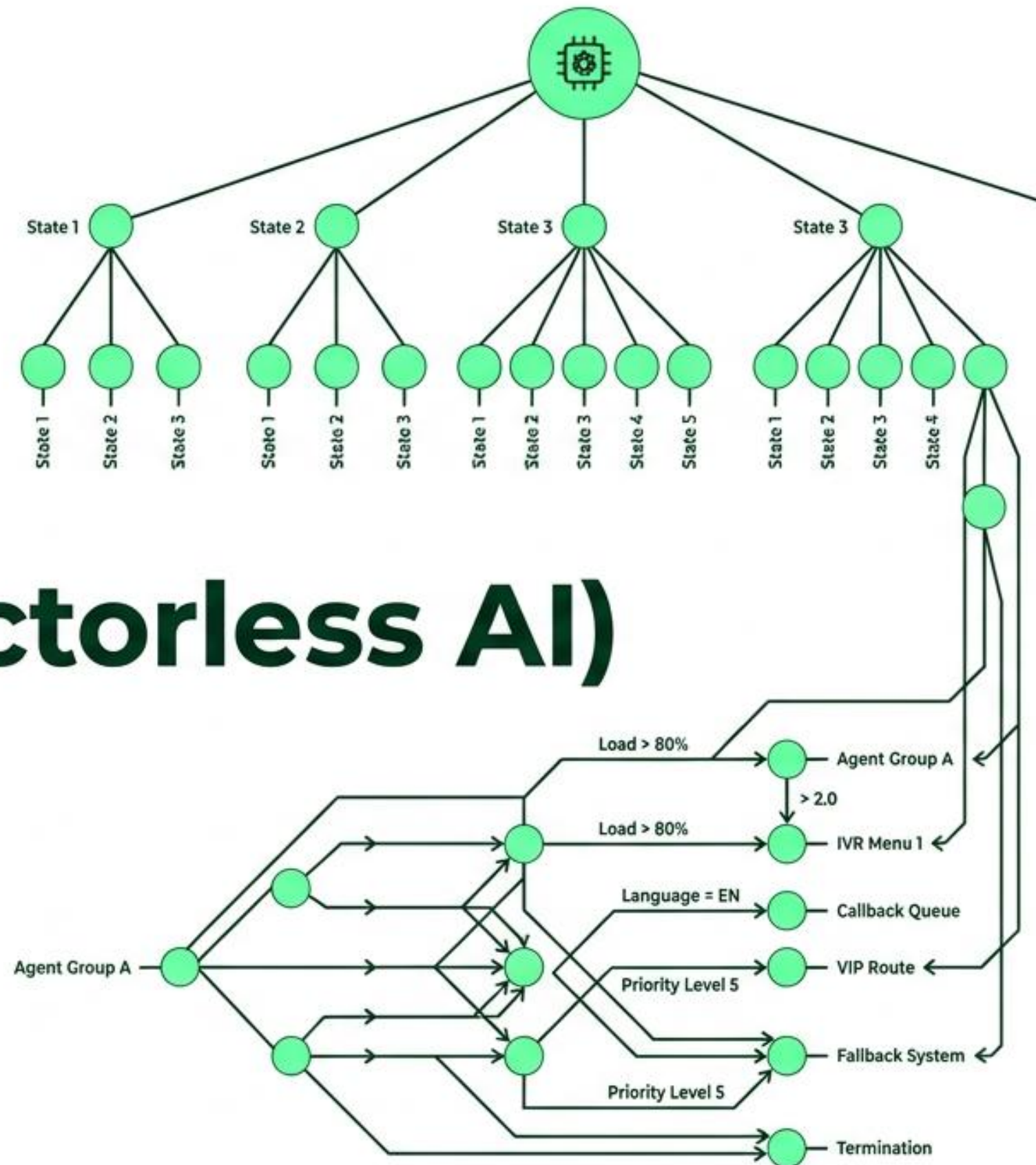


Почему будущее Enterprise AI — «безвекторно» (Vectorless AI)

Детерминированный голосовой роутинг в HighLoad-телефонии





Exifan

ENTERPRISE PROFILE // AI ARCHITECT



Андрей Носов



Chief AI Architect в RAFT



**Руководитель школы
AI-архитекторов OTUS**



PhD Tampere University



Канал @AI_ARCHNADZOR



andrey.nosoff@gmail.com

План

1. Постановка проблемы и критерии успеха в телефонии.

2. Разбор Бейзлайна (Vector RAG) и гипотез.

3. Как Vectorless решает проблему Realtime (разбор бюджета задержки).

4. Специфика телефонии: Yandex Realtime API, Q.931, AMR и Grammar Decoding.

5. Реальность продакшена: STT-шум, HITL-аудит, TCO, бэклог и матрица применимости.

Цели бизнеса

Кейс



1. МАСШТАБ СЕТИ

50+ отелей



2. КОНТАКТ-ЦЕНТР

150 операторов



3. ЦЕЛЕВОЙ SLA

0% ошибок

Критерии успеха: Target KPIs & SLA

100%

Accuracy: 100% точность в финансовых цифрах. 0% галлюцинаций по штрафам.

