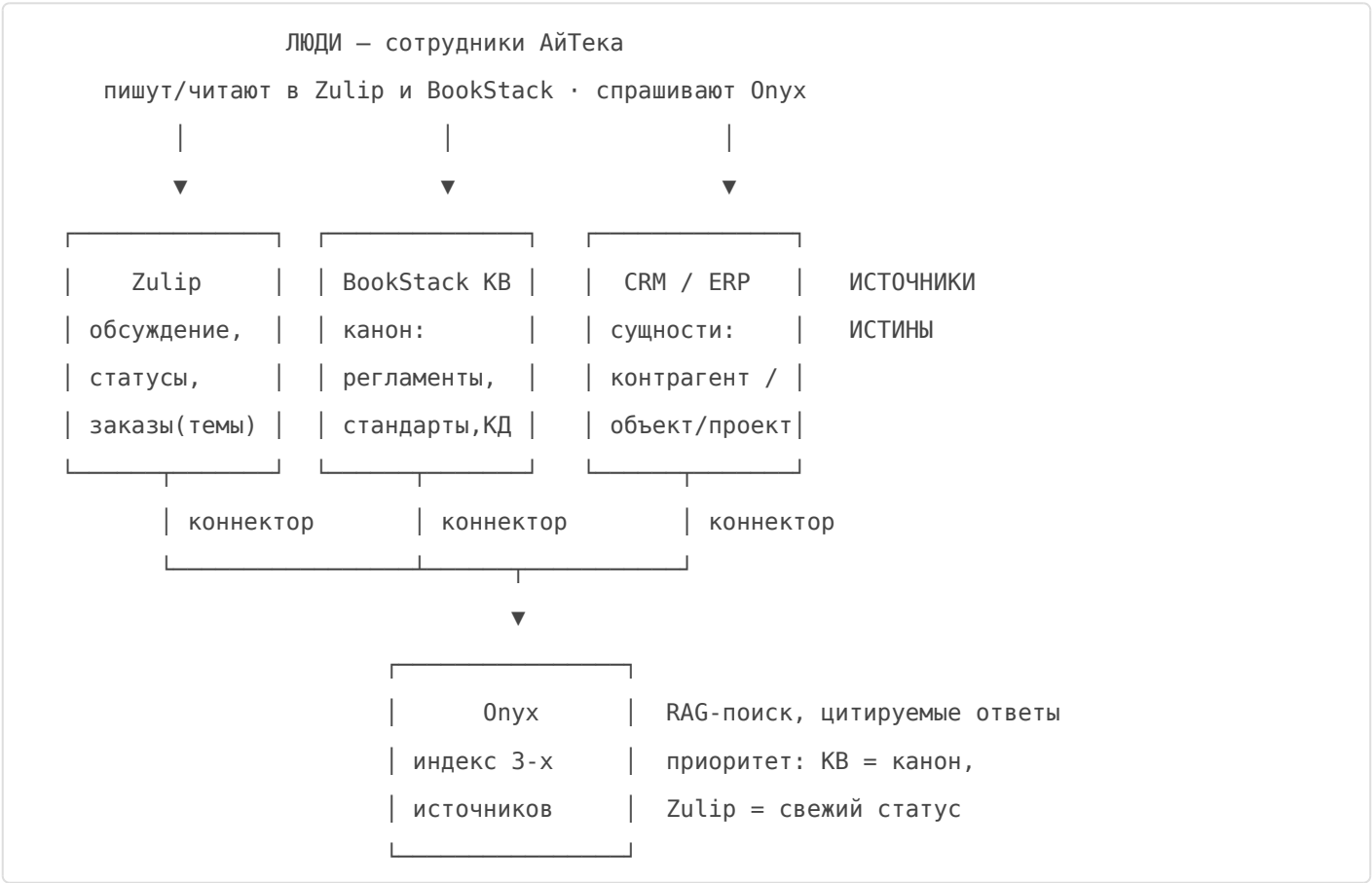
Архитектура и связи

Приложение Д к РГ-СМК-Zulip-01.

Карта системного ландшафта: какие системы за что отвечают, что куда передаётся и каким механизмом. Термины — в словаре (приложение Г).

