

Робот + человек: верстаем через Figma MCP быстрее вместе

Ковалёв Кирилл
Frontend-разработчик

Ковалёв Кирилл

Frontend-разработчик

Ростов-на-Дону



- 10 лет в IT
- Интересуюсь сборщиками приложений, производительностью
- Пишу про всякое в ТГ

Робот + человек: верстаем через Figma MCP быстрее вместе

О чем речь?

Какую проблему решаем



Проблема



Целевая схема

Добавить ссылку на figma



Готовый TSX + module.css

Целевая схема
Промт

Собери страницу [PAGE_URL]

Результат


SOMETIMES MAYBE GOOD
SOMETIMES MAYBE SH*T
umbro
nova sports
nova sports
nova sports

Поддержка проекта



Лимиты

Вайбкодеры идут домой в 12:30, потому что закончились лимиты

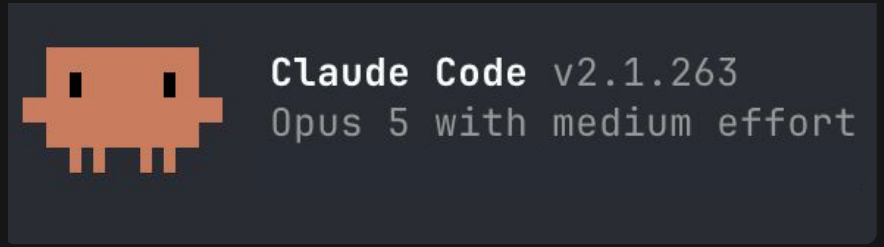


Почему не работает "из коробки"

Инструмент	Что отдаёт
v0, Lovable, Bolt	Свой UI-kit (Tailwind/shadcn), не ваш
Screenshot-to-code	Inline styles, угаданная структура
Figma Make	Свои компоненты, не маппит на код
Figma Dev Mode	`gap: 8px`, не `var(--spacing-2)`

MCP и агент

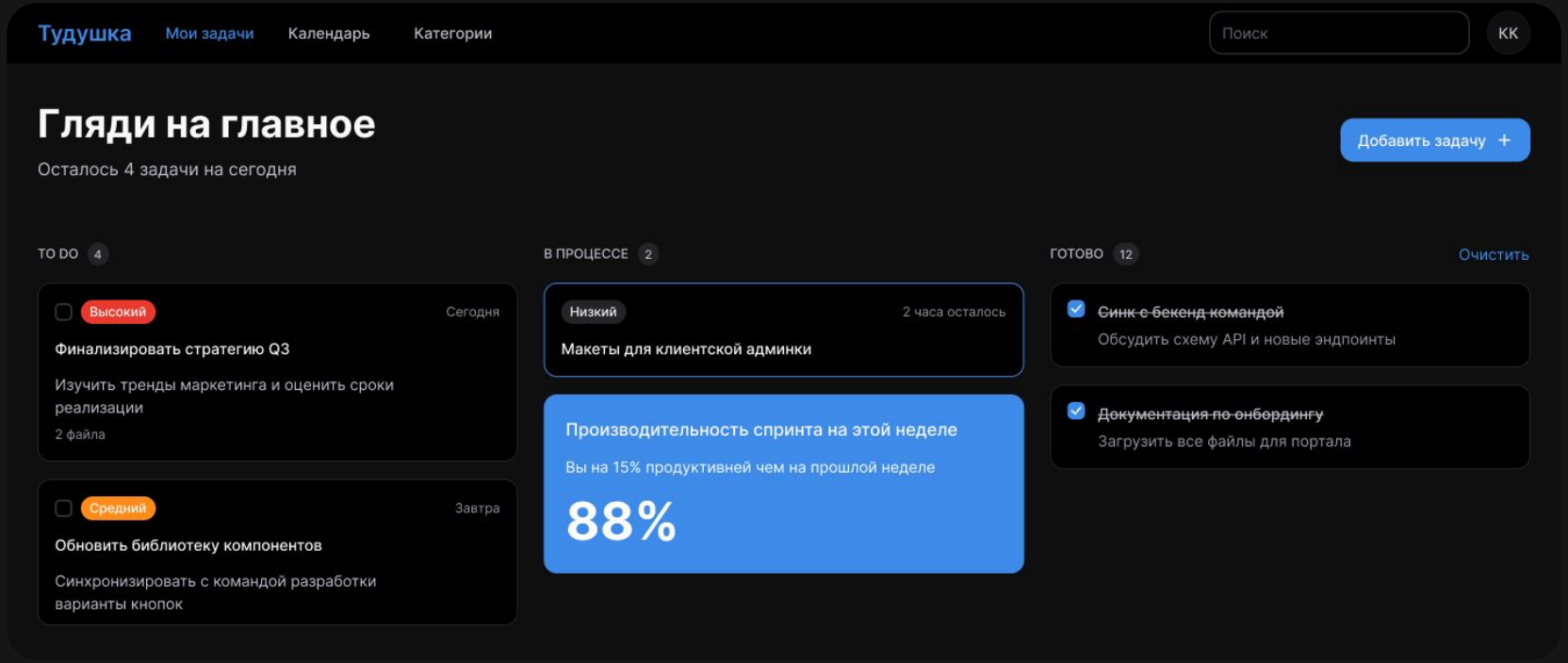
MCP — open-source протокол Anthropic. Сервер выставляет tools, клиент вызывает их по необходимости. Figma MCP — реализация для выделенного фрейма.



Агент — Claude Code в терминале. Figma MCP — локальный, 127.0.0.1:3845, из Figma Desktop.

- Скиллы лежат в репозитории: .claude/skills/
- Тот же набор скилов подключается к другому хосту конфигом

Макет


Тудушка
Мои задачи
Календарь
Категории
Поиск
КК
Гляди на главное
Осталось 4 задачи на сегодня
Добавить задачу +
TO DO 4
В ПРОЦЕССЕ 2
ГОТОВО 12
Очистить
Высокий
Сегодня
Финализировать стратегию Q3
Изучить тренды маркетинга и оценить сроки реализации
2 файла
Средний
Завтра
Обновить библиотеку компонентов
Синхронизировать с командой разработки варианты кнопок
Низкий
2 часа осталось
Макеты для клиентской админки
Производительность спринта на этой неделе
Вы на 15% продуктивней чем на прошлой неделе
88%
Синк с бекенд командой
Обсудить схему API и новые эндпоинты
Документация по онбордингу
Загрузить все файлы для портала