< 400 мс

Realtime SLA: p50 < 400 мс, p99 < 700 мс от конца реплики до ответа TTS.

30 / 5

Throughput: 30 соединительных линий / 5 CPS - Calls Per Second.

> 80%

Conversion: > 80% успешно завершенных целевых диалогов без оператора.

Архитектура

PDF-регламенты:

Динамические
невозвратные тарифы,
таблицы штрафов,
сноски.

Bitrix24[©]

CRM Webhooks:

Триггеры из
Битрикс24 (ID
брони, сумма,
дата).

Yandex

RTP Audio Stream:

Стриминг аудио
через Yandex
Realtime API
(WebSocket).

AI System

CRM Action:

Перевод сделки,
высылка SMS
с эквайрингом или
программная
аннуляция.

TTS-Шаблон:

Динамический
синтез
утвержденного
юристами текста.

Бейзлайн

Наш Бейзлайн: Нативный Vector RAG

Тест на 100 вопросах по тарифам

Стек бейзлайна

LangChain + Qdrant +
text-embedding-3-large +
Qwen-3.5-72B-Instruct /
GPT-5.4

Результаты

Precision@1: 52%

Table Accuracy: 38%

ALARM STATE

Financial Hallucination Rate: 24% —
недопустимо для прода

ALARM STATE

Latency: 3.2 секунды — неприменимо
для телефонии

Ограничения классических подходов: "Эффект shreddера" в Vector RAG



Иллюзия семантического поиска

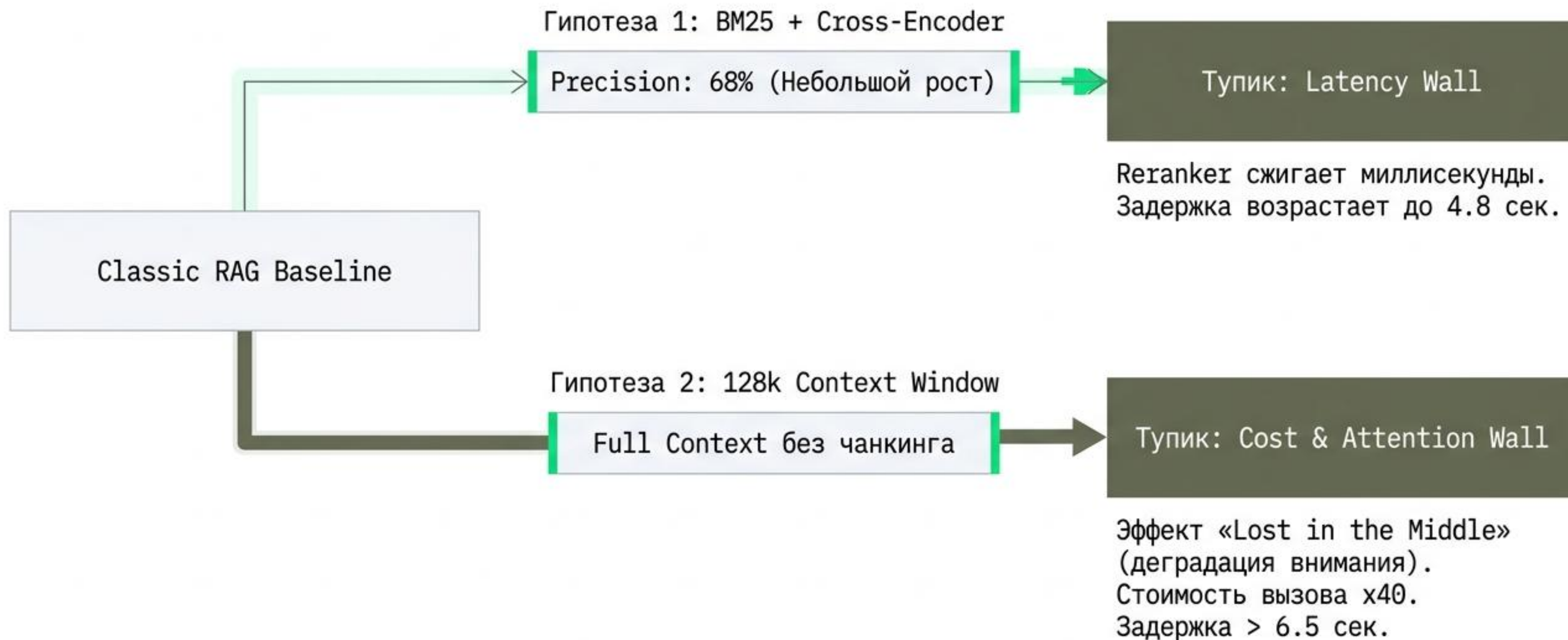
Фактическая релевантность

Similarity $\neq$ Relevance

Близость 98%, Релевантность 0%
Запросы: "Q3 2026" vs "Q4 2023"
Вывод: Косинусная близость дает 98% сходства из-за общего словаря, но для бизнеса фактическая релевантность равна 0%.