Стек одним предложением: сущности — в CRM/ERP, документы — в KB (BookStack), обсуждение — в Zulip, поиск поверх всех — Онух; склейка событий — через n8n.

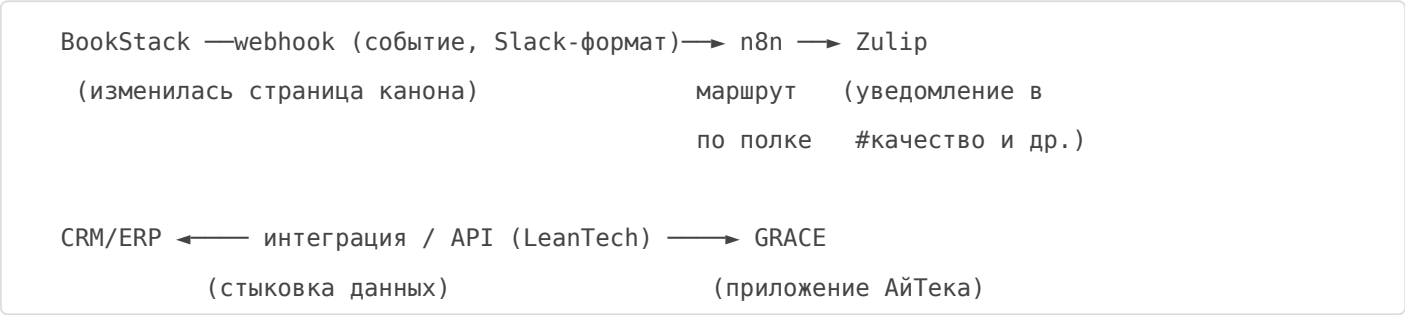
1. Слои и поиск



Каждый слой — единственный источник истины в своей зоне: **Zulip** = что происходит и как пришли к решению; **KB (BookStack)** = что является истиной (документы); **CRM/ERP** =

сущности и транзакции. Onyx ничего не хранит сам — он индексирует все три и отвечает поверх них.

2. Интеграции (событийная склейка)



n8n — шина-оркестратор: принимает события, маршрутизирует и форматирует. GRACE — приложение АйТека; его стыковку с данными (CRM/ERP) ведёт LeanTech.

3. Связи — что куда и как

Откуда → Куда	Механизм	Что передаётся	Отвечает
Zulip → Onyx	коннектор Zulip (бот)	темы/сообщения (обсуждение)	LeanTech
BookStack → Onyx	коннектор BookStack	канон-страницы	LeanTech
CRM/ERP → Onyx	коннектор / API	сущности	LeanTech
BookStack → Zulip	webhook → n8n → incoming webhook	уведомления об изменении канона	LeanTech
CRM/ERP ↔ GRACE	интеграция / API	данные (стыковка)	LeanTech
Zulip ↔ CRM/ERP	ссылки в карточке проекта (+ API)	связь темы-заказа с записью	LeanTech + Продажи
Onyx → Люди	чат / поиск	цитируемые ответы	—

Граница доступа = граница индексации: коннектор видит только то, что разрешено его токену. Чувствительное (цены, переговоры, закрытые полки KB) — не индексируется либо идёт в отдельный document set.

4. BookStack ↔ Zulip — how-to

BookStack умеет отправлять исходящие вебхуки в Slack-совместимом формате (Zulip их принимает напрямую), и их можно фильтровать по событию аудит-лога. Два варианта:

- **Просто:** вебхук BookStack → напрямую в Zulip incoming webhook URL → одна тема в одном канале. Без маршрутизации.
- **С маршрутизацией (рекомендуется):** вебхук BookStack → n8n → нужный канал/тема в зависимости от полки. n8n даёт фильтр, формат и роутинг.

Настройка:

1. В Zulip: создать incoming webhook bot, сгенерировать URL на нужный канал/тему (или направить на n8n).
2. В BookStack (`Settings → Webhooks`): endpoint = URL n8n (или Zulip напрямую); выбрать события (создание/обновление страницы), не «все подряд».
3. Включить асинхронную очередь BookStack (`bookstack-queue.service`), чтобы вебхуки не тормозили интерфейс.
4. В n8n: ветка по полке/книге → формирование сообщения → отправка в Zulip incoming webhook.

