

Claude&MCPs в работе техлида



Claude&MCPs в работе техлида



Как **Иван Луценко**
перестал бояться и
полюбил LLM, Model
Context Protocol,
Firebase Crashlytics и
назначение 1:1

Обо мне

03

Роль: Android Tech Lead
Bereke Business, Bereke Bank



Обо мне

04

Роль: Android Tech Lead
Bereke Business, Bereke Bank
Специализация: платформенная
Android-разработка



Обо мне

05

Роль: Android Tech Lead
Bereke Business, Bereke Bank
Специализация: платформенная
Android-разработка
Опыт: пет-проекты, аутсорс-проекты,
банковское приложение



Обо мне

06

Роль: Android Tech Lead

Bereke Business, Bereke Bank

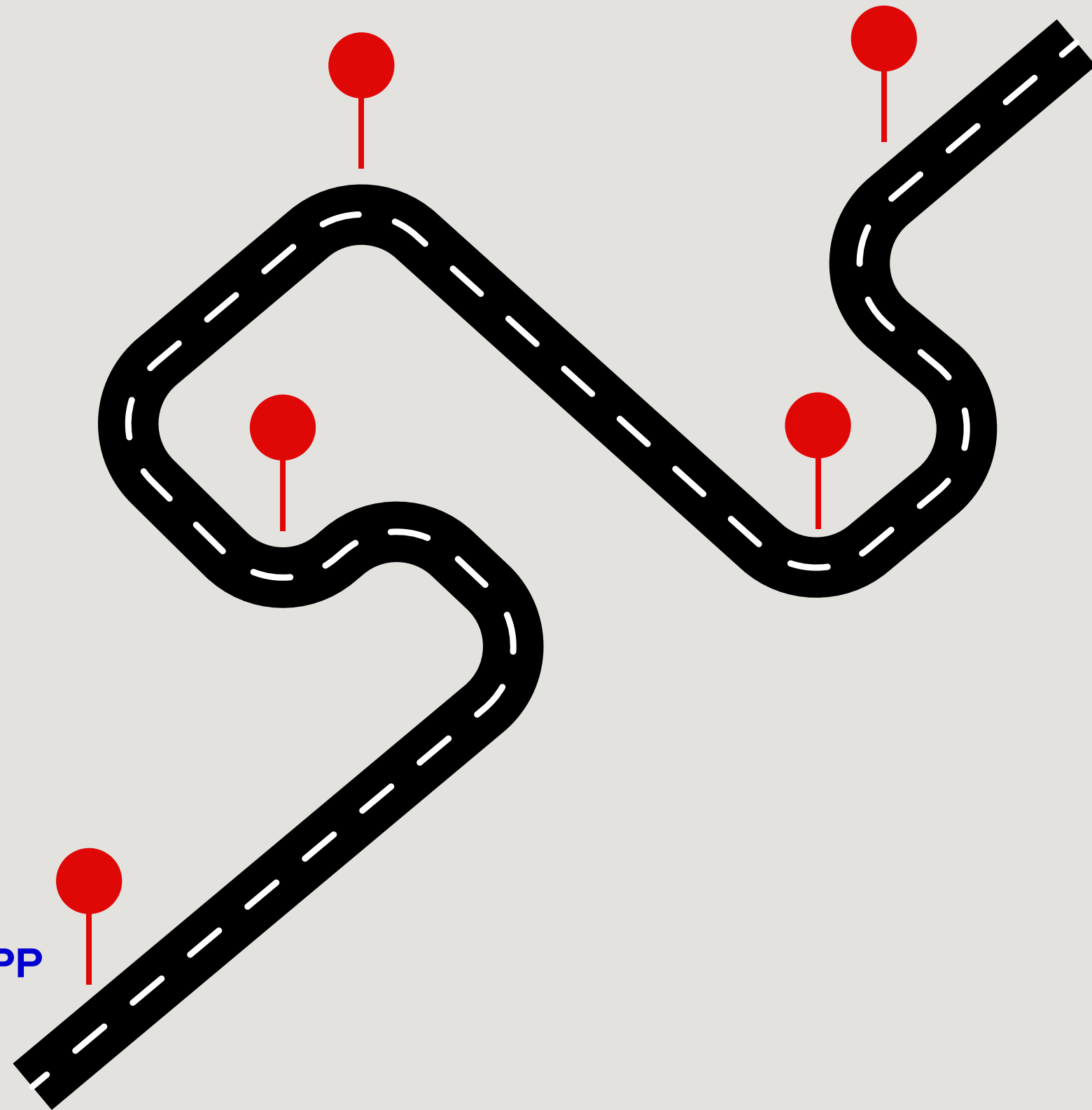
Специализация: платформенная
Android-разработка