Рабочий флоу



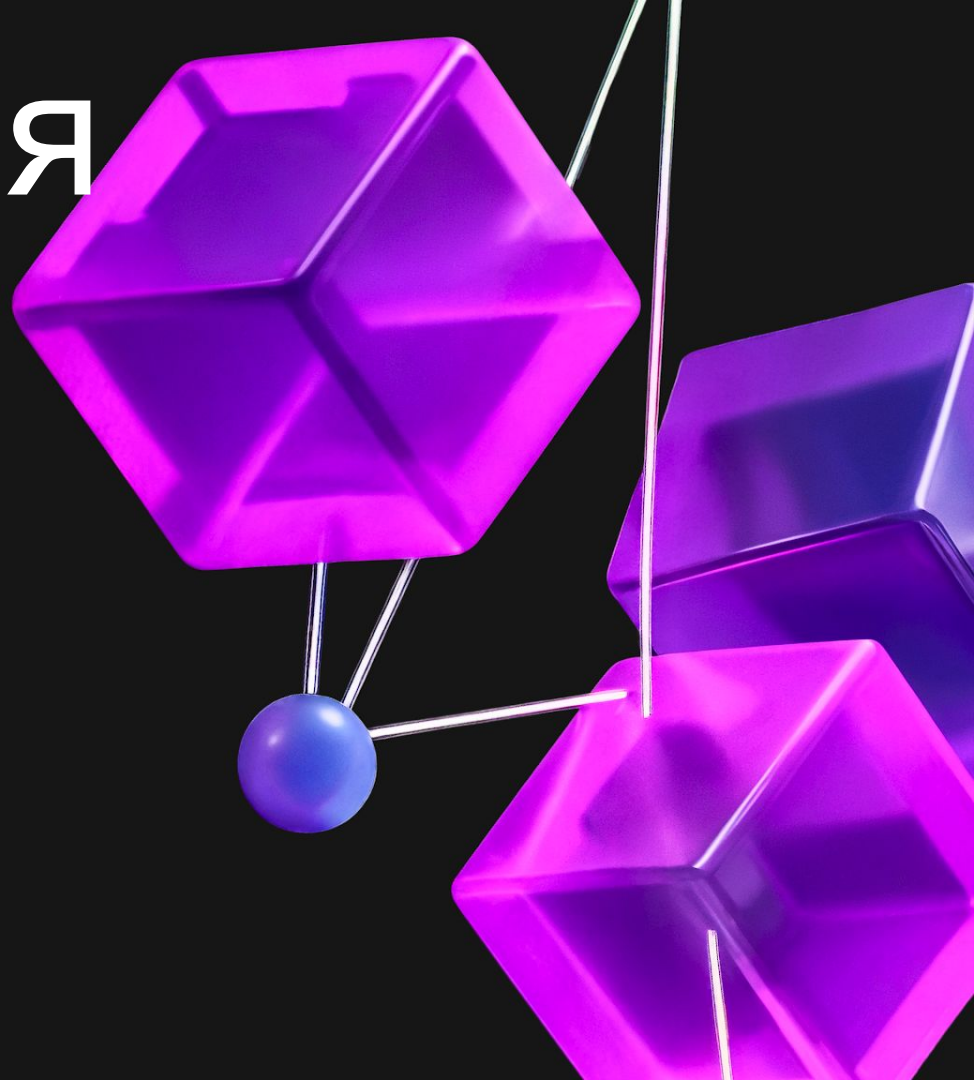
Правильно
настроенный

Figma MCP

Зависит
от команды

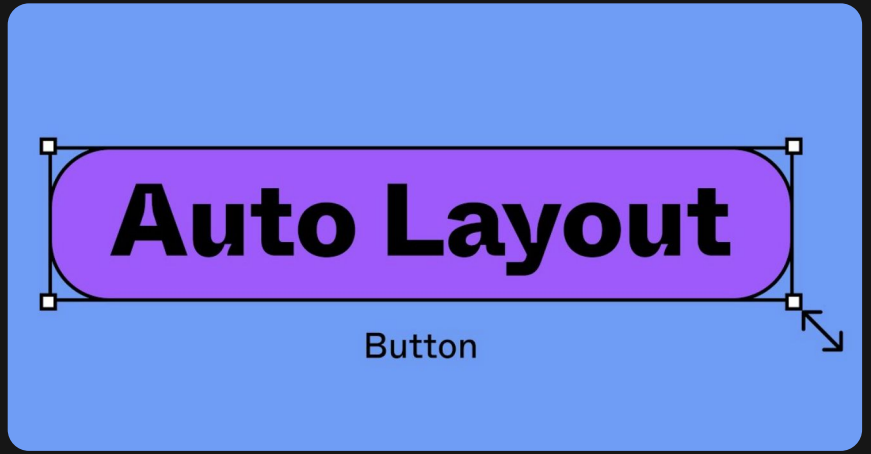
Требования к макету

Почему макет — это первый gate пайплайна



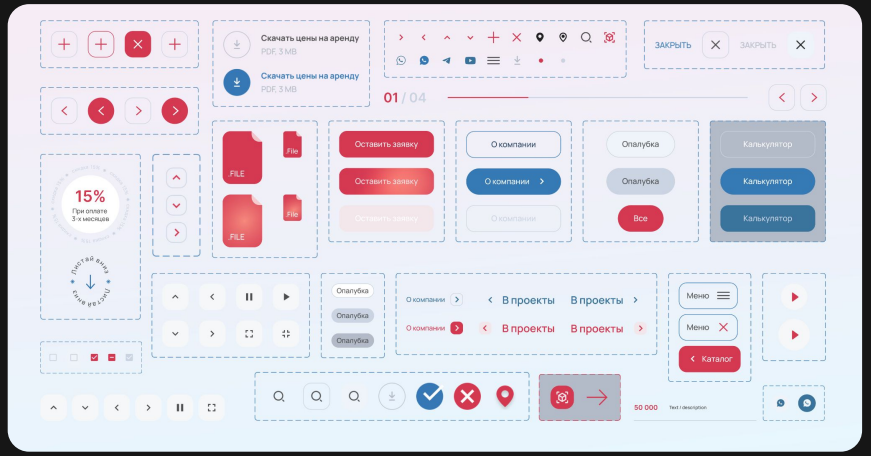
Auto Layout везде

- Каждый контейнер с дочерними элементами — в Auto Layout
- Вложенность Auto Layout соответствует вложенности DOM



Использовать компоненты из библиотеки

- Настоящий компонент из библиотеки — Figma JSON содержит `componentKey`, который однозначно идентифицирует компонент.
- Скопированный фрейм — `componentKey` отсутствует. Агент видит прямоугольник с текстом. Точность падает.
- В design context инстанс узнаётся механически: id слоя начинается с `I` (или у потомка id вида `I<layerId>`). Это правило запишем в контракт



Токены в стилях, не хардкод

- Макет использует те же Figma Variables, которые привязаны к дизайн-токенам ui-kit
- Привязанная переменная — Figma MCP отдаёт её имя, правило `slash → dash` превращает его в CSS-токен без поиска

Name	Light
primary	grey/950
secondary	grey/500
tertiary	grey/400
quaternary	grey/300
on-accent	generic/0

Semantic naming слоёв

Уровень	Паттерн	Пример
Компонент из UI-kit	ComponentName/Variant	Button/Primary, Input/Error
Контейнер-обёртка	Семантическое имя	actions, form-header, card-body
Текстовый элемент	Роль	title, description, label
Иконка	Icon/name	Icon/chevron-right

VariantProperties совпадают с props UI-kit

Чеклист: AI-ready Figma-макет



Структура

- Auto Layout
- Соответствует DOM-структуре
- Нет абсолютного позиционирования там, где его можно избежать



Наименование

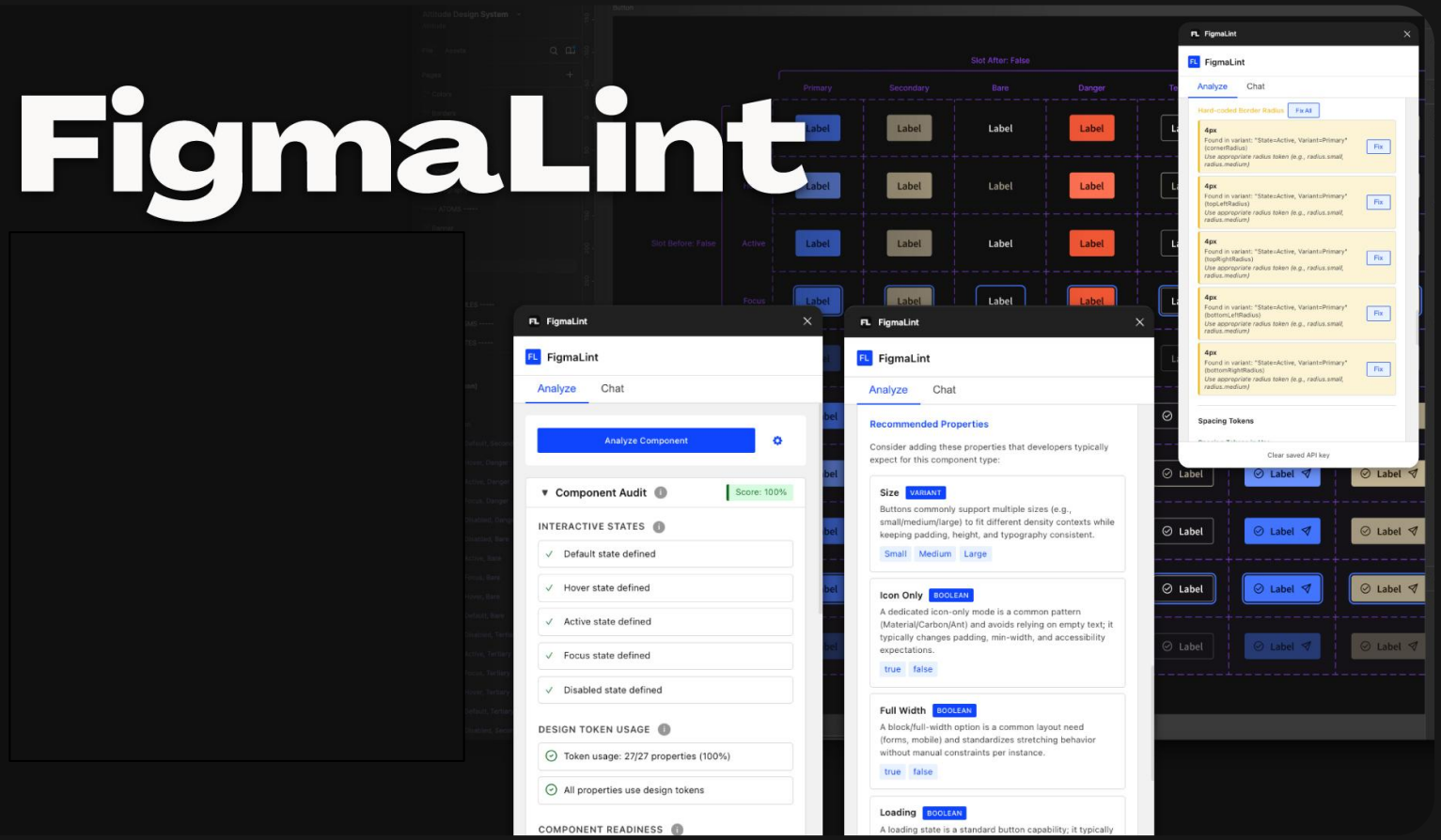
- ComponentName/Variant (Button/Primary)
- Семантические имена (actions, card-body)



Компоненты и токены

- Используются инстансы из библиотеки
- Все цвета, отступы, радиусы — через Figma Variables
- VariantProperties совпадают с ai-map