Маршрутизация (полка → канал):

Событие в BookStack	Куда в Zulip
Полка «СМК»: страница создана/обновлена	<code>#качество</code> > тема <code>База знаний: обновления</code>
Полка «Инженерный»: страница создана/обновлена	<code>#проектирование</code> > <code>База знаний: обновления</code>
Полка «Коммерческий»: страница обновлена	<code>#продажи</code> > <code>База знаний: обновления</code>
Любая полка: удаление страницы	<code>#общее</code> (или служебный лог)

Уведомления держать тихими: только создание/обновление страниц, не каждый промежуточный сейв. Обсуждение изменения — в Zulip под `Регламент:`, со ссылкой на страницу KB; сам документ остаётся каноном в BookStack.

5. Индексация Онух

- **Источники:** Zulip (обсуждение), BookStack/KB (канон), CRM/ERP (сущности).
- **Приоритет в ответах:** KB — для «что является истиной», Zulip — для «что нового по статусу».

- **Якоря retrieval:** сущности (контрагент/объект/проект) и префиксы тем — по ним строятся ассистенты и срезы.
 - **Доступ:** индексируется то, что видит токен коннектора; закрытые полки/каналы — не подписывать или отдельный document set.
-

6. Ключевые потоки

1. **Обновление канона.** Правка страницы в KB → webhook → n8n → уведомление в `#качество`; Онух переиндексирует страницу. Канон один, в BookStack.
 2. **Решение → норма.** `Решение:` в Zulip, если становится правилом, оформляется страницей в KB и дальше живёт каноном.
 3. **Сквозной запрос.** Вопрос в Онух → ответ собирается из KB (канон) + Zulip (статус) + CRM/ERP (сущность), с цитатами.
 4. **Жизнь заказа.** Сущность в CRM/ERP → карточка `Проект:` в Zulip ссылается на запись → статус по шлюзам G0-G8 → применимые стандарты подтягиваются из KB.
-

7. Границы и ответственность

- **LeanTech (партнёр по внедрению):** платформа Zulip, KB/BookStack, Онух, n8n; все коннекторы и интеграции; стыковка GRACE.
- **Отдел менеджмента качества (СМК):** владелец процесса и словаря; контроль соблюдения; раздел СМК в KB.
- **Отдел цифровизации АйТека:** разработка приложения GRACE.
- **Ведущие отделы стадий:** наполнение своих каналов и разделов KB.

Полная матрица — регламент, п. 6.

Связанные документы

- `itech-zulip-reglament-start.md` — ядро (стандарт).
- `itech-zulip-ontology.md` — приложение А: грамматика тем.
- `itech-zulip-channels-valuestream.md` — приложение Б: карта потока.
- `itech-zulip-project-entity-model.md` — приложение В: сущностная модель.
- `itech-glossary.md` — приложение Г: словарь.

Приложение Е: Структура базы знаний (BookStack)

Раскладка KB в трёх ярусах: **содержание** (что знаем) → **управление** (как организуем) → **инфраструктура** (на чём работает). Это рабочий чертёж для наполнения BookStack.

“ **Объектная основа.** Книги яруса «Содержание» — не папки отделов, а **проекции бизнес-объектов** (CoreStream A1–A11). Данные Grace = *зёрна* (экземпляры записей); объектная модель = *контекст*, из которого разворачиваются и **процесс** (методы объекта), и **метрики** (эконометрика по фазам жизненного цикла). BookStack фиксирует контекст-знание, Grace держит данные, агент соединяет их через теги `объект-с:` и `фаза:` .

Три яруса

СОДЕРЖАНИЕ (что знаем)	→ 4 контурные полки, книги = блоки БЗ-1...11 = проекции объектов
УПРАВЛЕНИЕ (как организуем)	→ полка «Стандарты и управление знаниями» (стандарт · ведение · зрелость)
ИНФРАСТРУКТУРА (на чём)	→ полка «Цифровая инфраструктура» (Zulip · BookStack · Onyx · n8n · GRACE · CRM/ERP)

Разрез повторяет принцип «порядок → инструменты»: содержание и управление тул-независимы; инфраструктура — конкретные системы, реализующие порядок. Сменится инструмент — правится только инфра-книга, мета-полка не дрогнет.