Опыт: пет-проекты, аутсорс-проекты,
банковское приложение

Текущий фокус: AI-ассистируемая разработка
и управление проектами



LLM В МОЕЙ ЖИЗНИ



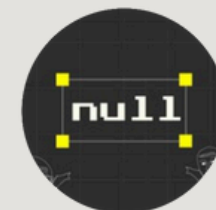
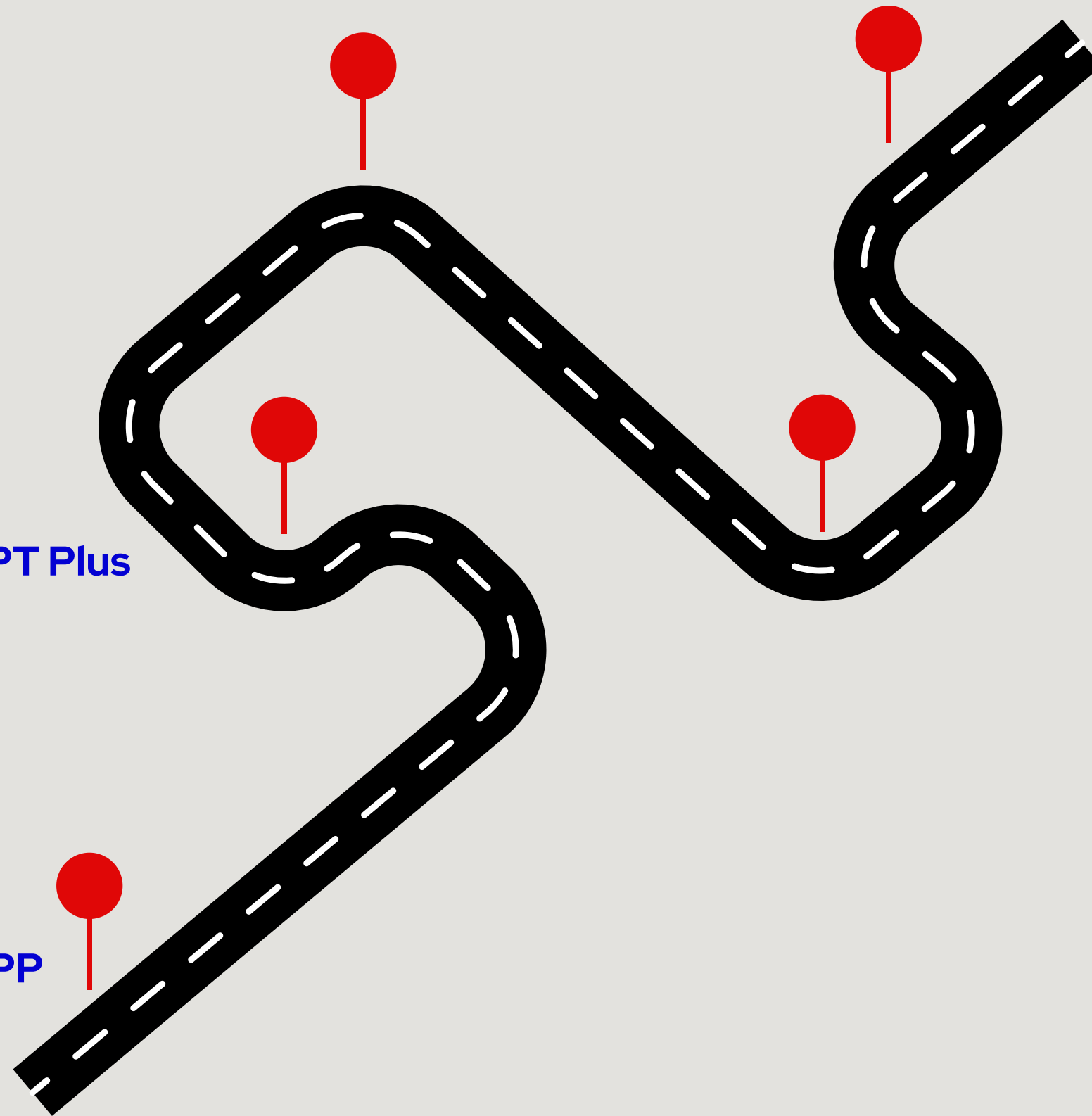
@ALMATY_NULLPTR
_PARTY