FigmaLint

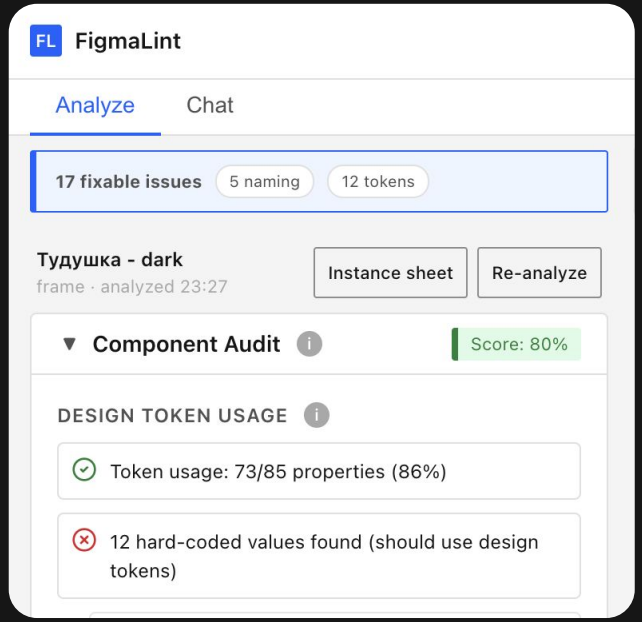

FigmaLint
Analyze
Chat
Analyze Component
Component Audit
Score: 100%
INTERACTIVE STATES
Default state defined
Hover state defined
Active state defined
Focus state defined
Disabled state defined
DESIGN TOKEN USAGE
Token usage: 27/27 properties (100%)
All properties use design tokens
COMPONENT READINESS
Recommended Properties
Consider adding these properties that developers typically expect for this component type:
Size
VARIANT
Buttons commonly support multiple sizes (e.g., small/medium/large) to fit different density contexts while keeping padding, height, and typography consistent.
Small
Medium
Large
Icon Only
BOOLEAN
A dedicated icon-only mode is a common pattern (Material/Carbon/Ant) and avoids relying on empty text; it typically changes padding, min-width, and accessibility expectations.
true
false
Full Width
BOOLEAN
A block/full-width option is a common layout need (forms, mobile) and standardizes stretching behavior without manual constraints per instance.
true
false
Loading
BOOLEAN
Spacing Tokens
Clear saved API key

FigmaLint: чеклист, который проверяет плагин

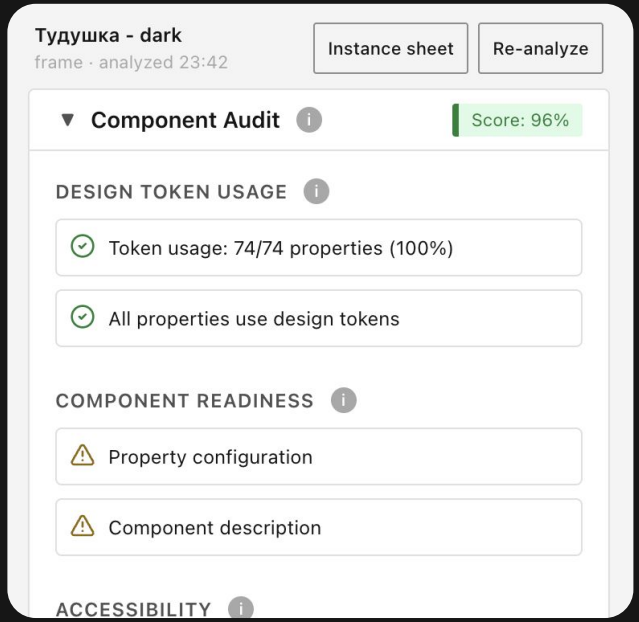
Пункт чеклиста	Паттерн
Инстансы из библиотеки	Detached-инстансы на странице; фреймы с именем компонента, отвязанные от него
Токены через Variables	Захардкоженные цвета, отступы, радиусы, типографика vs привязанные переменные
Семантические имена	Generic-имена «Frame 1», «Rectangle 4» + предложение семантического
Состояния компонента	Отсутствующие hover / focus / disabled / pressed
Итог	Readiness score: токены ×2, состояния ×3, доступность (контраст, touch target)

FigmaLint: до auto-fix → после

До



После



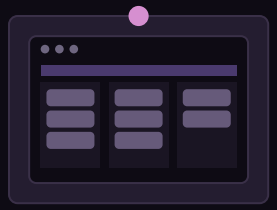
Требования к UI-kit

Почему UI-kit — это контракт для агента





Что видит агент?



get_design_context

figma-2159-2910.tsx

```
const imgCheckbox = "http://localhost:3845/assets/d4b6…svg";

export default function Dark() {
  return (
    <div className="relative size-full"
      data-node-id="2159:2910" data-name="Тудушка - dark">
      <div className="absolute
bg-[var(--neutrals\/background,#0f0f10)]
        flex flex-col left-0 top-0 w-[1440px]"
        data-node-id="2159:2911" data-name="Todo List">
        <div className="bg-[var(--neutrals\/surface,black)] …
h-[56px]"
          data-node-id="2159:2912" data-name="Navbar">
            …
```

4

```
SUPER CRITICAL: The generated React+Tailwind code MUST
be converted to match the target project's technology stack…
DO NOT install any Tailwind as a dependency…
These styles are contained in the design: Display/MD: Font(…)
Image assets are stored on a localhost server…
```

1. Макет пришёл как React-компонент
2. Адрес и имя каждого слоя
3. Стили — Tailwind-классы: внутри переменная Figma и значение
4. Инструкции для модели

С правильным макетом агент не угадывает

TypeScript как единственный источник истины

- "strict": true в tsconfig — обязательно
- Все variant props — union types, никаких string
- JSDoc с @example на каждом пропе
- Никаких any, минимум unknown

```ts
export interface ButtonProps {
  /** Визуальный стиль кнопки @example variant="primary" */
  variant: 'primary' | 'secondary';
  /** Размер кнопки @example size="md" */
  size: 'sm' | 'md' | 'lg';
  disabled?: boolean;
  loading?: boolean;
  children: React.ReactNode;
}
```

Токены — трёхуровневая иерархия

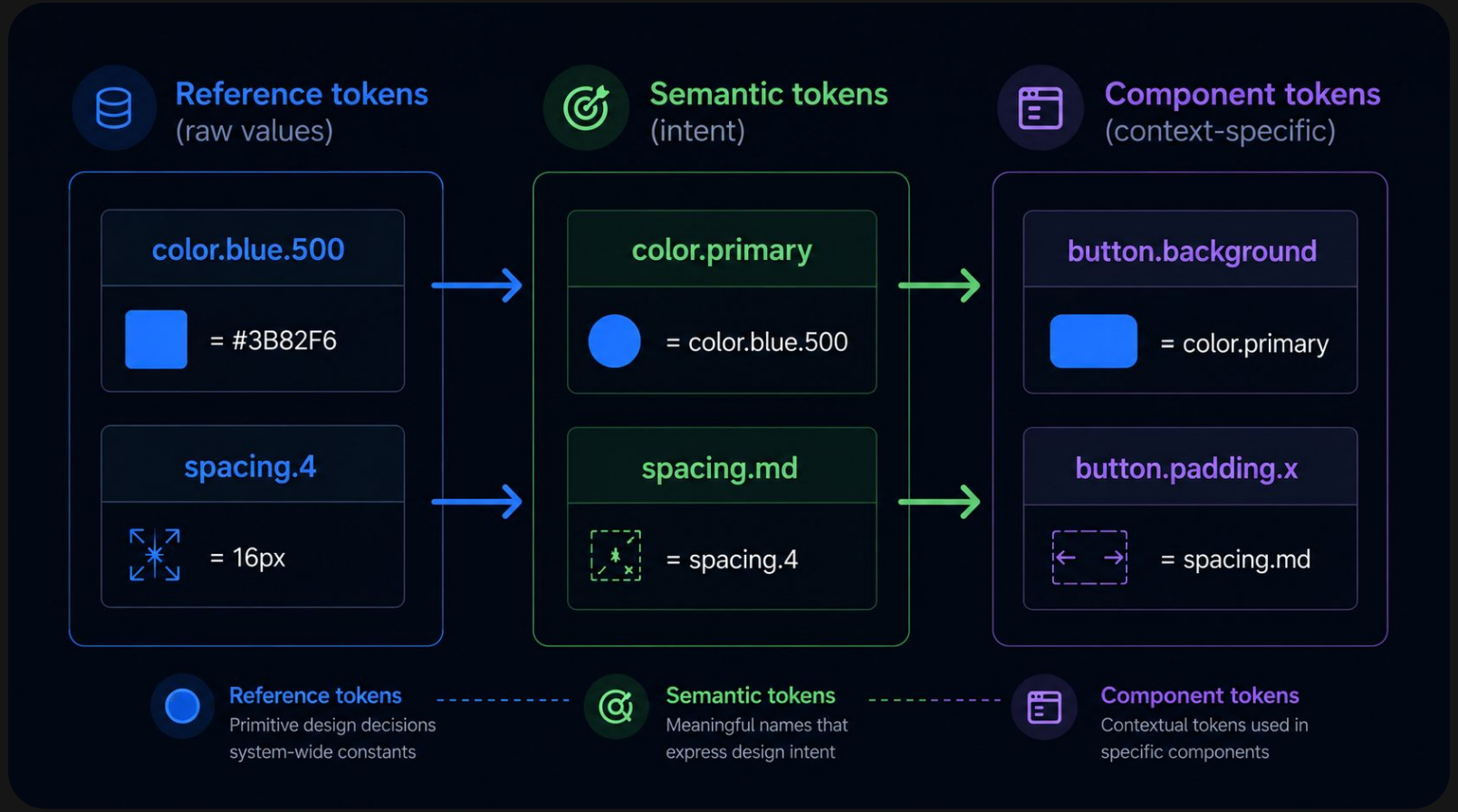
- Reference — сырые значения, в компонентах не используются
- Semantic — намерение, используются в компонентах
- Component — точечная настройка
- CSS ui-kit генерируется из Figma Variables: один токен = одна переменная Figma

```css
/* Reference */
:root { --color-blue-500: #3B82F6; }

/* Semantic */
:root { --color-primary: var(--color-blue-500); }

/* Component */
:root {
--button-background:var(--color-primary);
}

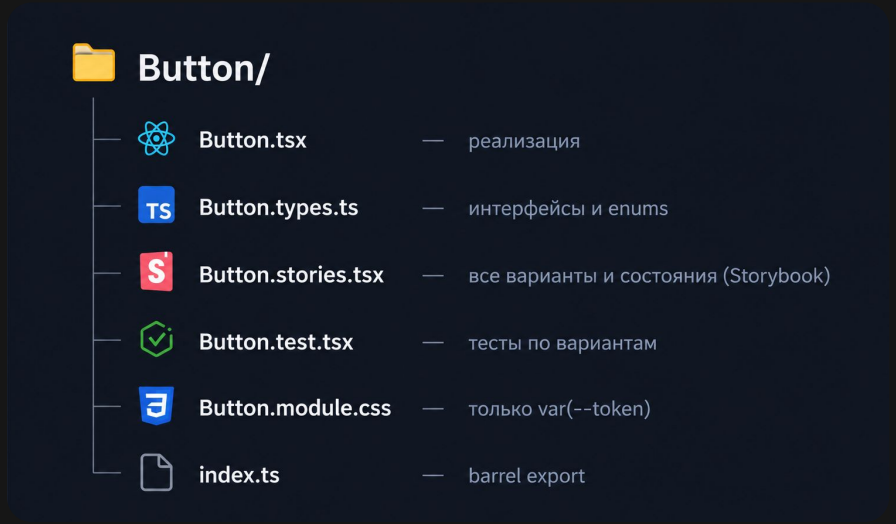
.button { background: var(--button-background); }
```