Семантическая близость (Cosine Similarity)

Попытки спасти RAG: Почему классика не работает в Realtime



Смена парадигмы

Парадигма Vectorless AI

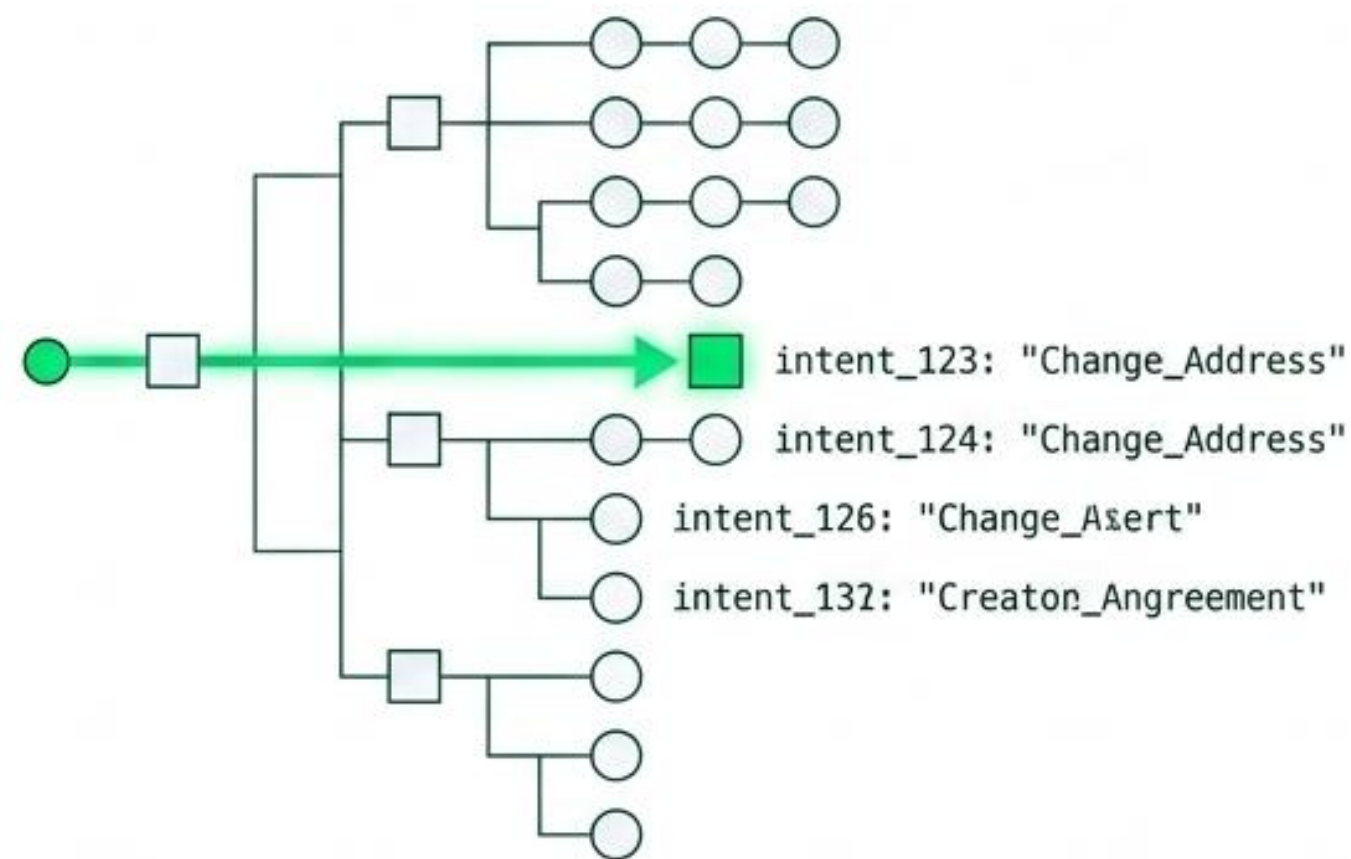
Отказ от векторов в рантайме. LLM работает исключительно как Zero-Shot роутер.

Обычный RAG



Тяжелый поиск по векторам прямо во время звонка. Медленно, дорого, вероятно.

Vectorless AI



- **Offline:** Индексация сложных PDF-регламентов в детерминированное JSON-дерево (PageIndex).
- **Online (В момент звонка):** LLM не читает и не ищет текст. Она лишь определяет ID нужного интента.

Практика: Архитектура PageIndex в оперативной памяти

```
{
  "doc_id": "hotel_tariffs_2026",
  "nodes": {
    "node_001": {
      "title": "Невозвратный тариф",
      "summary": "Штраф 100% при аннуляции менее чем за 3 дня",
      "children": ["node_001_table"],
      "raw_text_pointer": "ram://tariffs/non_refundable_raw.txt"
    }
  }
}
```

Структура, которая позволяет LLM принимать решения за миллисекунды без необходимости читать исходный текст

План масштабирования

1. Модульная компиляция (Forest of Trees)

Master Router



Домен → поддерево

Tariffs_Tree.json

SPA_Tree.json

Dining_Tree.json

Каждый регламент — изолированное поддерево; модель работает только с нужным.