2024 — по настоящее время NPP

LLM В МОЕЙ ЖИЗНИ

осень 2024 – весна 2025 Chat GPT Plus

2024 – по настоящее время NPP



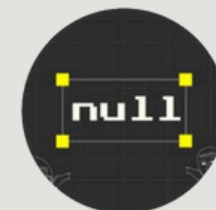
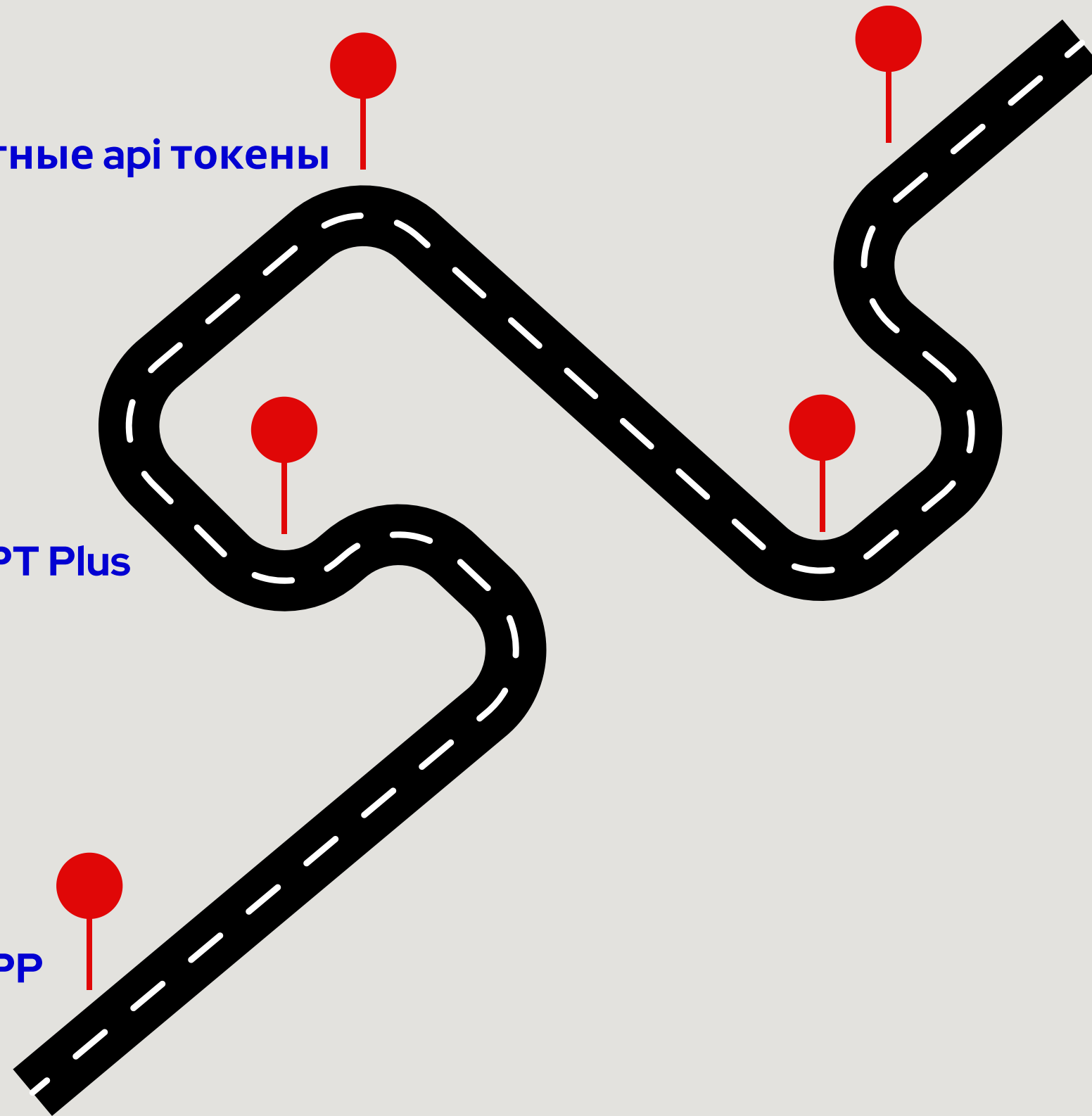
@ALMATY_NULLPTR
_PARTY