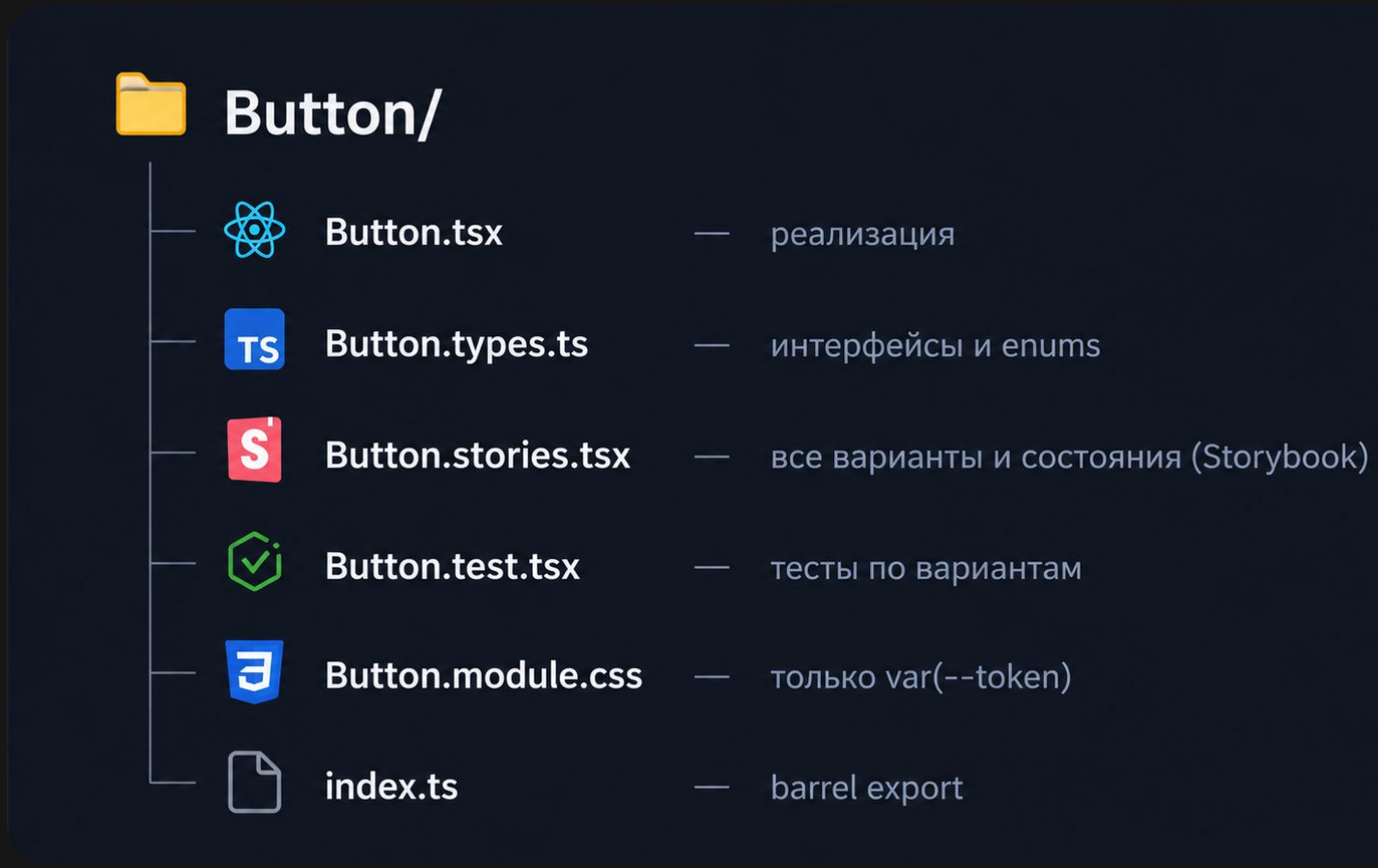

Reference tokens
(raw values)
Semantic tokens
(intent)
Component tokens
(context-specific)
color.blue.500
= #3B82F6
spacing.4
= 16px
color.primary
= color.blue.500
spacing.md
= spacing.4
button.background
= color.primary
button.padding.x
= spacing.md
Reference tokens
Primitive design decisions
system-wide constants
Semantic tokens
Meaningful names that
express design intent
Component tokens
Contextual tokens used in
specific components

Имена состояний Figma = TypeScript

Component Contract: файловая структура

- Каждый компонент UI-kit — отдельная директория с явным контрактом




Button/
Button.tsx — реализация
Button.types.ts — интерфейсы и enums
Button.stories.tsx — все варианты и состояния (Storybook)
Button.test.tsx — тесты по вариантам
Button.module.css — только var(--token)
index.ts — barrel export

Чеклист: AI-ready UI-kit



TypeScript

- Strict mode включён
- Все variant props — union types, не string
- JSDoc с @example на каждом пропе
- Нет any, нет prop explosion



Токены

- Трёхуровневая иерархия
- Компоненты используют semantic/component токены
- токены доступны через Figma Variables

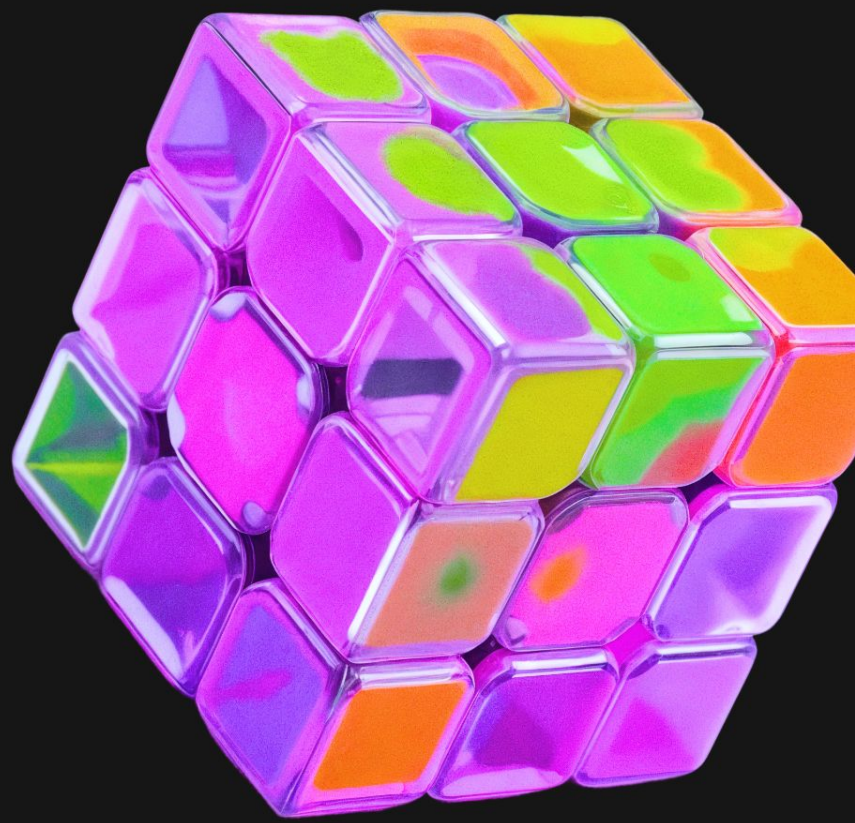


Маппинг и синхронизация

- figmaNodeId зафиксирован для каждой записи
- State matrix совпадает в Figma

Контракты и скиллы

Договоренности на уровне кода и AI



Как синхронизировать React и Figma?

