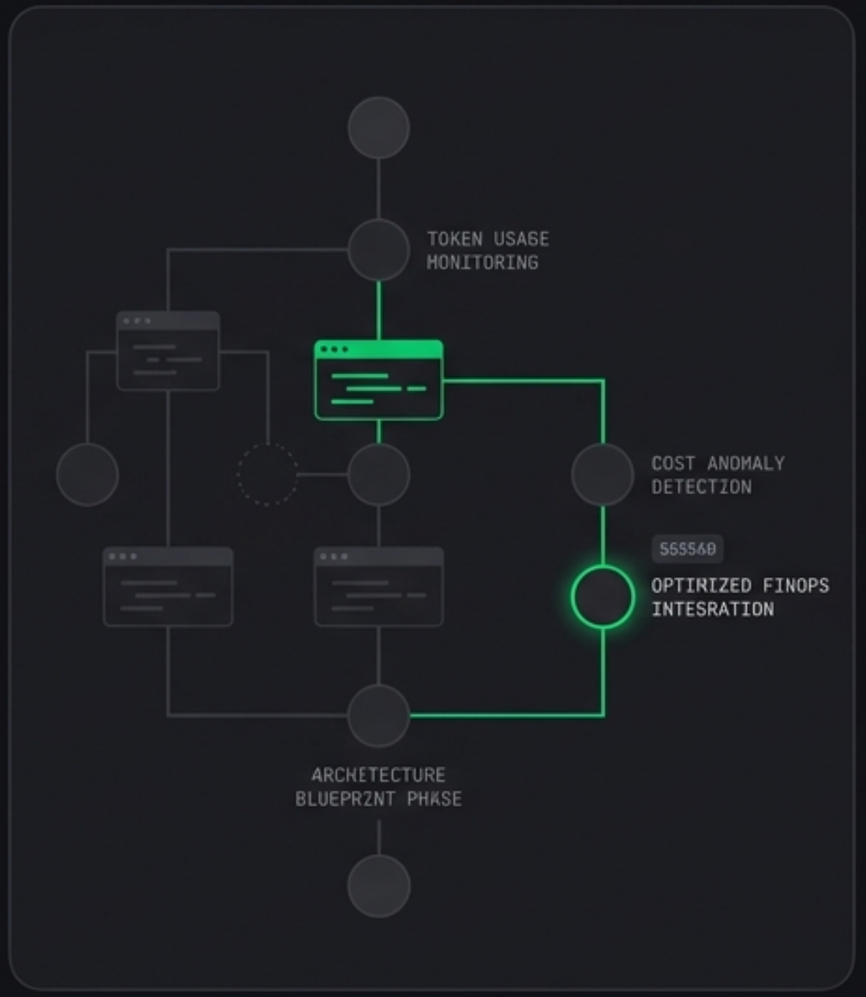
INCIDENT REPORT & ARCHITECTURE

Управление расходами на AI: Тормоза проектируют до запуска

Архитектурный разбор проблемы выгорания токенов и интеграция FinOps на этапе чертежа.