LLM В МОЕЙ ЖИЗНИ

зима 2024-25 – aider + бесплатные api токены

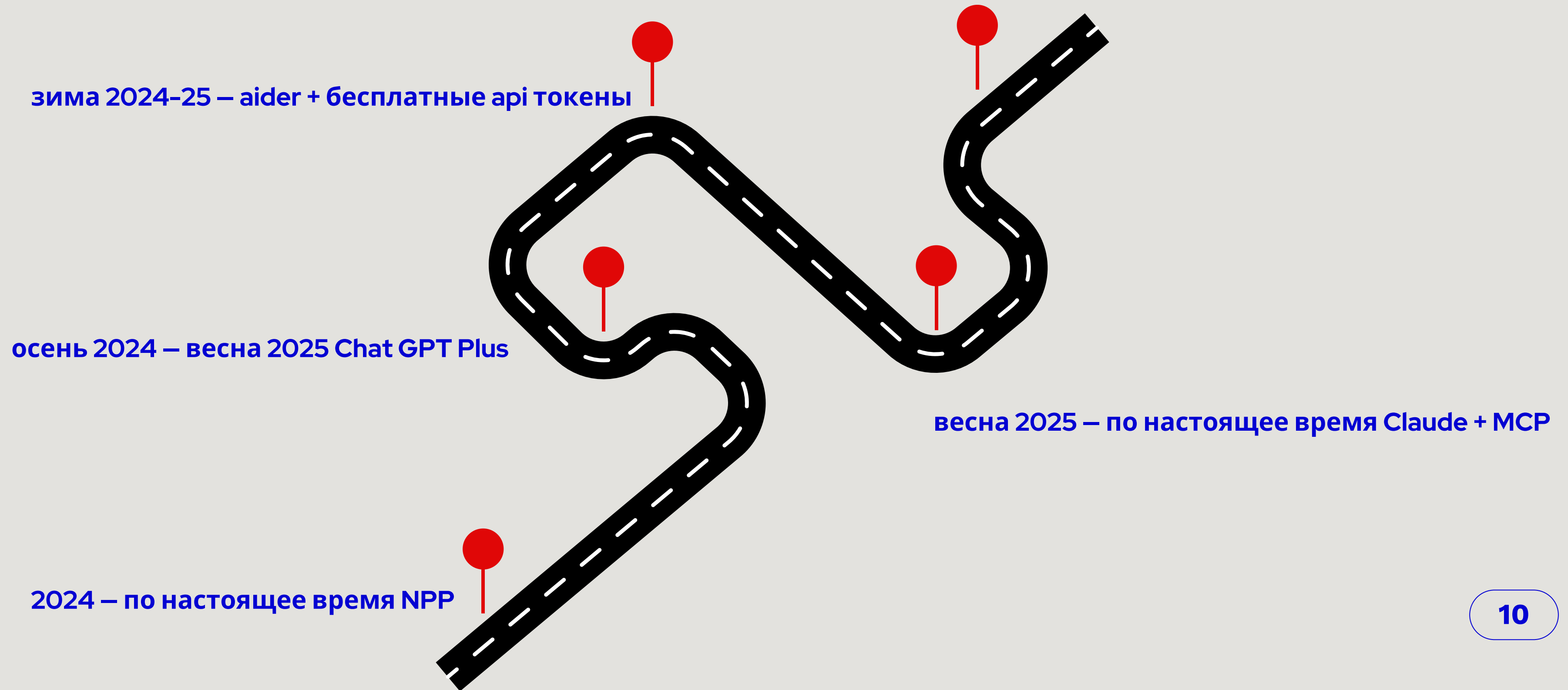
осень 2024 – весна 2025 Chat GPT Plus

2024 – по настоящее время NPP

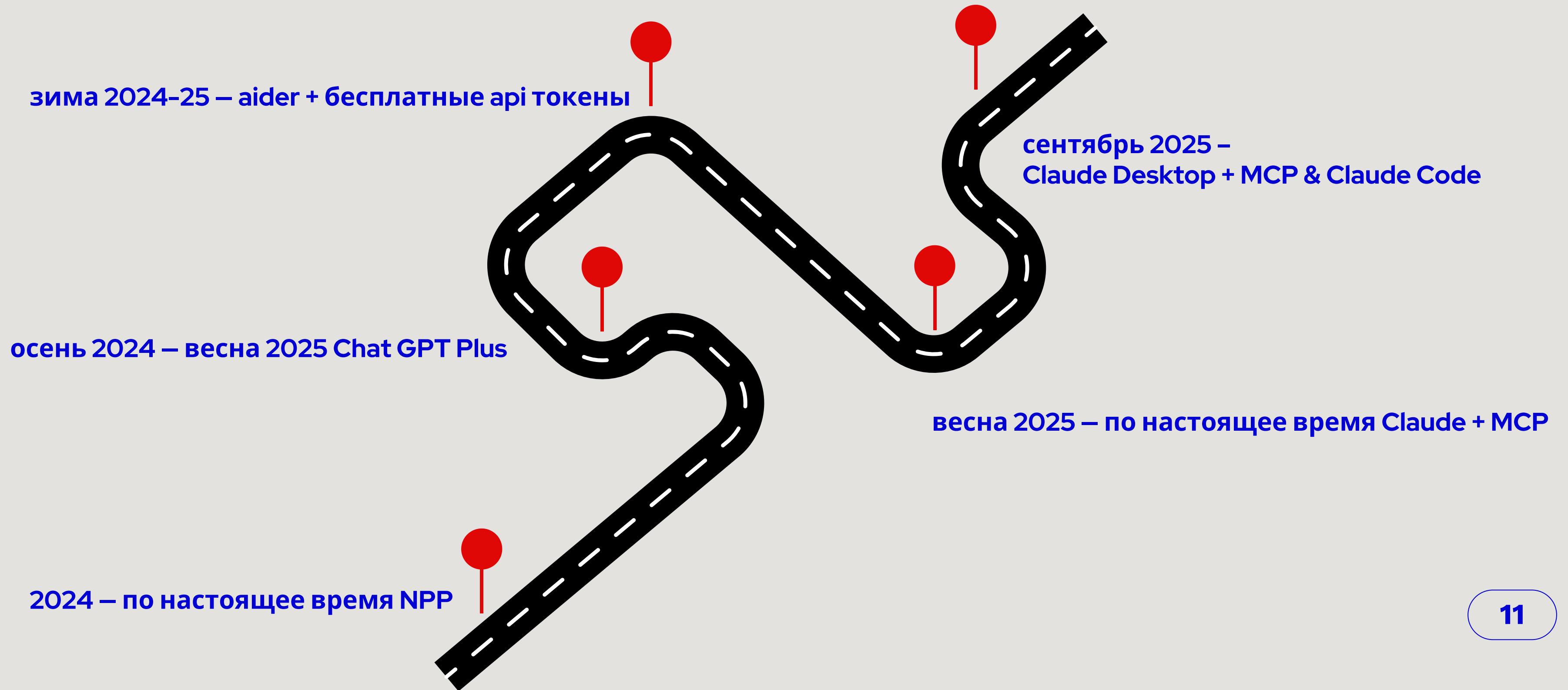


@ALMATY_NULLPTR
_PARTY

LLM В МОЕЙ ЖИЗНИ



LLM В МОЕЙ ЖИЗНИ



LLM В МОЕЙ ЖИЗНИ



Claude vs ChatGPT

Ключевая особенность:

Claude понимает контекст и может работать с внешними инструментами через MCP

Отличия от ChatGPT:

Более консервативные, но надежные ответы + нативная поддержка MCP, специализация на тексте и коде, ни картинок, ни прозвищ, ни заигрывания с широкой аудиторией



Claude vs ChatGPT

Ключевая особенность:

Claude понимает контекст и может работать с внешними инструментами через MCP

Отличия от ChatGPT:

Более консервативные, но надежные ответы + нативная поддержка MCP, специализация на тексте и коде, ни картинок, ни прозвищ, ни заигрывания с широкой аудиторией