4 эксперимента

Ева

Простой промт

Простые правила и инструкции

Руфь

MD файл

Маппинг на уровне проекта

Лия

AI-map

Настройки на уровне кода ui-kit

Эсфирь

AI-kit

Настройки на уровне кода проекта

Sonnet 4.7 (май 2026)

Эксперимент	Время работы минуты	Лимит токенов % от лимита за 5 ч	Подготовка мин
Ева простой промт	15	9	5
Руфь MD-файл	5	8	90
Лия AI-map: ui-kit	9	9	30
Эсфирь AI-kit: проект	4	6	150

Время работы Лимит токенов Подготовка

Цвет — внутри своего столбца: меньшее значение = насыщеннее = лучше.

Sonnet 4.7 (май 2026)

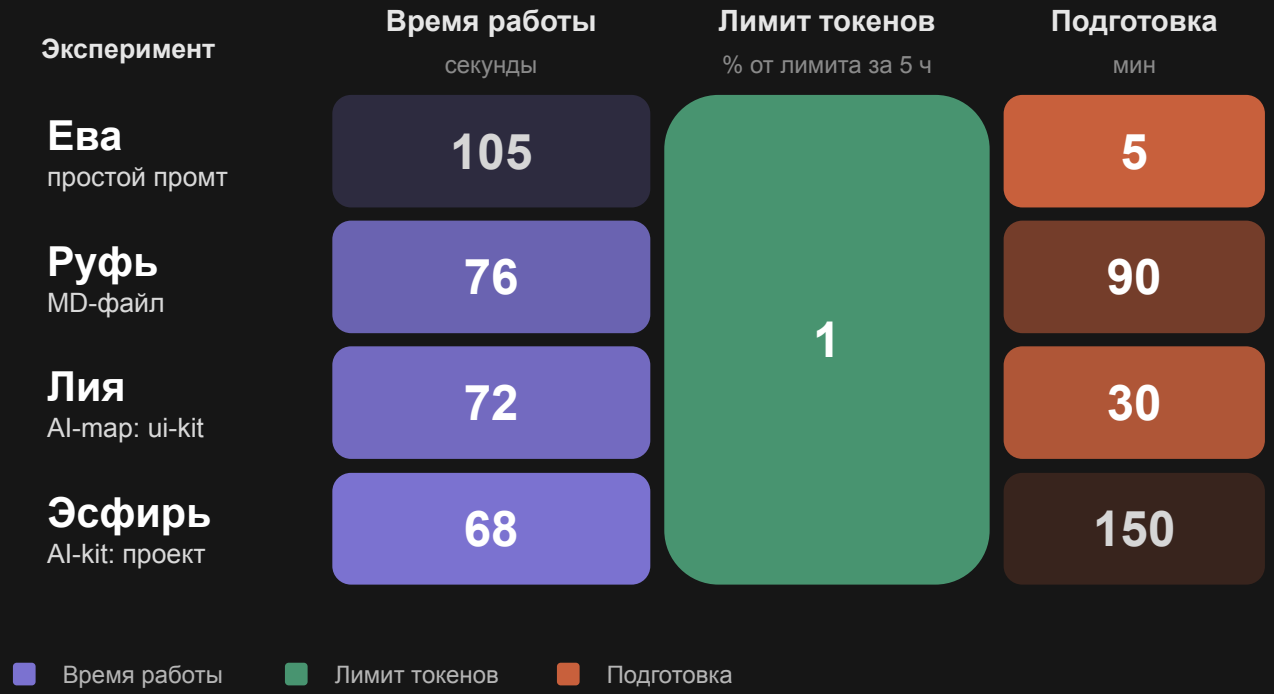
Эксперимент	Время работы минуты	Лимит токенов % от лимита за 5 ч	Подготовка мин
Ева простой промт	15	9	5
Руфь MD-файл	5	8	90
Лия AI-map: ui-kit			30
Эсфирь AI-kit: проект	4	6	150

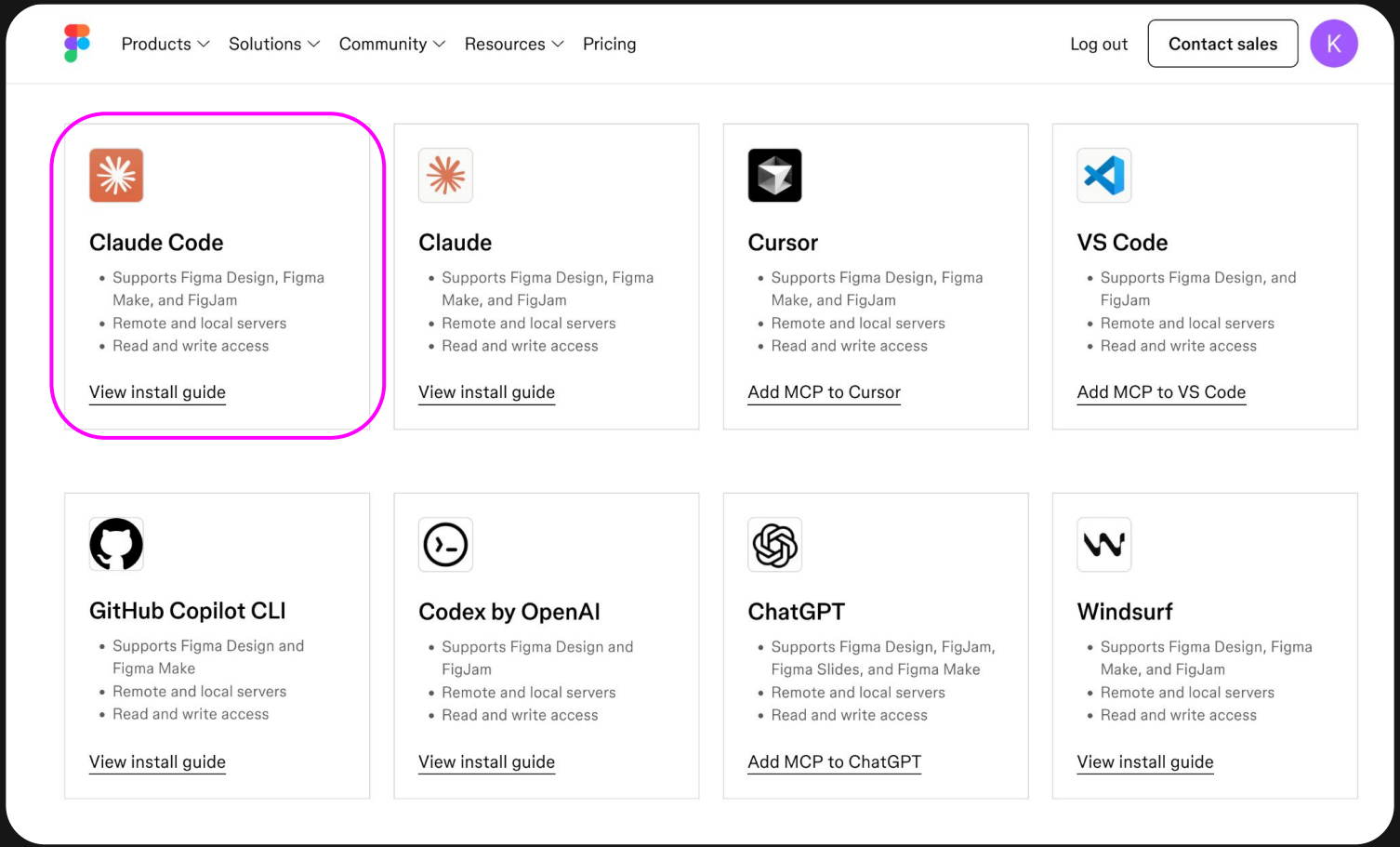


Время работы Лимит токенов Подготовка

Цвет — внутри своего столбца: меньшее значение = насыщеннее = лучше.

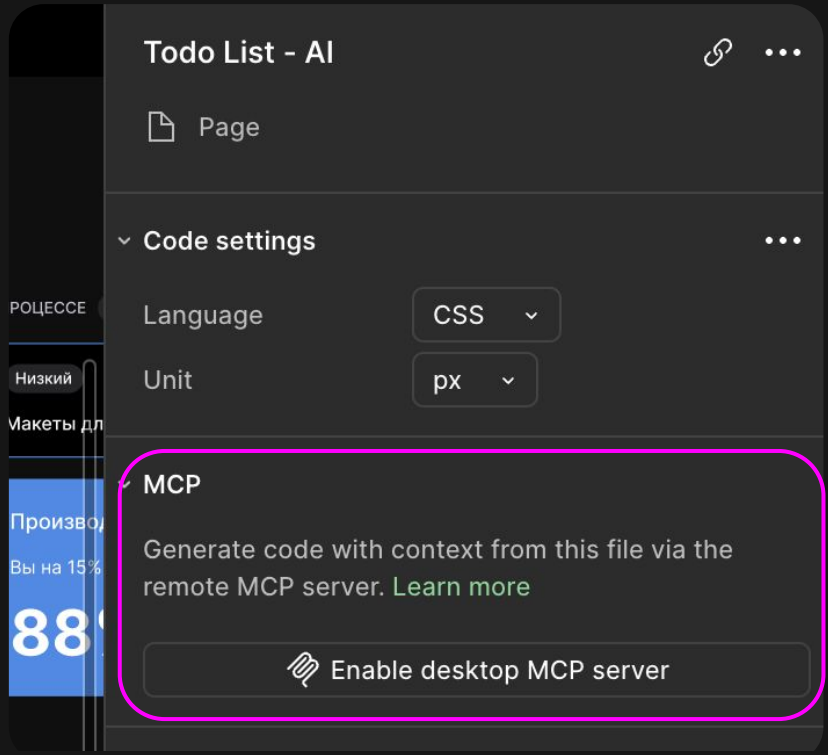
Sonnet 5.5 (октябрь 2026)