Июнь 2026

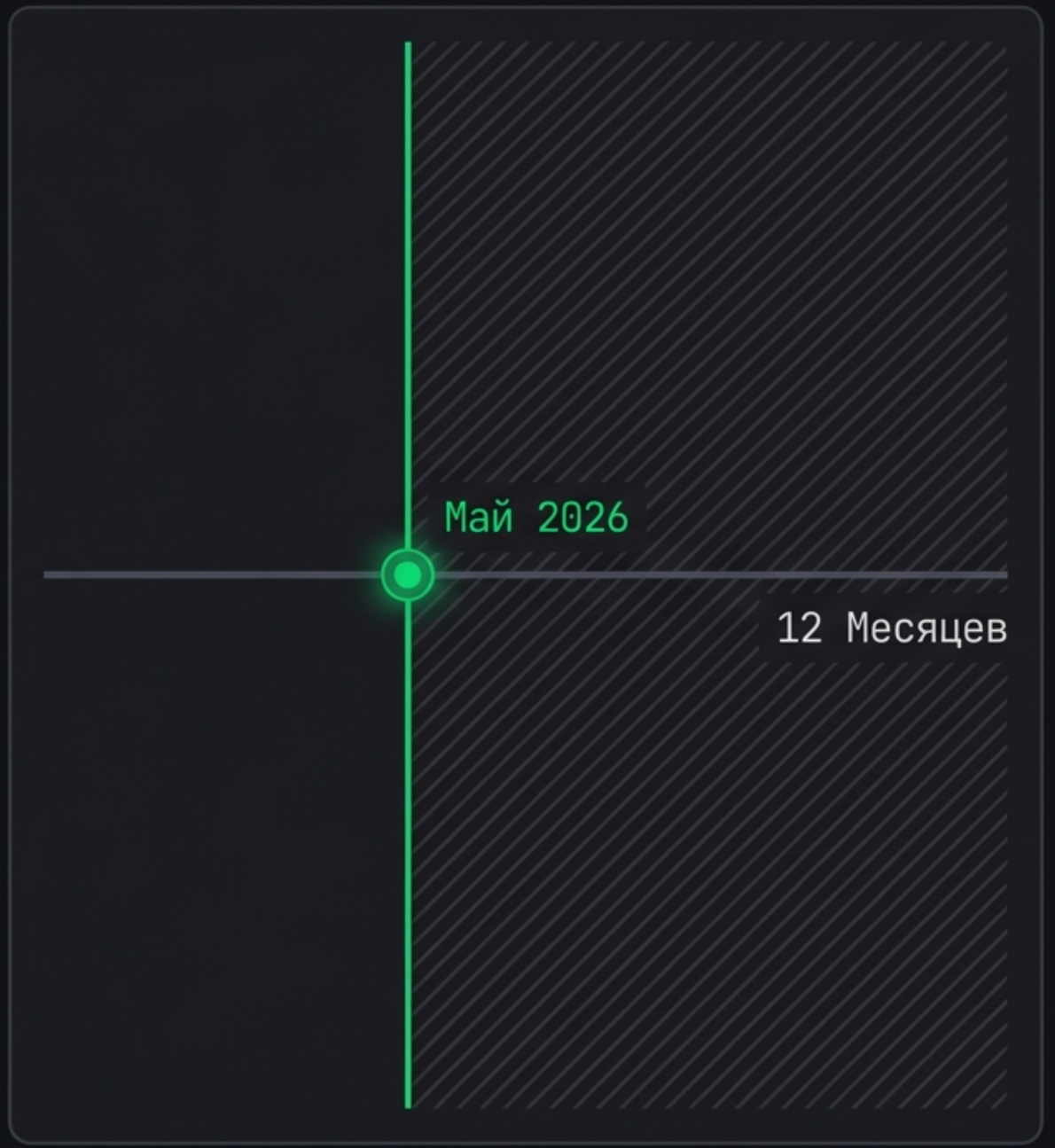
CASE STUDY: UBER

В мае 2026 года CTO Uber публично признал: годовой бюджет на AI-кодинг был выжжен за 4 месяца.

Средний расход: $150–$250/мес на инженера.

Power-users: до $500–$2000/мес на человека.

Проблема не в цене токена. Цена одинакова для всех. Разъехалась архитектура — инженерам выдали мощный агент без тормозов.



Цена модели (Public API)

Публичный прайс-лист на 1M токенов. Статичная, предсказуемая величина.

Стоимость контура (Total System Cost)

Реальное потребление. Зависит от длины контекста, кэш-промахиваний и безлимитного доступа.

Совокупная стоимость корпоративного AI-агента в первый год:

$108k–$306k (CapEx: $70k–$150k, OpEx: $3.2k–$13k/мес).