Claude vs ChatGPT

Ключевая особенность:

Claude понимает контекст и может работать с внешними инструментами через MCP

Отличия от ChatGPT:

Более консервативные, но надежные ответы + нативная поддержка MCP, специализация на тексте и коде, ни картинок, ни прозвищ, ни заигрывания с широкой аудиторией

Проблема: N×M интеграций

До MCP

- Каждая AI-система требует **отдельных** интеграций
- Claude + GitHub = **отдельный код**
- GPT + GitHub = **другой код**
- Gemini + GitHub = **третий код**
- N моделей × M сервисов = **хаос**

Проблема: $N \times M$ интеграций

17

До MCP

- Каждая AI-система требует **отдельных** интеграций
- Claude + GitHub = **отдельный код**
- GPT + GitHub = **другой код**
- Gemini + GitHub = **третий код**
- N моделей \times M сервисов = **хаос**

с MCP

- Единый протокол для всех
- M серверов + N клиентов
- Пишешь один MCP сервер
- Работает со всеми AI моделями
- **$N + M$ вместо $N \times M$**

Проблема: $N \times M$ интеграций

До MCP

- Каждая AI-система требует **отдельных** интеграций
- Claude + GitHub = **отдельный код**
- GPT + GitHub = **другой код**
- Gemini + GitHub = **третий код**
- N моделей \times M сервисов = **хаос**

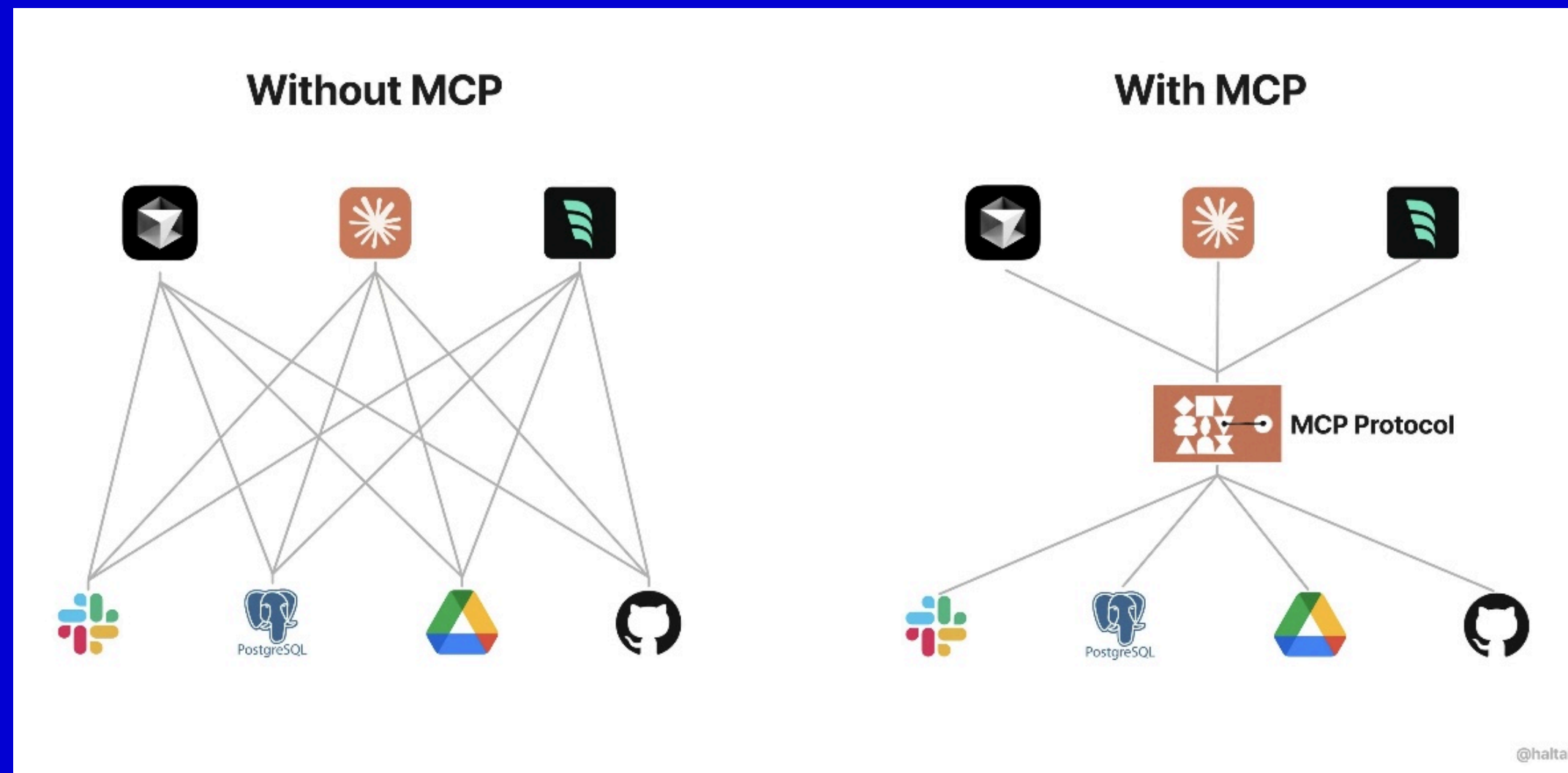
с MCP

- Единый протокол для всех
- M серверов + N клиентов
- Пишешь один MCP сервер
- Работает со всеми AI моделями
- **$N + M$ вместо $N \times M$**