Products
Solutions
Community
Resources
Pricing
Log out
Contact sales
K
Claude Code
• Supports Figma Design, Figma Make, and FigJam
• Remote and local servers
• Read and write access
View install guide
Claude
• Supports Figma Design, Figma Make, and FigJam
• Remote and local servers
• Read and write access
View install guide
Cursor
• Supports Figma Design, Figma Make, and FigJam
• Remote and local servers
• Read and write access
Add MCP to Cursor
VS Code
• Supports Figma Design, and FigJam
• Remote and local servers
• Read and write access
Add MCP to VS Code
GitHub Copilot CLI
• Supports Figma Design and Figma Make
• Remote and local servers
• Read and write access
View install guide
Codex by OpenAI
• Supports Figma Design and FigJam
• Remote and local servers
• Read and write access
View install guide
ChatGPT
• Supports Figma Design, FigJam, Figma Slides, and Figma Make
• Remote and local servers
• Read and write access
Add MCP to ChatGPT
Windsurf
• Supports Figma Design, Figma Make, and FigJam
• Remote and local servers
• Read and write access
View install guide

Как включить Figma MCP

- Figma Desktop, Dev или Full seat
- Панель Inspect → раздел MCP → Enable desktop MCP server
- Сервер локальный: 127.0.0.1:3845
- В Claude Code: mcp__figma-desktop__* — только он, remote в скилах запрещён



Где хранить контракт?


БИБЛИОТЕКА
ПРОЕКТ

Целевая схема

Добавить ссылку на figma

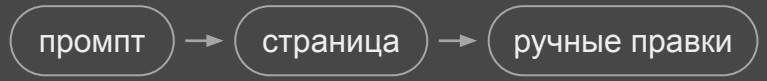


Готовый TSX + module.css

Чиним агента, а не страницу

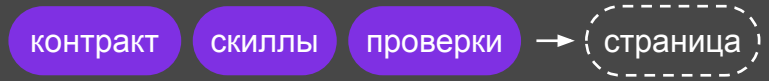
Страницу верстает агент. Если результат кривой, чиним не саму страницу, а то, что агент читает, как он работает и чем мы его проверяем.

Раньше



Один промпт и справочник на 583 строки. Агент угадывал, результат плавал.

Теперь



Страница — одноразовый артефакт. Долговечны контракт, скиллы и проверки.

Три слоя вокруг агента

Знания, процесс и контроль разведены по слоям. Агент лишь связывает их.

Figma →

1
Знания
как устроен ui-kit и токены

.ai-kit/

2
Процесс
что делать и в каком порядке

.claude/skills/ figma-page-gen · sync-ai-map

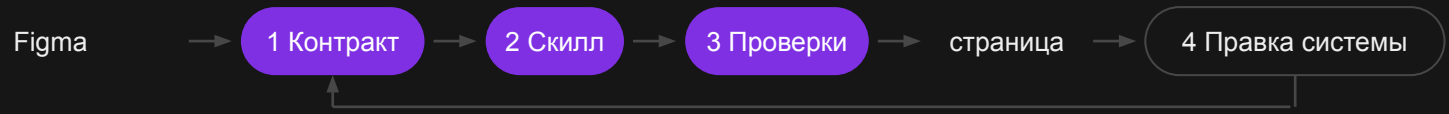
3
Контроль
что считать готовым

.ai-kit/bin/ скрипты-проверки

→ **Страница**

src/pages/<Page>/

наружу экспортируется только страница



Контракт: дизайн-система для ИИ

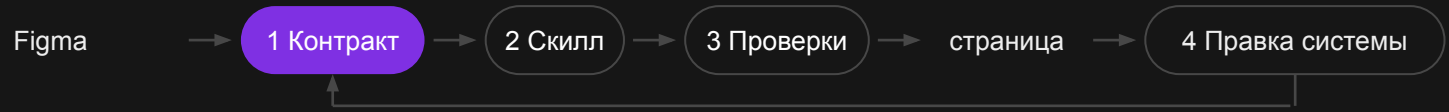
Всё, что можно сгенерировать, генерируется, а не пишется руками.

.ai-kit/

- kit.json — пишем
- <Component>.kit.md × 8 — пишем

kit.json · <Component>.kit.md

Реестр компонентов Figma: implemented · shim · tokens-only · missing.
Пропсы и маппинг «вариант Figma → проп» для 8 компонентов.



Контракт: дизайн-система для ИИ

Всё, что можно сгенерировать, генерируется, а не пишется руками.

.ai-kit/

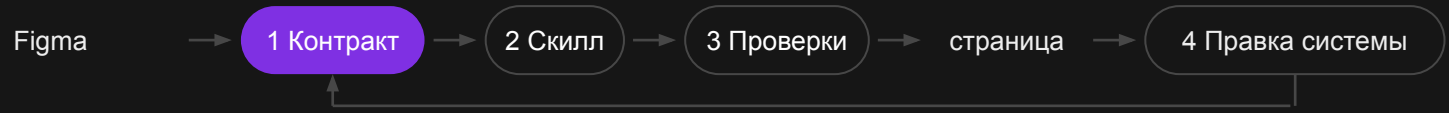
- kit.json — пишем
- <Component>.kit.md × 8 — пишем
- tokens.txt — генерируем

kit.json · <Component>.kit.md

Реестр компонентов Figma: implemented · shim · tokens-only · missing.
Пропсы и маппинг «вариант Figma → проп» для 8 компонентов.

tokens.txt

NAME · TIER · REF · LIGHT · DARK генерируются из CSS ui-kit.
Цепочки var() раскрыты, в исходники лезть не нужно.



Контракт: дизайн-система для ИИ

Всё, что можно сгенерировать, генерируется, а не пишется руками.

.ai-kit/

- kit.json — пишем
- <Component>.kit.md × 8 — пишем
- tokens.txt — генерируем
- sync-report.md — генерируем

src/shims/ — долг

kit.json · <Component>.kit.md

Реестр компонентов Figma: implemented · shim · tokens-only · missing.
Пропсы и маппинг «вариант Figma → проп» для 8 компонентов.

tokens.txt

NAME · TIER · REF · LIGHT · DARK генерируются из CSS ui-kit.
Цепочки var() раскрыты, в исходники лезть не нужно.

src/shims/ · /sync-ai-map

Техдолг ui-kit (Button ghost). У каждого shim есть условие удаления.
/sync-ai-map сверяет контракт с Figma и типами → sync-report.md.

Figma → 1 Контракт → 2 Скилл → 3 Проверки → страница → 4 Правка системы

Скилл: «не угадывай»

Всё, что можно вычислить, считает скрипт: он не придумает лишнего.
Модель только собирает страницу.

скрипт | **модель**

preflight → чтение → запись → проверка

Preflight — один вызов

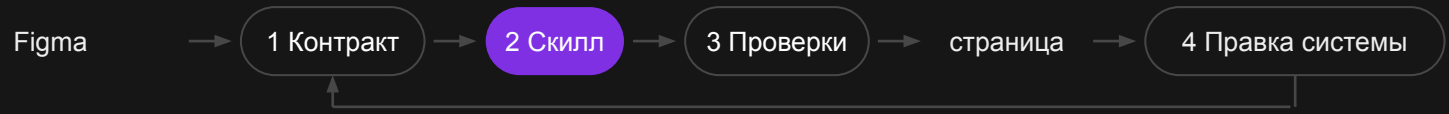
Обновляет токены, проводит аудит и считает figma-digest узла.

Читаем всё, пишем раз

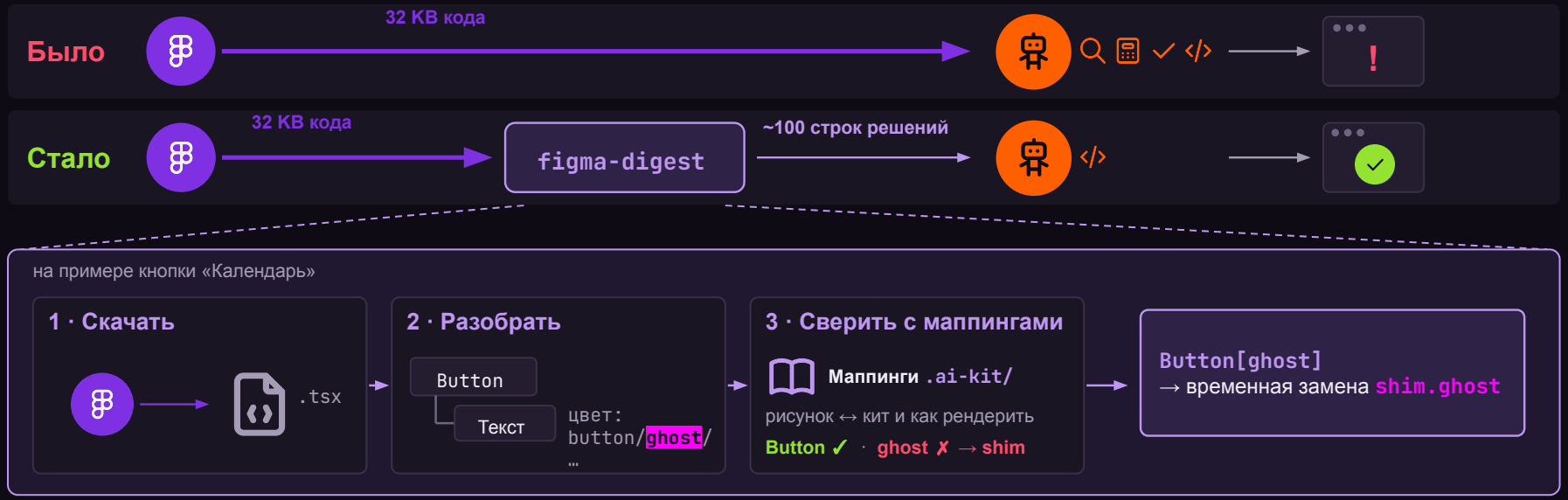
Пока всё нужное не прочитано, ничего не записываем. Каждый файл создаём один раз и сразу целиком.