2. Разделение статики и динамики (Static vs Dynamic)

ГРАФ – ТОЛЬКО СТАТИКА

Условия брони

Часы работы

Состав услуг



ДИНАМИКА – API-ВЫЗОВ ПО ID

PMS: Opera / Fidelio

Цена номера сегодня • слоты массажа

3. Каскадный двухуровневый спуск (Multi-hop Hierarchy)

Уровень 1 · Классификатор



DOM_SPA



Уровень 2 · Ветка СПА



Процедура

Окно контекста не переполняется

4. Контекстное отсечение подграфа (Subtree Pruning)

Триггер звонка: «Неоплаченная бронь»



DOM_TARIFFS ✓

~~DOM_SPA~~

~~DOM_TRANSFER~~

В контекст — только DOM_TARIFFS: ветки бассейна и трансфера не расходуют память GPU.

Конвеер Vectorless AI

☰ ФАЗА 1: OFFLINE INGESTION (ВНЕ ЗВОНКА)

- 1 PDF-регламент отеля → коммит в CI/CD пайплайн
- 2 VLM-парсинг: заголовки • таблицы • сноски
- 3 lat.md: линтинг ссылочной целостности графа
- 4 Юрист + финдиректор → заверяют schema.json

🕒 2–3 минуты • раз в релиз • на задержку звонка не влияет

Конвейер Vectorless AI

⚡ ФАЗА 2: ONLINE RUNTIME & HOT-RELOAD (В ЗВОНКЕ)

Граф в RAM

компактный Python dict • 0.8 МБ



Выборка узла

0.05 мс в памяти процесса

Векторная БД полностью отключена • 0 мс на эмбединги

Смена тарифа посреди дня → reload ветки в Redis/RAM за 12 мс • без перезапуска vLLM и переобучения

Результаты

Бенчмарк

	Baseline (Vector RAG)	H1	H2	Vectorless
 Точность (Accuracy)	52%	68%	61%	98.7%
 Финансовые галлюцинации	24%	12%	18%	0% (Fail-Safe)
 Задержка p50 (Latency)	3.2 сек	4.8 сек	6.5 сек	< 380 мс
 Расход токенов на такт	~12,000	~15,000	~85,000	~200

Vectorless AI полностью удовлетворяет жестким SLA телефонии (0% галлюцинаций, p50 < 400 мс)

Безопасность

БАНК РОССИИ (ЦБ РФ)

Критичный комплаенс

Рекомендации по управлению рисками применения ИИ (2025–2026)

Запрет вероятностного поведения в транзакциях. Все финансовые и скоринговые решения робота должны иметь 100% воспроизводимый след аудита.

Решение в Vectorless: Vectorless AI логирует цепочку ID узлов графа: [node_tariff -> branch_standard -> rule_cancel].

ГОСТ Р ИСО/МЭК (РФ)

Стандарты надежности РФ

ГОСТ Р ИСО/МЭК 42001-2024 & ГОСТ Р 53844

Национальные стандарты систем менеджмента искусственного интеллекта и качества телекоммуникационных услуг связи в РФ.

Решение в Vectorless: Детерминированный обход дерева исключает дрейф признаков и неконтролируемые галлюцинации.

ЗАКОНОДАТЕЛЬСТВО РФ

Юридическая ответственность

Закон «О защите прав потребителей» (ст. 32 ЗоЗПП РФ)

Потребитель имеет право отказаться от договора в любое время при условии оплаты фактически понесенных расходов (ФПР).

Решение в Vectorless: Fail-safe модуль блокирует списание штрафа 100% при наличии уважительных причин из сносок.

МИНЦИФРЫ РФ

Критичный суверенитет

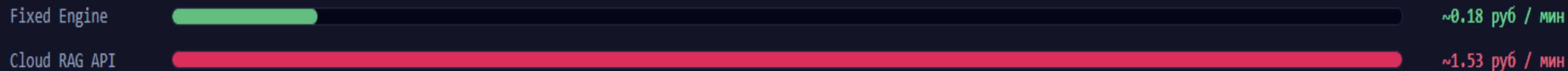
152-ФЗ «О персональных данных» и суверенный контур

Строгая локализация биометрических и персональных данных абонентов в границах РФ без передачи в зарубежные облачные API.

Решение в Vectorless: Полностью суверенный контур: локальный vLLM на собственном GPU-сервере в РФ без утечки голосовых данных.

Экономика

СЕБЕСТОИМОСТЬ МИНУТЫ



↗ Годовая экономия Enterprise-кластера телефонии:

+26 000 000 ₽ / год

Лайфхаки

Фильтрация краевых телеком-кейсов до LLM



LLM вызывается только тогда, когда на линии подтвержденный живой человек.

Узкие места архитектуры и их купирование

STT-шум

Проблема: Потеря частицы «НЕ».

Решение: Купируется связкой STT Confidence Score + Fail-Safe Python Script.

Multi-Intent (Barge-In)

Проблема: 3 несвязных вопроса в одной фразе («Оплачу завтра, а бассейн с подогревом?»).