MCP = "USB-C порт для AI-приложений"
Универсальный порт между AI-моделями и внешними данными

Проблема: N×M интеграций

19



Anthropic
Ноябрь 2024
Создатели МСР

Быстрое
принятие
индустрией

Anthropic
Ноябрь 2024
Создатели MCP

OpenAI
Март 2025
Поддержка в API

Быстрое
принятие
индустрией

Anthropic
Ноябрь 2024
Создатели MCP

OpenAI
Март 2025
Поддержка в API

Google
Апрель 2025
Gemini интеграция

Быстрое
принятие
индустрией

Быстрое принятие индустрией

Anthropic
Ноябрь 2024
Создатели MCP

OpenAI
Март 2025
Поддержка в API

Google
Апрель 2025
Gemini интеграция

Microsoft
Май 2025
VS Code Preview

Anthropic
Ноябрь 2024
Создатели MCP

Статистика: **5000+** активных MCP серверов (май 2025)

**По версии международного агентства OBS*

OpenAI
Март 2025
Поддержка в API

Google
Апрель 2025
Gemini интеграция

Microsoft
Май 2025
VS Code Preview

Быстрое
принятие
индустрией

Мой МСР стек

25

Официальные от Anthropic

- ➡ Google Drive документооборот
- ➡ Sequential Thinking решение проблем
- ➡ Web Search поиск в интернете
- ➡ Desktop Commander файловая система

Мой МСР стек

26

Официальные от Anthropic

- ➡ Google Drive документооборот
- ➡ Sequential Thinking решение проблем
- ➡ Web Search поиск в интернете
- ➡ Desktop Commander файловая система

Сторонние

- ➡ Serena анализ кода и работа с ним
- ➡ JetBrains интеграция с IntelliJ
- ➡ Task Tracker управление проектами*
- ➡ Research Agent углубленный поиск
- ➡ ACI Unified интегратор интеграций :)

** Task Tracker разработан мной
с помощью Claude*

Мой МСР стек

27

Официальные от Anthropic

- ➡ Google Drive документооборот
- ➡ Sequential Thinking решение проблем
- ➡ Web Search поиск в интернете
- ➡ Desktop Commander файловая система

Сторонние

- ➡ Serena анализ кода и работа с ним
- ➡ JetBrains интеграция с IntelliJ
- ➡ Task Tracker управление проектами*
- ➡ Research Agent углубленный поиск
- ➡ ACI Unified интегратор интеграций :)

** Task Tracker разработан мной
с помощью Claude*

Основной фокус: AI-ассистированная разработка + управление проектами

Кейс 1: Анализ крашей Android-приложения

28

Краткое содержание промпта:

Роль: Staff Android Developer, эксперт по дебагу мирового уровня

Цель проекта: Поддержание crash-free пользователей >99% в банковском Android-приложении для B2B

Инструменты и подходы:

- Анализ стектрейсов из Firebase Crashlytics
- Доступ к коду через Model Context Protocol (MCP)
- Git Blame для назначения ответственных
- Системная классификация и приоритезация багов
- Генерация кратких и детальных отчётов
- для JIRA

Бонус: \$5000 за каждый точно проанализированный баг с найденным assignee (ну как минимум так в промпте)

● ● ● Crashlytics → Claude MCP workflow

- 1 Пользователь: [стектрейс из Firebase Crashlytics]
- 2 Claude:
- 3 1. serena:*read_file()*
// Чтение кода из стектрейса
- 4 2. serena:*find_symbol*("PaymentService")
// Поиск проблемного класса
- 5 3. ???
- 6 4. git blame analysis
// Определение ответственного
- 7 5. Генерация JIRA-тикета с готовым решением
- 8
- 9 → Результат: 2-3 часа расследования → 10-15 минут автоматизированного анализа

Кейс 1: Анализ крашей Android-приложения

29

```
# Crashlytics - Stack trace
# Application: kz.berekebank.business.app
# Platform: android
# Version: 1.49.0 (157)
# Issue: 399568e3bbffc845256f2a6fe83db9ec
# Session: 68EC9E1F01F800015F390AA2804819CE_DNE_0_v2
# Date: Mon Oct 13 2025 11:41:26 GMT+0500 (GMT+05:00)
Fatal Exception: kz.berekebank.business.core.presentation.extensions.NoImplementedParentException: No parent implements class
kz.berekebank.business.core.presentation.maintabs.MainTabsBottomNavigationHelper
    at kz.berekebank.business.core.presentation.extensions.FragmentExtKt.getParent(FragmentExt.kt:158)
    at
kz.berekebank.business.feature.accounts.accounts_impl.AccountsFragmentLauncherImpl.hideBottomNavBar(AccountsFragmentLauncherImpl.kt:50)
    at
kz.berekebank.business.feature.accounts.accounts_impl.AccountsFragmentLauncherImpl.launchFromExactAccount(AccountsFragmentLauncherImpl.
kt:24)
    at
kz.berekebank.business.mainscreen_impl.domain.usecase.AccountsUseCaseImpl.getAccounts$lambda$3$lambda$2(AccountsUseCaseImpl.kt:73)
    at kz.berekebank.business.mainscreen_impl.domain.usecase.AccountsUseCaseImpl.$r8$lambda$rUergiyIbZm3GMVNgGQmgkxjgAM()
    at kz.berekebank.business.mainscreen_impl.domain.usecase.AccountsUseCaseImpl$$ExternalSyntheticLambda0.invoke(D8$$SyntheticClass)
    at kz.berekebank.business.mainscreen_impl.presentation.view.ProductViewKt.ProductItem$lambda$21$lambda$20(ProductView.kt:173)
    at kz.berekebank.business.mainscreen_impl.presentation.view.ProductViewKt.$r8$lambda$4J6xtWLSnR-F4DITovk7G_BDic()
    at kz.berekebank.business.mainscreen_impl.presentation.view.ProductViewKt$$ExternalSyntheticLambda9.invoke(D8$$SyntheticClass)
    at androidx.compose.foundation.ClickableNode$clickPointerInput$3.invoke-k-4lQ0M(Clickable.kt:699)
    at androidx.compose.foundation.ClickableNode$clickPointerInput$3.invoke(Clickable.kt:693)
```