Дерево

KB (BookStack)
— СОДЕРЖАНИЕ (книги = блоки знаний БЗ = проекции объектов A1–A11) —

- |
- | — Полка: СМК / ISO 9001 · объект A2 Бизнес-система
- | | — Книга: БЗ-1 Бизнес-процессы и СМК
- | | — Книга: БЗ-2 Записи качества
- |
- | — Полка: Инженерный · объект A3 Продукт
- | | — Книга: БЗ-3 Продукты и конфигурации ≠ Коммерческий
- | | — Книга: БЗ-4 Нормативная база
- |
- | — Полка: Производственный · объект A8 Товар
- | | — Книга: БЗ-5 Технологические стандарты
- | | — Книга: БЗ-6 База дефектов и решений ≠ СМК
- |
- | — Полка: Коммерческий · объект A7 Потребитель (компания)
- | | — Книга: БЗ-7 Клиентская база
- | | — Книга: БЗ-8 Конкурентная среда
- | | — Книга: БЗ-9 Коммуникации и скрипты
- | | — Книга: БЗ-10 Прецеденты проектов ≠ все контуры · объект «Здание»
- | | — Книга: БЗ-11 История сделок
- |
- | — УПРАВЛЕНИЕ (как организуем) —
- |
- | — Полка: Стандарты и управление знаниями
- | | — Книга: Рабочее пространство и знания – стандарт ← НАШ КОМПЛЕКТ
- | | | — Глава: Стандарт
- | | | | — Стр.: РГ-СМК-Zulip-01 – Регламент
- | | | — Глава: Приложения
- | | | | — Стр.: А – Грамматика тем и индексация Опух
- | | | | — Стр.: Б – Карта каналов (поток создания ценности)
- | | | | — Стр.: В – Сущностная модель и жизненный цикл
- | | | | — Стр.: Г – Словарь
- | | | | — Стр.: Д – Архитектура и связи
- | | | | — Стр.: Е – Структура базы знаний
- | | — Книга: Ведение базы знаний (как работаем в BookStack)
- | | — Книга: Модель зрелости управления знаниями
- | | | — Глава: Модель (инструмент LeanLab, переиспользуемый)
- | | | | — Стр.: Шкала уровней 0–5
- | | | | — Стр.: Организационная ось (зрелость контура)
- | | | | — Стр.: Объектная ось (зрелость блока)
- | | | | — Стр.: Анкета Э0 (методика замера)

└ Глава: Применение – Этап 0 (АйТек)

└ Стр.: Карта замера – 11 блоков (до / цель)

└ Стр.: Оценка контуров (до)

└ Стр.: Уровень → эффект (язык питча)

— ИНФРАСТРУКТУРА (на чём работает) —

└ Полка: Цифровая инфраструктура

└ Книга: Zulip (развёртывание, боты, интеграции)

└ Книга: BookStack (настройка, бэкап, вебхуки)

└ Книга: Опух (коннекторы, ассистенты, доступы)

└ Книга: n8n (сценарии)

└ Книга: GRACE (приложение АйТека, стыковка)

└ Книга: CRM / ERP (внедрение, интеграции)

📖 — книга стоит на нескольких полках (BookStack это позволяет). Многополочные: БЗ-3 (Инженерный + Коммерческий), БЗ-6 (Производственный + СМК), БЗ-10 (все контуры).

“ Полки — адаптируемы: по мере наполнения добавляем дома для остальных объектов (А5 Трудовые ресурсы → HR, А10 Капитал → Финансовый, А4 Активы → АХО, А6/А9 Снабжение и логистика).

Содержание: книги = блоки знаний = проекции объектов

Книги контурных полок — это 11 блоков знаний (БЗ-1...11), а не произвольный набор. Каждый блок — **проекция бизнес-объекта CoreStream**: он представляет один объект в контуре знаний, со своими атрибутами, фазами жизненного цикла и связями.