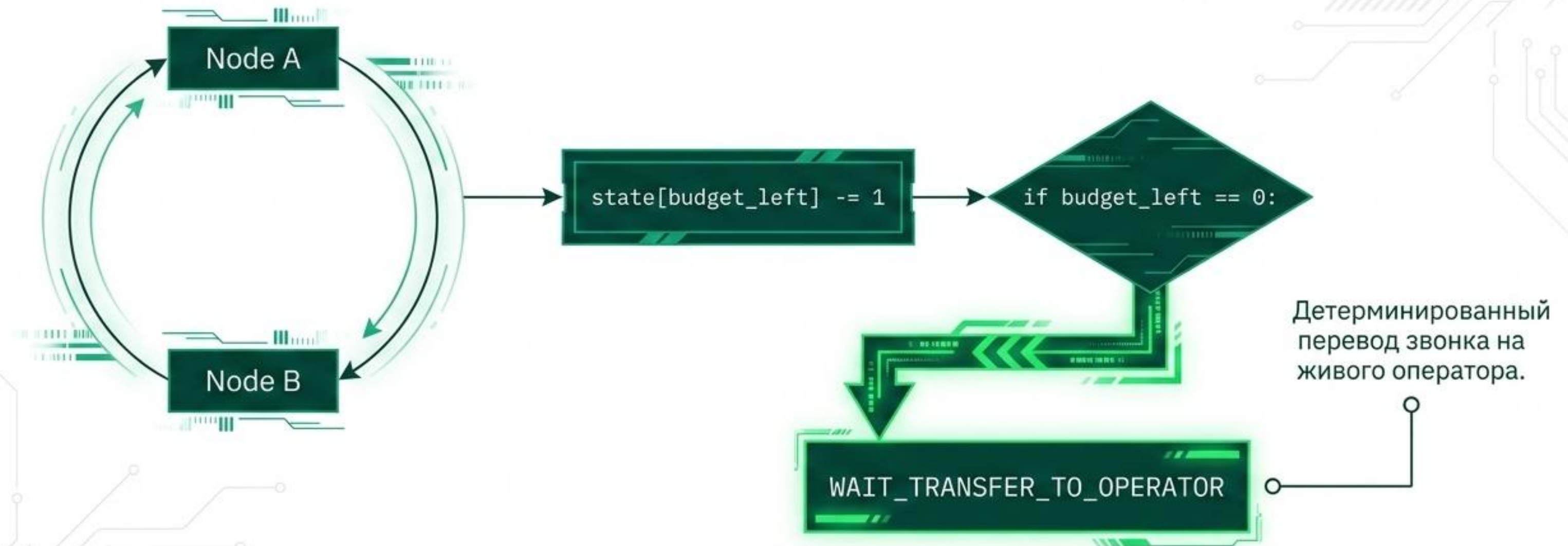
Решение: Модель отдаёт WAIT_TRANSFER_TO_OPERATOR. Лучше перевести 15% звонков на человека, чем заставить бота нагаллюцинировать.

Перебивание (Barge-In)

Проблема: Асинхронность речи.

Решение: Требуется мгновенной отмены рендеринга TTS по событию WebSocket.

Traversal Budget: Защита от зацикливания



Если нужный узел (ответ) не найден за 2 перехода, система принудительно прерывает работу LLM. Стопроцентная защита от зависаний на линии.