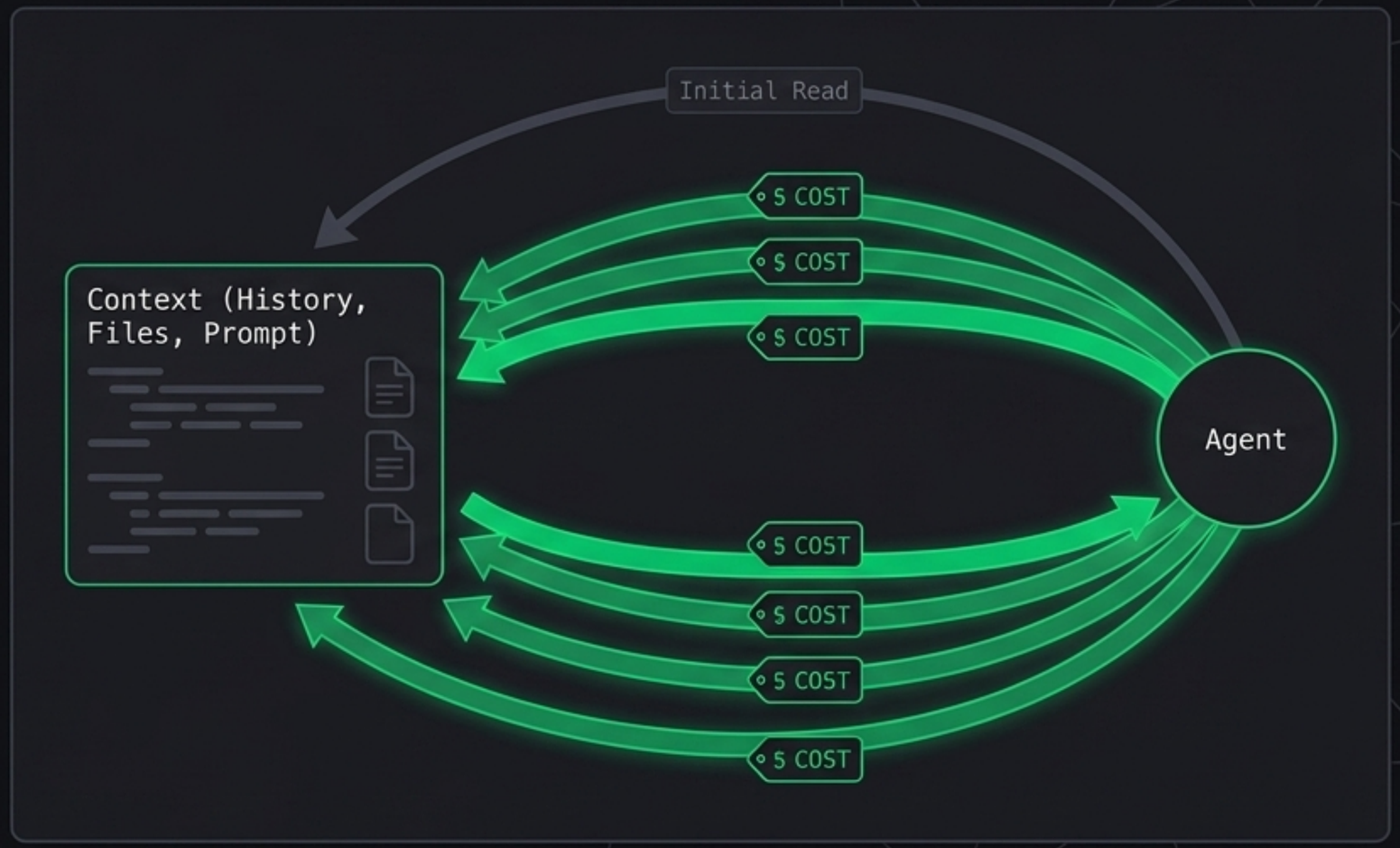
Ежегодное обслуживание съедает **15–25%** от стоимости разработки (подгонка к API и схемам данных). API токены — лишь верхушка айсберга.