И ещё **964** строки

Кейс 1: Анализ крашей Android-приложения

КРАТКИЙ ОТЧЕТ ДЛЯ JIRA

[Critical] DocumentsRootFragment crash: "No view found for frameLayout_main"

Проблема: При переключении на вкладку "Операции" происходит краш из-за race condition в fragment management

Анализ:

```
IllegalArgumentException: No view found for id frameLayout_main
at FragmentStateManager.createView(FragmentStateManager.java:567)
at FragmentManager.executeOpsTogether(FragmentManager.java:2214)
```

Корневая причина: Race condition в `MainTabsHostFragment.showBottomNavFragment()`: `popBackStack()` выполняется асинхронно, а `commit()` начинается до завершения удаления предыдущего фрагмента, что приводит к попытке добавления фрагмента в несуществующий контейнер.

Проблемный код (MainTabsHostFragment.kt, строки 230-238):

```
kotlin

childFragmentManager.popBackStack(FRAGMENT_STACK_ROOT, POP_BACK_STACK_INCLUSIVE)
childFragmentManager.commit(allowStateLoss = true) {
    replace(fragmentContainerId, fragment)
    addToBackStack(FRAGMENT_STACK_ROOT)
}
```

Решение:

```
kotlin
```

↓

Дополнительные изменения:

- Добавить проверку `if (view != null && isAdded)` перед fragment операциями
- Рассмотреть миграцию на Navigation Component для надежного lifecycle management

Приоритет: Critical

Файл:

`app/src/main/java/kz/berekebank/business/app/clean/presentation/maintabs/MainTabsHostFragment.kt`

Строки: 230-238

Assignee: `uo.krasnonozhenko` (основной кандидат - последний работал над фиксом BasePagerAdapter в коммите 9ee2aad6dc "CPT-3081 Fix FC in BasePagerAdapter" от 22 сентября 2025, где уже пытался решить похожую проблему с fragment lifecycle)

Альтернативный assignee: `u17902` (автор изменений в showBottomNavFragment, коммиты 896ff6f49a4 и 3f3e8ce3dd8 от 21-22 ноября 2024, которые потенциально создали эту проблему)

Воспроизведение:

- Быстро переключаться между вкладками "Главная" ↔ "Операции" (3-4 раза с интервалом <500ms)
- Альтернативно: открыть "Операции" → свернуть приложение → вернуться через 1-2 минуты → переключить вкладку

Валидация фикса: Monkey testing с фокусом на навигацию, метрика success: 0 крашей за 10000 navigation events

↓

Кейс 1: Анализ крашей Android-приложения, пример:

- ➔ Корневая причина
- ➔ Проблемный код
- ➔ Решение, несколько вариантов

Кейс 1: Анализ крашей Android-приложения, пример:

- ➔ Корневая причина
- ➔ Проблемный код
- ➔ Решение, несколько вариантов
- ➔ Ответственный
- ➔ Сценарий воспроизведения
- ➔ Два представления – полный мне и короткий для QA

Кейс 2: Проектный трекер 2.0

Задача: **Universal Project Tracker Generator**

Создание универсального инструмента управления проектами с двойным интерфейсом: **Obsidian + MCP**, принимающего на вход простой список задач, а выдающего папку с **.md**, с которыми сам же и работает

▼ Projects

▼ Kover Testing Plan

▼ Phases

Phase 1 - Basic Tests Completion

Phase 2 - Testing Individual Cov...

▼ Sessions

Session 2025-01-17

▼ Tasks

Task 1.1 - Finalize smoke test

Task 2.3 - Проверить BadView...

Task 2.3 - Check BadViewModel

!Project Dashboard

Project Bible

▼ Templates

Phase Template

Project Template

Session Log Template

Task Template

!Projects Dashboard

MCP Integration Guide

Project Workflow

Project Bible

 **Kover Testing Plan - Project Bible**

 **Цель проекта**

Создание и тестирование системы автоматической проверки покрытия кода для Kotlin проектов с использованием Kover. Система должна обеспечивать индивидуальную проверку файлов на соответствие минимальному порогу покрытия (80%).