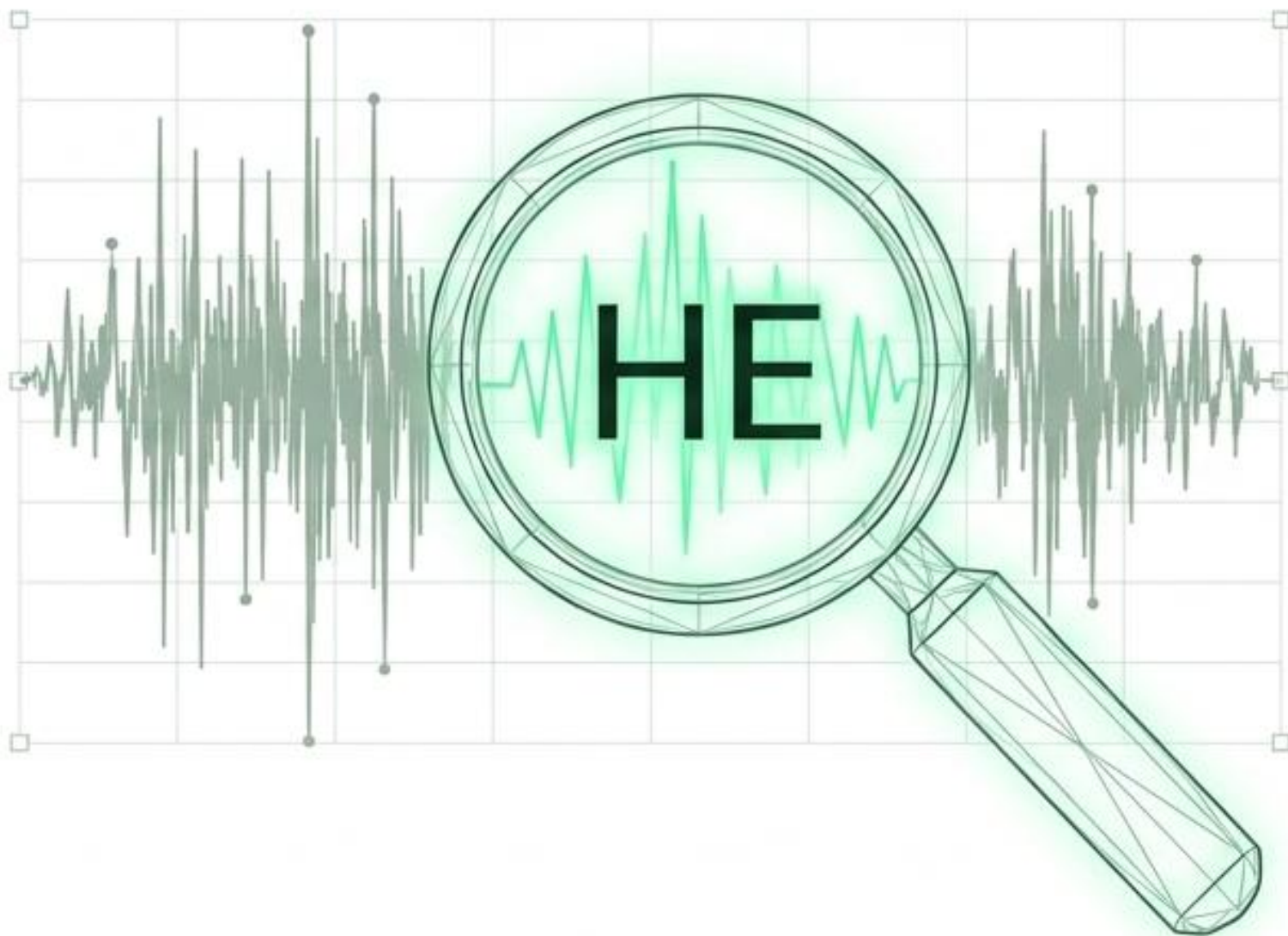
Генерация с ограничениями: vLLM + outlines

```
from Enum import Enum
import outlines

class CallIntent(str, Enum):
    CONFIRM = "confirm_amount"
    DEFERRAL = "deferral_salary"
    REFUSAL = "explicit_refusal"
    FAKE_HUMAN = "fake_human"

### vLLM v0.10+ с ограничением токенов через BNF-грамматику (35 мс на GPU)
model = outlines.models.vllm(
    "Qwen/Qwen3.6-27B-Instruct", quantization="fp8", tensor_parallel_size=1
)
generator = outlines.generate.json(model, CallIntent)
intent = generator("Ой, я сейчас в Дубае, карточка не работает")
### Output: "deferral_salary"
```

Защита от потери частицы «НЕ» в шумном тракте

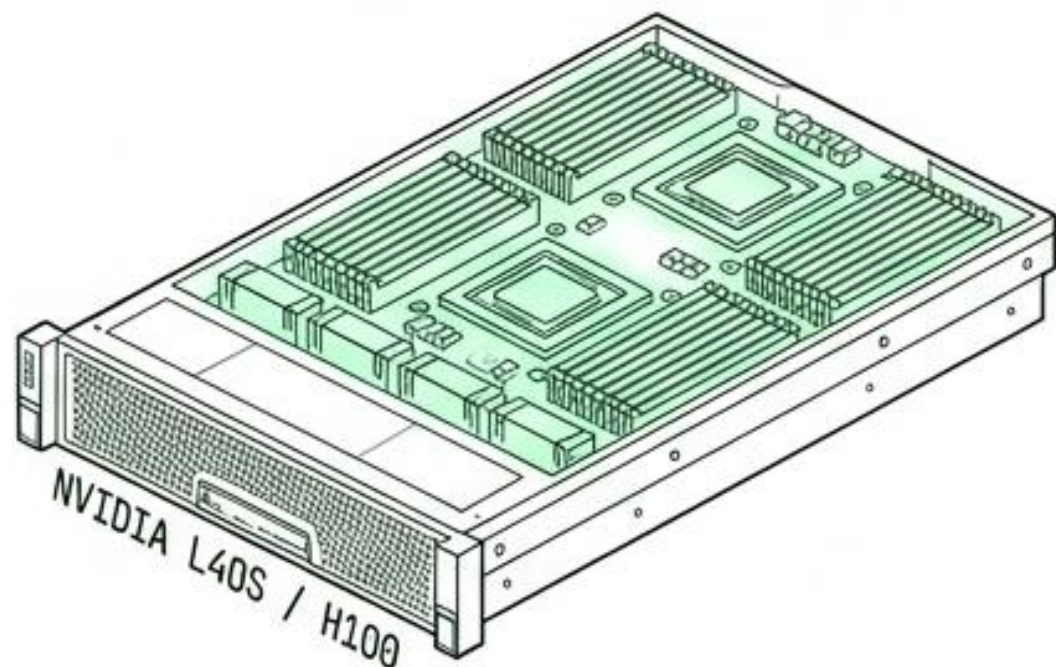


```
def verify_cancellation_safety(
    transcript: str, predicted_intent: str, stt_confidence: float
) -> str:
    """
    Защита от ошибки STT: если канал шумный и уверенность
    распознавания < 85%, или обнаружены маркеры отрицания
    -> блокировка отмены брони.
    """

    negation_markers = ["не", "нет", "пока не", "не уверен",
                        "не отменяйте"]
```