MECHANICS OF OVERSPEND

Эффект мультипликатора: Контекст в цикле

Современный агент в задаче кодинга прогоняет через модель не один запрос, а десятки итераций. Он читает файлы, держит историю и перечитывает её на каждом шаге.

Без кэширования каждая итерация оплачивает весь контекст заново по полной цене входного токена. Один инженер выходит на $2000/мес не из-за дорогих токенов, а из-за сотен повторных оплат одного и того же системного промпта.

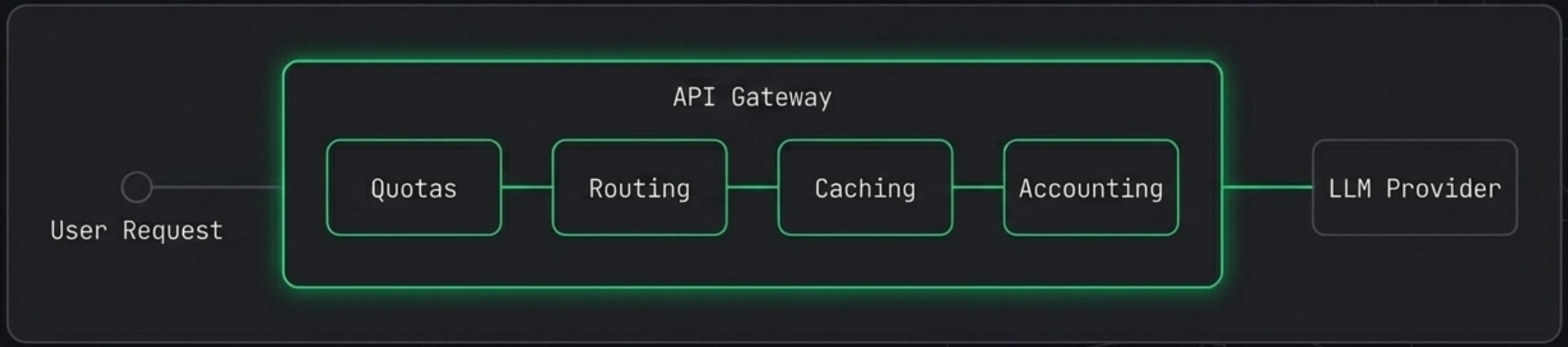


THE PARADIGM SHIFT

FinOps — это слой архитектуры, а не счёт от провайдера

FinOps Foundation определяет контроль затрат не как мониторинг постфактум, а как встроенную операционную модель. Сложность затрат, скорость роста и непредсказуемость AI-нагрузок требуют аппаратных «тормозов» до написания первой строки кода.

Четыре механизма контроля должны проектироваться внутри шлюза (Gateway) вместе с агентом.



COMPONENT 01

Квоты на уровне шлюза

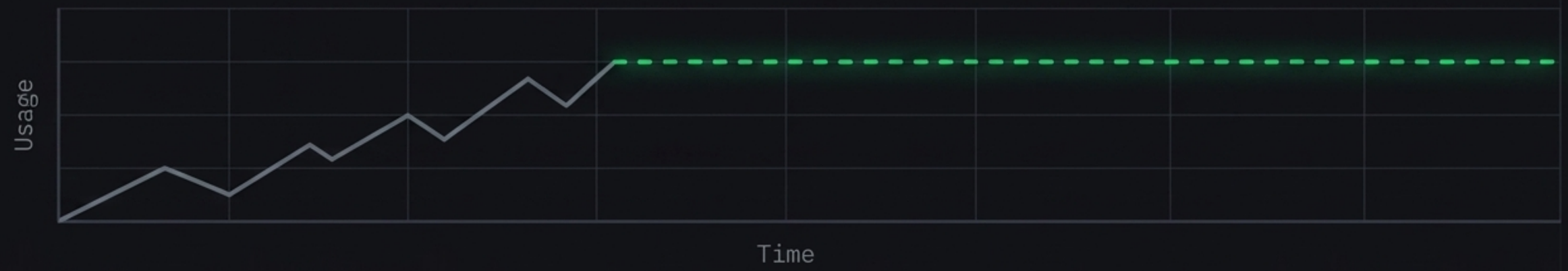
Лимит расхода на пользователя, команду или проект в единицу времени. Мягкий потолок, после которого запрос требует аппрува или автоматически переводится на дешёвую модель. Если не заложить квоту на старте, добавление её после того, как команда привыкнет к безлимиту, воспринимается как болезненный отъём привилегий. Отсутствие этого шлюза стоило Uber их годового бюджета.