 **Архитектура системы**

Основные компоненты:

- individual_coverage_hook.sh - основной скрипт проверки покрытия
- run_individual_coverage.sh - скрипт-обертка для удобного запуска
- test_coverage_high.sh - тестирование файлов с высоким покрытием
- test_coverage_low.sh - тестирование файлов с низким покрытием
- test_coverage_scenarios.sh - полный набор тестовых сценариев

Тестовые классы:

- GoodRepository - класс с высоким покрытием (>80%)
- BadViewModel - класс с плохим покрытием (<80%)
- BorderlineUseCase - граничный случай (~80%)

 **Выполнено за сессию**

-  Создан главный Projects Dashboard
-  Настроен Project Workflow с MCP командами
-  Создан Project Dashboard для Kover Testing Plan
-  Написан Project Bible с полным описанием
-  Созданы Templates для автоматизации (Project, Task, Session, Phase)
-  Создана документация для Task 2.3
-  Протестирована интеграция с claude-task-tracker
-  Получен статус проекта: 7/29 задач (24% готово)

 **В процессе выполнения**

-  Документирование полного workflow
-  Подготовка к выполнению Task 2.3
-  Создание примеров автоматизации

Кейс 3: Автоматизация 1:1 встреч в календаре

34

Краткое содержание промпта:

Роль: Экспертная система для генерации .ics-файлов, полностью совместимых с Outlook, Google и macOS Calendar

Цель проекта: Создание индивидуальных и объединённых календарных файлов на основе текстового списка встреч

Ключевые особенности:

- Строгий .ics-формат без ошибок импорта
- Только английский язык, локальное время
- Метод PUBLISH, статус CONFIRMED, TRANSP:OPAQUE
- Генерация UID: firstname-1-1@company.com
- 2 файла: индивидуальные + общий
- Проверка соответствия формату (валидатор)

Преимущества:

- Надёжный импорт в любые календари
- Минимум ручной работы
- Предсказуемое поведение и формат



Crashlytics → Claude MCP workflow

- 1 # Рутинные задачи → автоматизация
- 2 Ввод: "Василий - среда, 22 мая, 16:00, vasyan@company.com"
- 3
- 4 1. Парсинг участников и времени
- 5 2. Генерация .ics *файлов* (индивидуальные + общий)
- 6 3. Валидация совместимости с Outlook/Google Calendar
- 7 4. *Локализация* (кириллица → транслитерация)
- 8
- 9 → Ручное создание встреч → автогенерация .ics файлов

TRAINEE

- Задача

Эволюция анатомии промпта

TRAINEE

- Задача

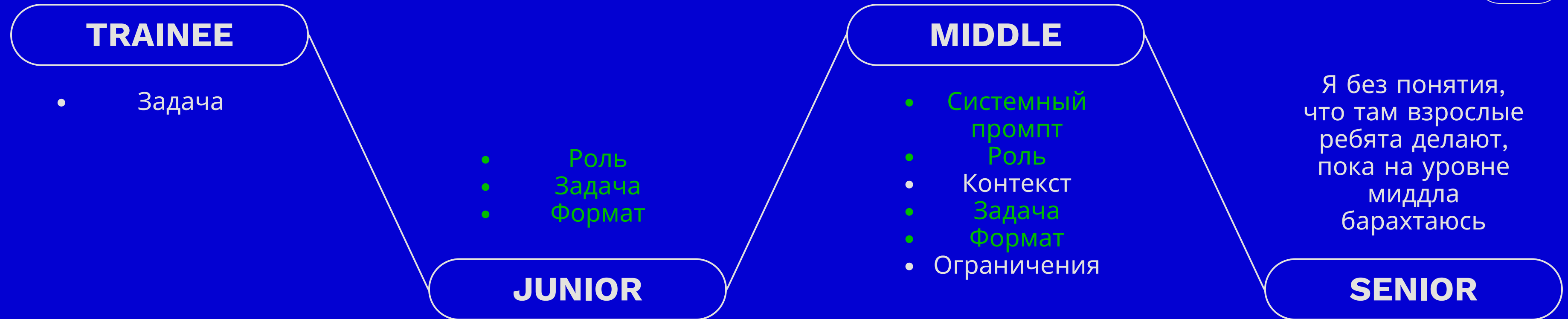
- Роль
- Задача
- Формат

JUNIOR

Эволюция анатомии промпта



Эволюция анатомии промпта



Эволюция анатомии промпта

Пример системного промпта

- Анализ промпта
- Назначение роли
- Создание внутренней системы оценки ответа



Пример системного промпта

- Анализ промпта
- Назначение роли
- Создание внутренней системы оценки ответа
- Контроль следования роли
- QoL инструкции, например ответы на языке промпта пользователя



Преимущества и ROI

41

1. **60–80% сокращение времени разработки***
2. Анализ кода **15 минут поиска → 30 секунд** через serena

Преимущества и ROI

42

- 60–80% **сокращение времени разработки***
- Анализ кода **15 минут поиска → 30 секунд** через serena
- **Отладка**, переключение окон → прямые команды в чате
- Планирование, **разрозненные инструменты** → единый Claude workflow

Преимущества и ROI

43

- 60–80% сокращение времени разработки*
- Анализ кода 15 минут поиска → 30 секунд через serena
- Отладка, переключение окон → прямые команды в чате
- Планирование, разрозненные инструменты → единый Claude workflow
- \$5000 за баг в enterprise проектах (или нет)
- 10–15 минут vs 2–3 часа анализа крашей
- Автоматизация рутины — освобождение времени