Полка (контур)	Книги	Объект CoreStream
СМК / ISO	БЗ-1 · БЗ-2	А2 Бизнес-система
Инженерный	БЗ-3 · БЗ-4	А3 Продукт
Производственный	БЗ-5 · БЗ-6	А8 Товар
Коммерческий	БЗ-7 · БЗ-8 · БЗ-9 · БЗ-11	А7 Потребитель (компания)

Полка (контур)	Книги	Объект CoreStream
(сквозная)	БЗ-10 Прецеденты проектов	Здание (объектный контур)

Каждый блок — самостоятельный актив; его зрелость измеряется моделью зрелости (см. полку «Управление»). Поэтому на контурную книгу вешаем теги `блок:`, `зрелость:` и `объект - cs:` — и карта замера становится живым срезом по структуре, а не отдельной таблицей.

«Здание» ≠ «Потребитель» (важный развод)

«Потребитель» в данных бывает трёх разных сущностей — их нельзя смешивать:

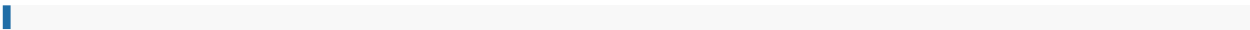
- **Компания** (`accounts`) = объект **A7 Потребитель**: у неё свой жизненный цикл продаж (потенциальный → горячий → покупатель → спящий), её квалифицируют, обслуживают, удерживают.
- **Здание** (`properties`) = **отдельный объект** (объектный контур): место, где стоит наше оборудование, — точка потребления, не покупатель. Знание о нём = прецеденты (БЗ-10).
- **Представитель** (`contacts`) = часть A7 (контактное лицо, интерфейс к компании), не самостоятельный объект.

Связывает их **Проект** (`projects`) — это **связь, а не объект**: несёт `account_id` (компания) + `property_id` (здание) + фазы. Прямого ребра «Здание → Компания» нет — только через Проект. Поэтому один объект-здание может быть связан со многими компаниями (проектировщик, генподрядчик, заказчик). Подробные карточки объектов — слой `corestream` (напр. `A7_potrebitel.md`).

Управление: наш комплект = 1 книга

`Рабочее пространство и знания – стандарт` — одна книга на полке «Стандарты и управление знаниями»: глава «Стандарт» (регламент) + глава «Приложения» (A–E). Название — про **порядок организации работы и знаний**; конкретные системы (Zulip, BookStack, Onyx) — лишь инструменты, поэтому их в названии нет. Реальный охват несут описание и теги `система:`.

Описание книги (поле description):



Стандарт организации работы, коммуникации и знаний вокруг потока создания ценности и сущностей (контрагент / объект / проект): что является каноном, что — обсуждением, где живут сущности и как это находится. Конкретные системы — лишь элементы, реализующие этот порядок: Zulip (обсуждение и координация), BookStack (канон и база знаний), Опух (поиск), CRM/ERP (сущности), n8n (склейка событий). Владелец — CMK, ведение — LeanTech. Код: РГ-CMK-Zulip-01.

Рядом на полке — ещё две книги: **Ведение базы знаний** (как именуем страницы, ведём теги, кто за какую полку отвечает) и **Модель зрелости** (диагностика «до/после»; глава «Модель» — переиспользуемый инструмент LeanLab, глава «Применение» — замер АйТека).

Словарь (приложение Г) — сквозной для всей KB; пока живёт страницей в книге стандарта. Если лексика станет общекорпоративной — выносим в отдельную книгу [Словарь](#) / [Справочник](#) на нескольких полках.

Жизненный цикл знания (как ведём KB)

Как знание живёт в KB — это цикл объекта **A11 «Знания»**: A11.1 Сбор → A11.2 Интеграция в процесс → A11.3 Применение → A11.4 Оценка → A11.5 Обновление → A11.6 Архивация. Этот цикл = операционная модель базы знаний и он же ложится на уровни зрелости (1→5): от «в головах» до «живой контекст AI-агента». Тег [статус:](#) (проект · действует · архив) отмечает положение страницы в этом цикле.

Граница «архитектура vs ранбуки»

Чтобы не задвоить: **архитектура (приложение Д)** — это карта и дизайн связей, она остаётся в книге стандарта («что с чем связано и почему»). Пошаговые ранбуки («как настроить вебхук, где бэкап, какие коннекторы») идут в инфра-книги соответствующих систем. Д — обзор, инфра-книги — глубина под ним; в стандарте — ссылка на них.