Rate Limit / Quota

Routing

Caching

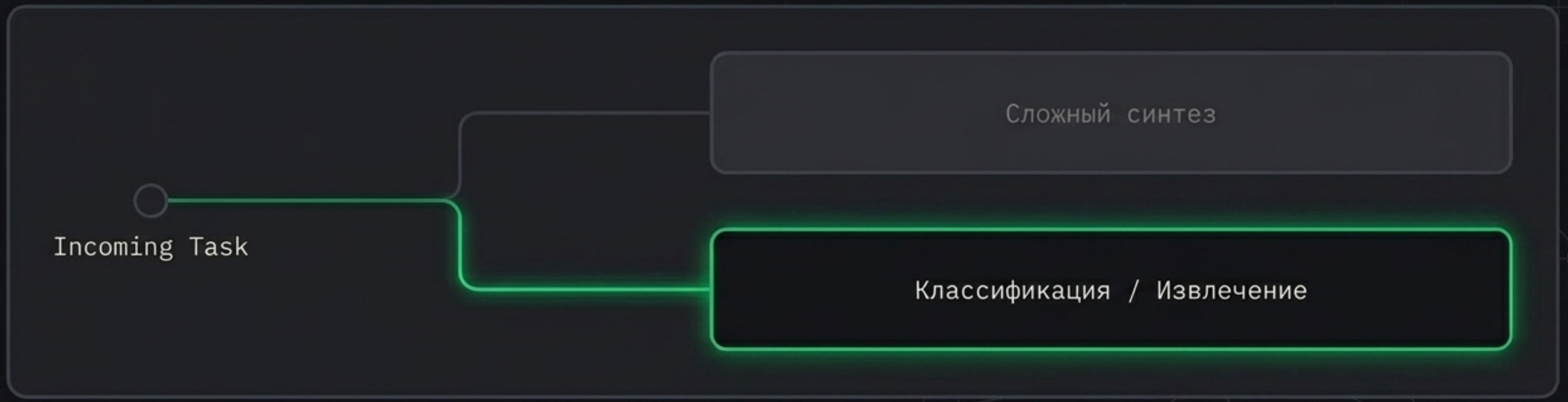
Accounting



COMPONENT 02

Маршрутизация по тиражам моделей

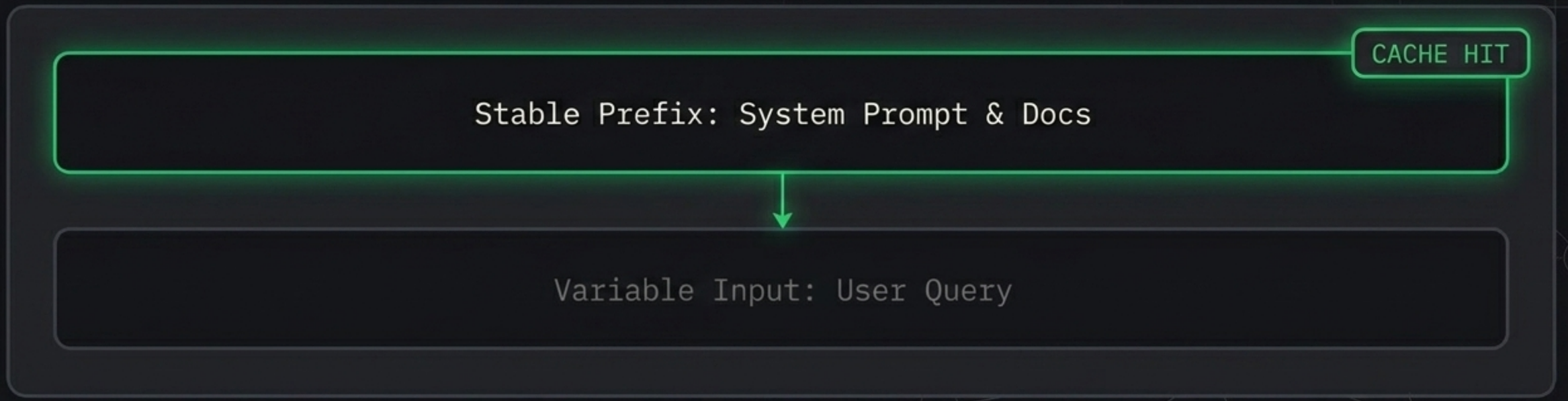
Не каждая задача требует флагмана. Разрыв во входной цене между топовой и компактной моделью одного провайдера достигает порядка. Контур, который гонит всё через топовую модель по умолчанию, переплачивает за каждую тривиальную операцию. Появление компактных моделей высокого качества сделало маршрутизацию архитектурным преимуществом.



COMPONENT 03

Структурирование промпта под кэш

Главный рычаг экономии, доступный бесплатно. В документации Anthropic кэш-чтение стоит существенно дешевле обычного токена.