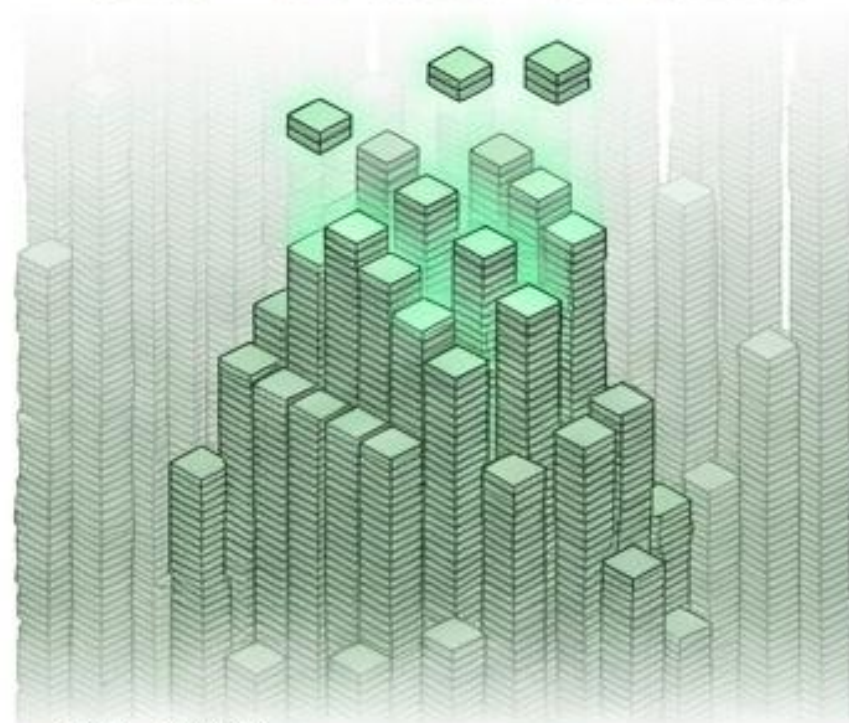
Экономика транзакции: GPU против Cloud API

The Fixed Engine



~\$3,000 / мес (Аренда 1x GPU под vLLM).
Стратегия: Утилизация простоя вне пиковых часов обзвона + Prefix Caching.

The Token Drain



15,000 токенов за такт
(Тяжелые Cloud API / RAG).
Оплата за каждое слово.

**Итоговая стоимость транзакции на 60% ниже,
чем постоянная прокачка через Cloud API.**

Blueprint

Матрица применимости Vectorless AI



Идеально:

Регламенты, юридические договоры, финансовые тарифы, API-спецификации, телефония, финтех.

Требование: Где нужна строгая иерархия и 0% ошибок.



Не подходит:

Поиск по неструктурированным логам, аналитика свободных отзывов, Open-Domain Q&A.