Схема тегов (контролируемые ключи)

Теги — сквозная метадата всей KB; ключи закрытые, как и префиксы тем.

Ключ	Значения	Назначение
раздел:	смк · инженерный · производство · коммерческий · управление-знаниями · инфраструктура	полка / ярус
блок:	бз-1 ... бз-11	блок знаний (для контурных книг)
объект-с:	а2 · а3 · а7 · а8 · а11 · здание	бизнес-объект CoreStream (книга-проекция) — мостик к Onyx/Grace по объекту
фаза:	а7.1 ... а7.6 (и фазы др. объектов)	фаза жизненного цикла объекта → срезы воронки и эконометрика
зрелость:	0 · 1 · 2 · 3 · 4 · 5	уровень зрелости (из модели зрелости)
тип:	стандарт · регламент · приложение · методика · шаблон · запись · словарь · ранбук	тип документа
система:	zulip · bookstack · onyx · crm-erp · n8n · grace	затрагиваемые системы (многозначный)
статус:	проект · действует · архив	жизненный цикл (фаза А11)
владелец:	смк · leantech · продажи · ...	ответственный
код:	напр. РГ-СМК-Zulip-01	код документа
объект: · контрагент:	напр. РС-Северная · Ромашка	якоря экземпляров (конкретное здание / компания) — мостик в CRM/ERP

Различать: объект-с: = тип объекта в модели (А7, здание); объект: / контрагент: = конкретный экземпляр (РС-Северная, Ромашка). Первое связывает с моделью и эконометрикой, второе — с записью в Grace.

Теги книги стандарта и страниц

Книга «Рабочее пространство и знания — стандарт»:

раздел: управление-знаниями

тип: стандарт

система: zulip · bookstack · onyx · crm-erp

статус: проект

владелец: смк · ведение: leantech

код: РГ-СМК-Zulip-01

Страницы (стандарт + приложения):

Страница	тип:	система:
РГ-СМК-Zulip-01 — Регламент	регламент	zulip
А — Грамматика тем	приложение	zulip · onyx
Б — Карта потока	приложение	zulip
В — Сущностная модель	приложение	crm-erp · zulip
Г — Словарь	словарь	— (сквозной)
Д — Архитектура и связи	приложение	zulip · bookstack · onyx · crm-erp · n8n
Е — Структура базы знаний	приложение	bookstack

Контурные книги (БЗ) получают `раздел:`, `блок: бз-N`, `объект-cs: aN`, `зрелость: N`; контентные страницы дополнительно — `фаза:`, `объект:` / `контрагент:`. Инфра-книги — `раздел: инфраструктура`, `тип: ранбук`, `система: <своя>`.

Связи (из приложений А, Д)

- **Zulip ↔ страница.** Канон-документ = страница KB. В Zulip под `Регламент:` идёт обсуждение со ссылкой на страницу; страница ссылается назад на пред-обоснование.
- **Вебхук по полке → канал.** Изменение страницы → n8n → уведомление в профильный канал (полка «СМК» → `#качество`). Детали — приложение Д.
- **Онух по полке/тегам.** Полка = document set; теги (`блок:`, `объект-cs:`, `фаза:`, `зрелость:`, `система:`) = фильтры. На них строятся ассистенты и срезы («что ≤2 по зрелости», «знание по объекту А7 фазы А7.3»). Чувствительные полки — ограниченный доступ или не индексировать.

Связанные документы

- `itech-zulip-reglament-start.md` — ядро (стандарт).
- `itech-zulip-ontology.md` — приложение А: грамматика тем.
- `itech-zulip-channels-valuestream.md` — приложение Б: карта потока.
- `itech-zulip-project-entity-model.md` — приложение В: сущностная модель.
- `itech-glossary.md` — приложение Г: словарь.
- `itech-architecture.md` — приложение Д: архитектура и связи.
- Модель зрелости — по материалам LeanLab KM Maturity Model (отдельная книга на мета-полке).
- Карточки объектов (слой `corestream`): `00_object_map.md`, `A7_potrebitel.md` — объектная модель, питающая структуру и теги.