Решения механические

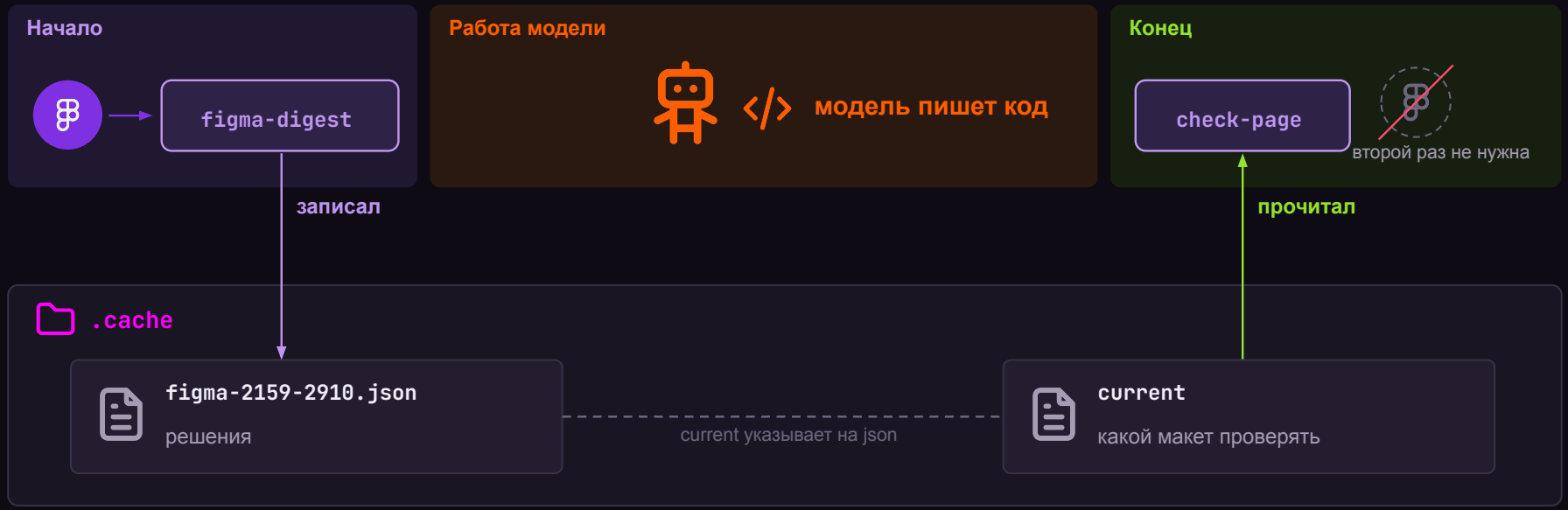
Компонент из UI-кита, shim или div, вариант Typography, вынос в компонент: на всё есть правило.



Figma-digest: разбор макета



Figma-digest: разбор макета



Когда «готово» решают скрипты

Прогон закончен, только когда три проверки вернули 0

Типы

tsc -p tsconfig.app.json

Без -p tsc проверяет 0 файлов из 12 и говорит, что всё чисто.

Токены

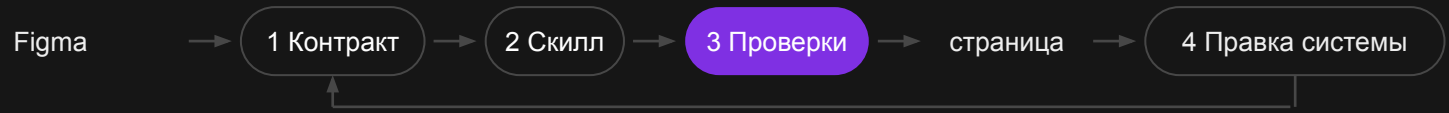
check-tokens

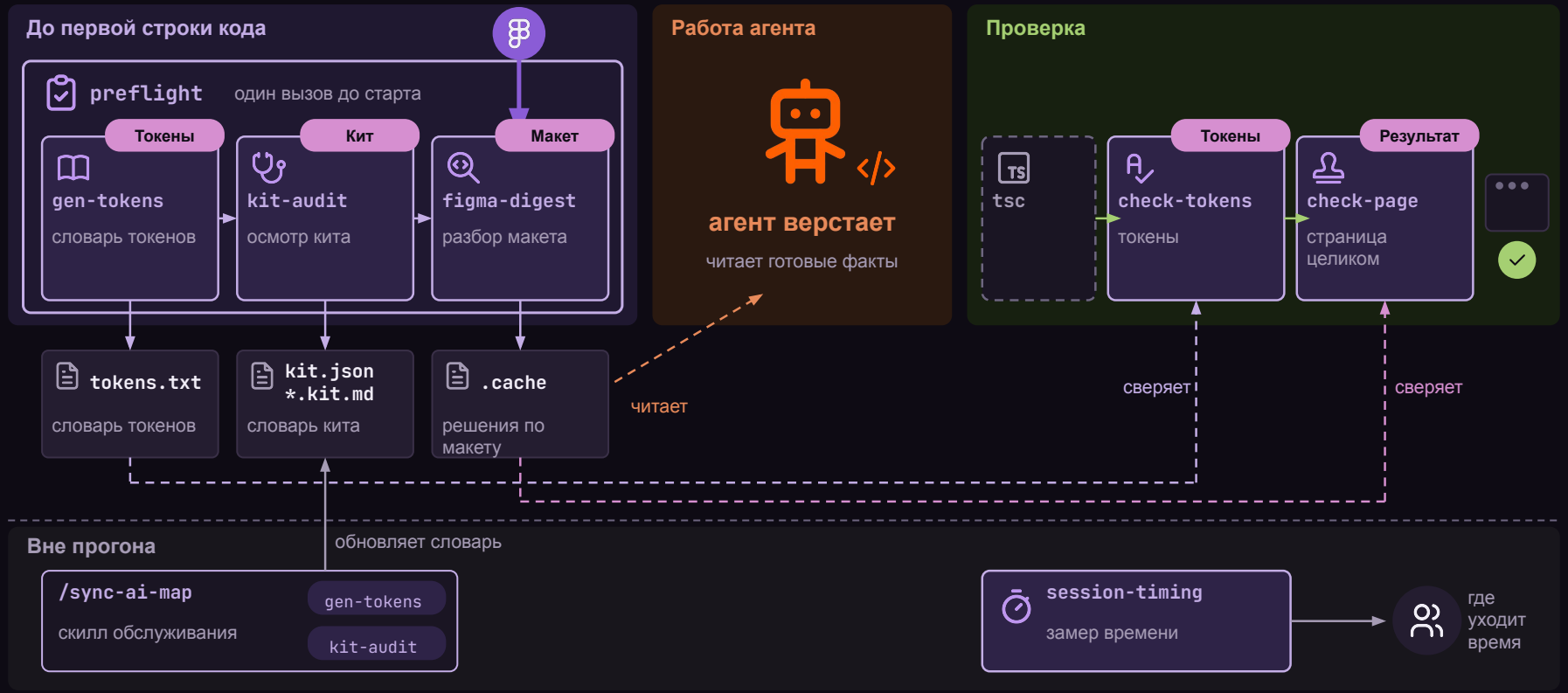
Каждый var(--x) есть в tokens.txt, захардкоженных значений нет.

Структура

check-page

Сверяет вёрстку с Figma: весь текст через Typography, без inline-стилей и лишних экспортов.




До первой строки кода
preflight
один вызов до старта
Токены
gen-tokens
словарь токенов
Кит
kit-audit
осмотр кита
Макет
figma-digest
разбор макета
Работа агента
агент верстает
читает готовые факты
Проверка
tsc
Токены
check-tokens
токены
Результат
check-page
страница целиком
tokens.txt
словарь токенов
kit.json
*.kit.md
словарь кита
.cache
решения по макету
читает
сверяет
сверяет
Вне прогона
обновляет словарь
/sync-ai-map
скилл обслуживания
gen-tokens
kit-audit
session-timing
замер времени
где уходит время

Петля улучшения

Каждая ошибка агента превращается в правило, а лучше в скрипт.
Агент не умнеет от прогона к прогону — система вокруг него становится жёстче.