При вынесении стабильной части в кэшируемый префикс, стоимость повторяющегося контекста падает в разы. Это требует проектирования промпта с первого дня — это не переключатель в настройках облака.

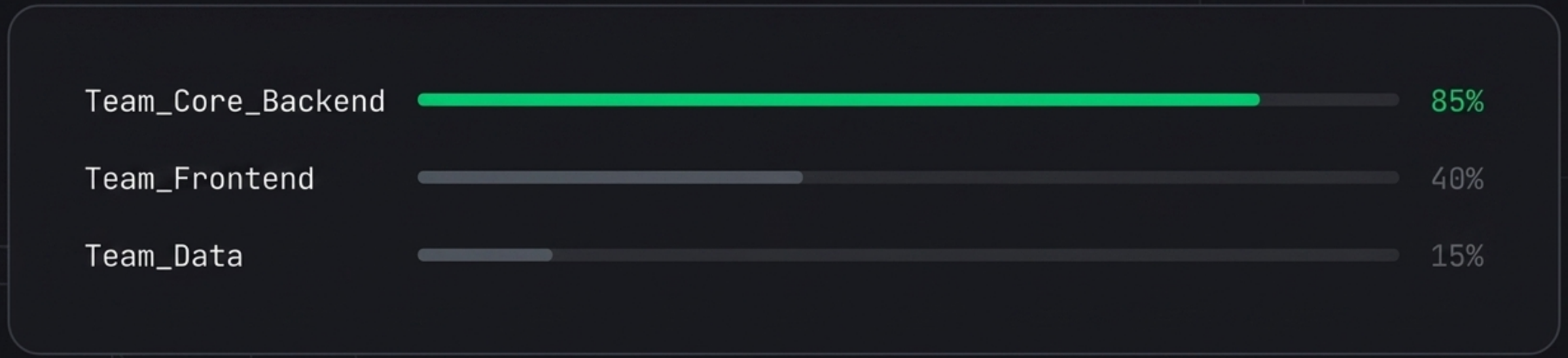


COMPONENT 04

Внутренний учёт и видимость

Биллинг по командам и проектам, который превращает безличный общий счёт в измеримую unit-экономику и персональную ответственность.

Пока расход анонимен, никто его не оптимизирует. Видимость затрат меняет поведение команд без прямых запретов. Перенос ответственности происходит туда, где принимается техническое решение.