Пошаговый алгоритм внедрения Vectorless-архитектуры

1. Ingestion: Спарсите PDF в JSON-дерево (PageIndex) через VLM.

2. Audit: Проведите HITL-аудит графа аналитиками.

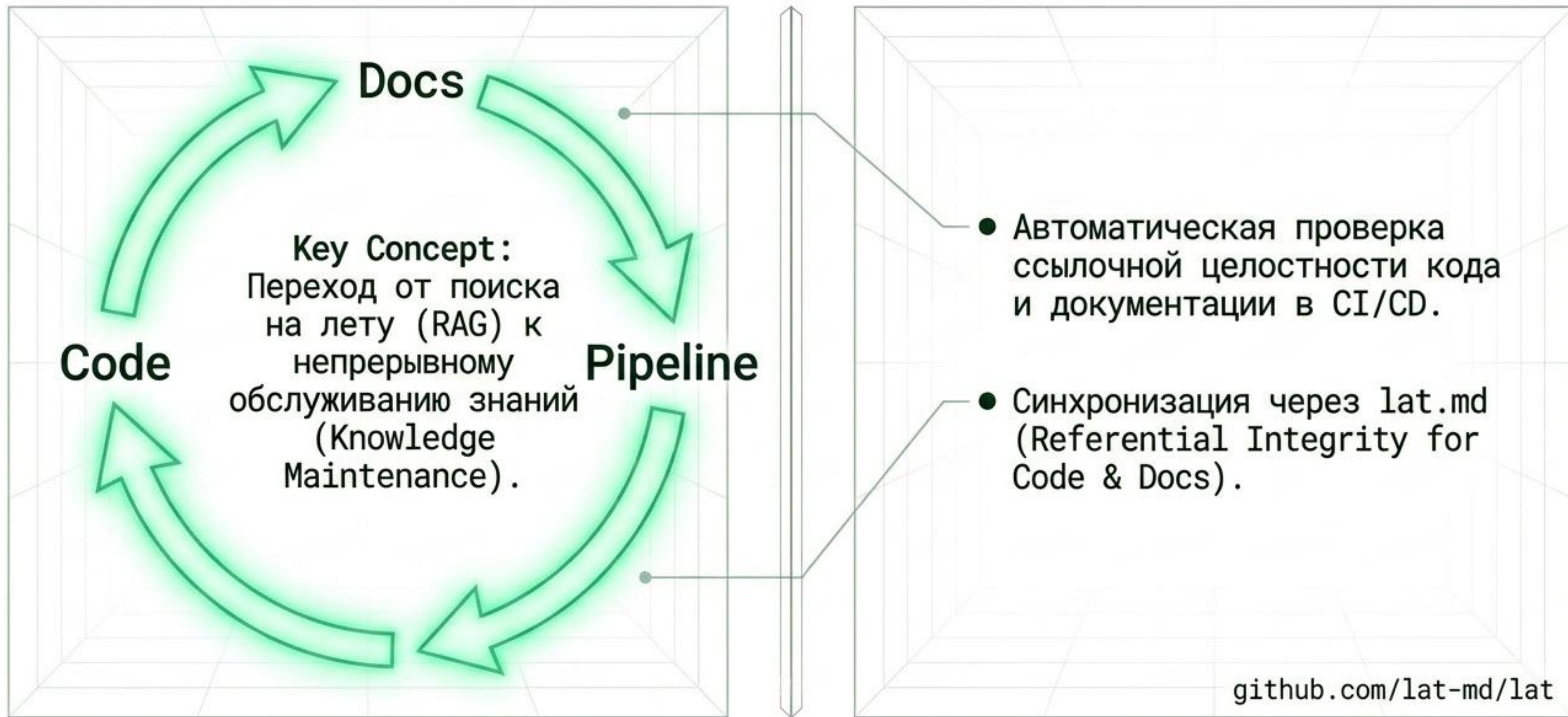
3. Schema: Выделите Enum-список интенгов и Pydantic-схемы.

4. Engine: Настройте vLLM / SGLang с поддержкой outlines (Grammar decoding).

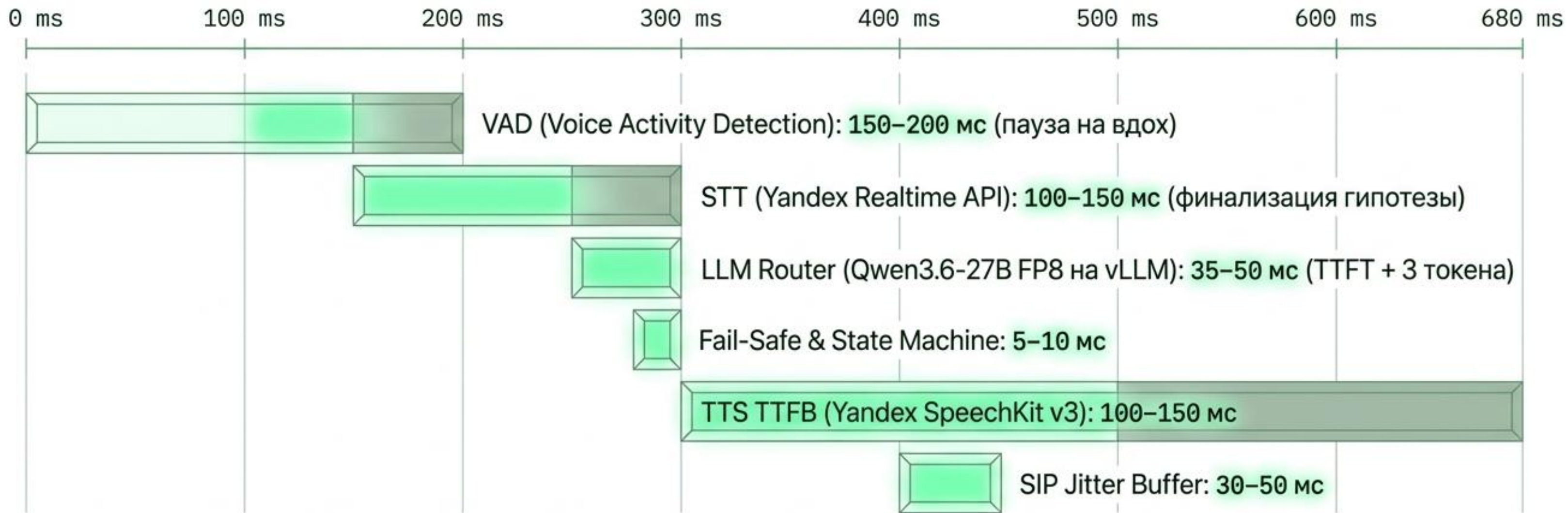
5. Fail-Safe: Обвесьте транзакции Python-валидаторами с учетом STT Confidence.

6. CI/CD: Свяжите документацию и код через lat.md (pre-commit hooks).

Эволюция базы знаний: Непрерывное обслуживание



Честный бюджет задержки: p50 против p99



Итог: p50 = ~420 ms | p99 = ~680 ms
(при пиковых батчах vLLM)



Андрей Носов

Chief AI Architect, RAFT

PhD Tampere University