ARCHITECTURE DIFF

Сравнение подходов к проектированию

Всё через флагман	FinOps на чертеже
Отъём привилегий после шока	Заложены в шлюз изначально
Всё идёт через флагман	Тривиальное уходит на дешёвые модели
Оплачивается заново каждый шаг	Промпт структурирован под кэш
Анонимный общий счёт	Биллинг по командам

COMPONENT 04

THE MACRO CONTEXT & TECH DEBT

Дешевизна моноархитектуры берётся в долг

Данные MIT NANDA

95%

корпоративных AI-пилотов не дают измеримого ROI через полгода. Причина — экономика контура ломается на архитектуре раньше, чем на качестве самой модели.

Сигнал от Microsoft

Пересмотр внутренних лицензий на сторонние агенты в пользу собственного стека к середине 2026 года.

Цель: возврат архитектурного контроля над расходами.

На старте пустить всё через один API быстрее. Но на масштабе долг гасится с процентами в виде перерасхода и мучительного рефакторинга работающего продакшена.

CODE REVIEW / ENGINEERING

Диагностика для архитектора

01 Тест на кэш: Какова доля попаданий (cache hit rate) в реальной нагрузке? Выделен ли стабильный префикс (системный промпт, доки) жестко от запроса пользователя?

02 Тест на маршрутизацию: Какая доля запросов объективно требует топовой модели? Идёт ли извлечение полей через флагман?

03 Тест на квоты: Есть ли в шлюзе хард-лимит? Что произойдет на уровне API, если скрипт зациклится на ночь? (Если ответ «ничего, кроме огромного счёта» — тормозов нет).

CODE REVIEW / MANAGEMENT

Диагностика для техлида / C-level

Тест на TCO

Заложена ли в смету совокупная стоимость контура и ежегодного обслуживания (15-25% от CapEx), или бизнес-план строится только на цене API?

Тест на видимость

Может ли каждая конкретная продуктовая команда в реальном времени увидеть свою персональную строку расхода на AI?

Тест на обратимость

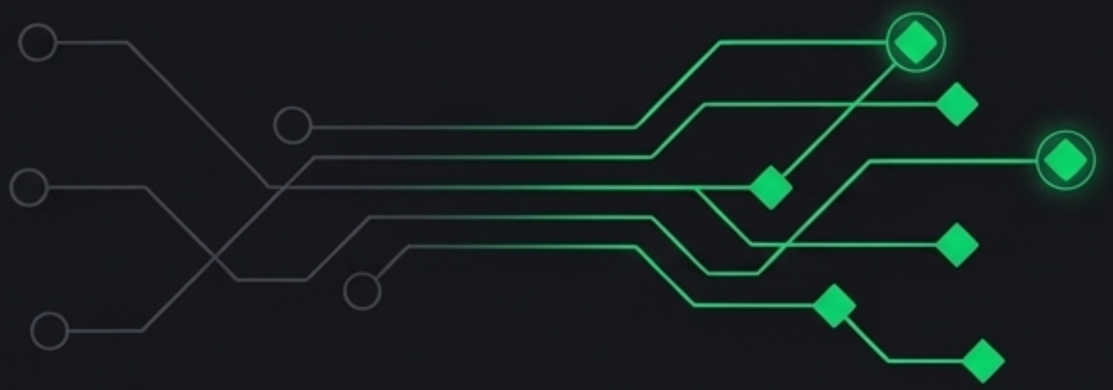
Сколько человеко-часов будет стоить внедрить роутинг и квоты через полгода работы инженеров на безлимите?

MARKET SIGNALS: 2026

Эволюция B2B-агентов

Защищённая строка бюджета

FinOps для AI становится обязательной частью базовой B2B-сметы. Без неё история Uber повторится у каждого при выходе из пилота.



Сдвиг парадигмы вендоров

Фокус провайдеров смещается с «насколько умна наша модель» на «как управлять расходами в нашем контуре». Управление бюджетом, лимиты и маршрутизация становятся встроенной частью продукта, а не самодельной обвязкой.

Экономика AI-контура решается архитектурой, а не прайс-листом. Проектируйте тормоза до запуска, а не после счёта.

Конец документа.