Ошибка агента → что стало

Пропущенный шаг (Haiku)	→	шаг внутри preflight
Придуманное имя токена	→	check-tokens
Выдуманный отчёт	→	check-page --report
Дубль скриншота	→	запрет в скилле
12 правок после черновика	→	«пиши файл один раз»

Figma → 1 Контракт → 2 Скилл → 3 Проверки → страница → 4 Правка системы

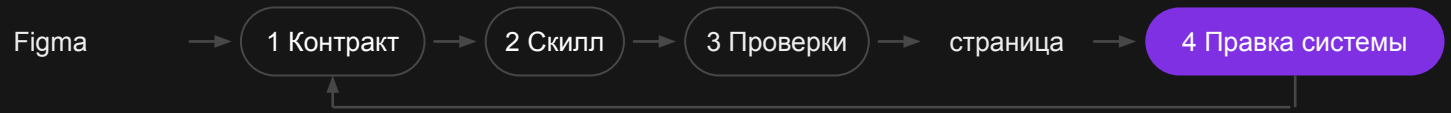
Петля улучшения

Что получили

- Todooshka — модуль из 7 компонентов, наружу одна страница.
- Для форм — form-gen и ADR 0001.

Что забрать себе

- Контракт для ИИ
- Скрипты вместо инструкций

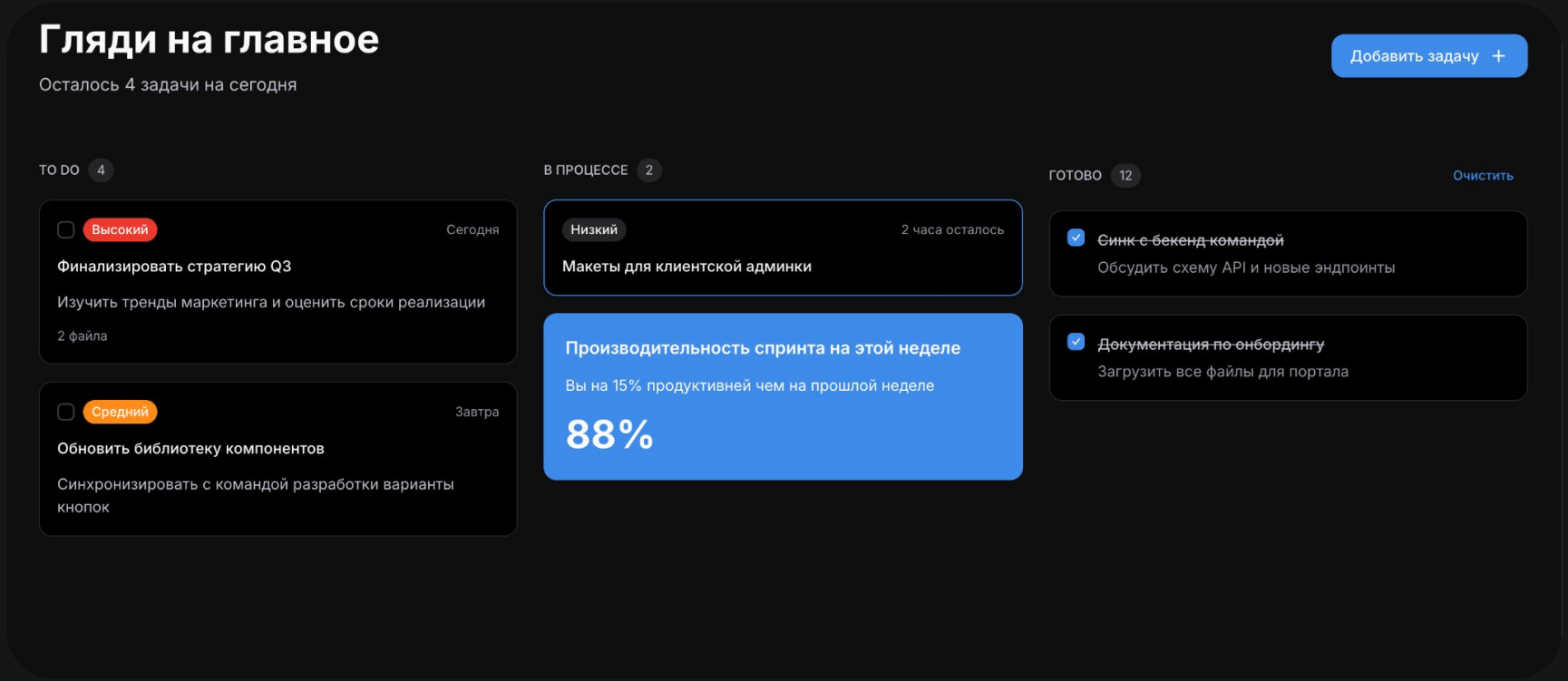


Целевая схема
Промт

Собери страницу [PAGE_URL]

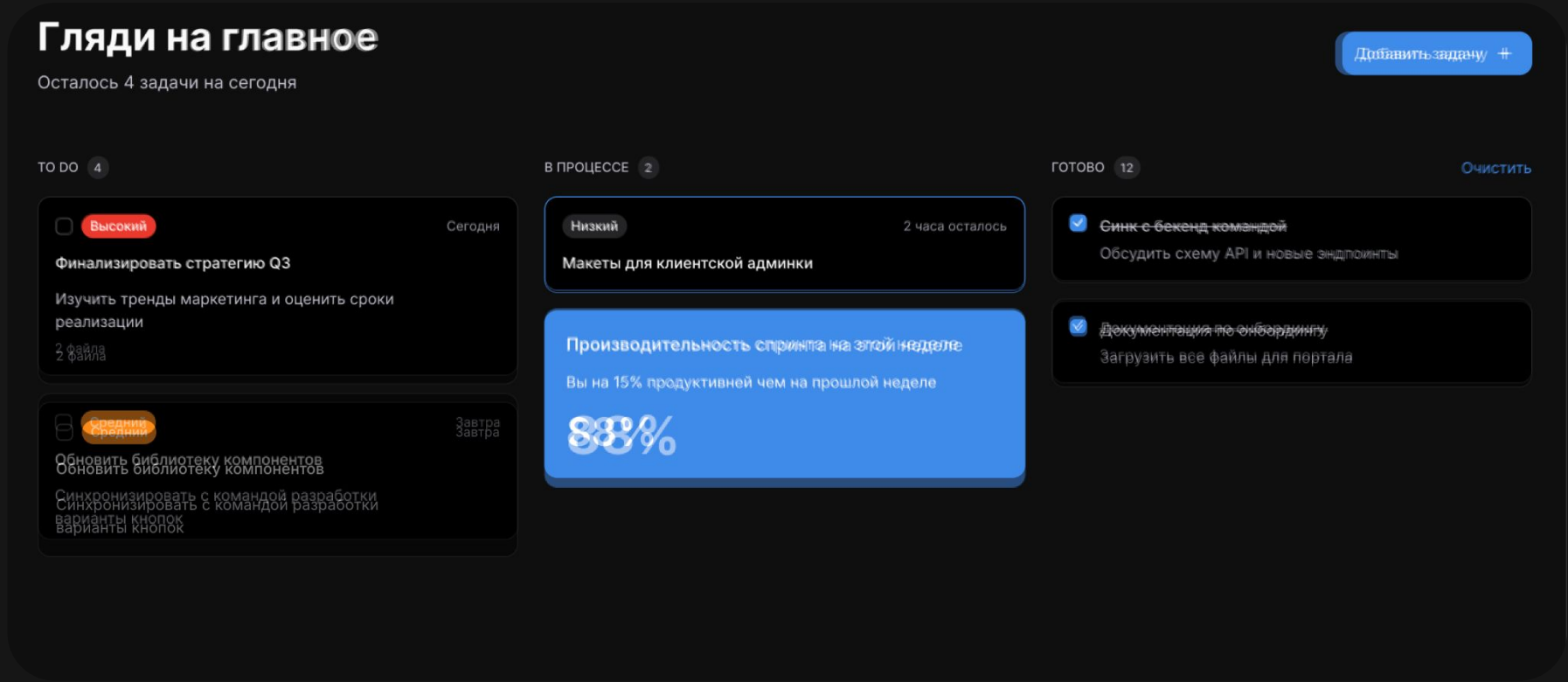


Результат


Гляди на главное
Осталось 4 задачи на сегодня
Добавить задачу +
TO DO 4
Высокий
Сегодня
Финализировать стратегию Q3
Изучить тренды маркетинга и оценить сроки реализации
2 файла
Средний
Завтра
Обновить библиотеку компонентов
Синхронизировать с командой разработки варианты кнопок
В ПРОЦЕССЕ 2
Низкий
2 часа осталось
Макеты для клиентской админки
Производительность спринта на этой неделе
Вы на 15% продуктивней чем на прошлой неделе
88%
ГОТОВО 12
Очистить
~~Синк с бекенд командой~~
Обсудить схему API и новые эндпоинты
~~Документация по онбордингу~~
Загрузить все файлы для портала



Pixel Perfect




DEV

Один вызов против трёх

Figma отдаёт всё одним ответом. Pixso — по инструменту на вопрос

Figma

`get_design_context`

один вызов → код React + Tailwind

Большой фрейм → только метаданные, варианты достаём из имён символов

Pixso

`get_variants`	→	матрица вариантов
`get_screenshot`	→	как выглядит макет
`design_to_code`	→	папка проекта — не сохранять только справка: раскладка и токены

Что приходит в ответе

Те же ноды, другой формат

	Figma	Pixso
ID в URL	`node-id=1-12018`	`item-id=1:12018`
ID после импорта	—	**совпадают ✓** маппинги не пришлось переписывать
Варианты	разбирать из кода	`Variant=Primary, Size=XL, State=Default`
Регистр значений	—	TitleCase → сверка без учёта регистра
Токены	`var(--neutrals\/surface)` → чинить слэш	CSS `design_to_code` уже с `dash`

Что поменялось в проекте

Контракт отвязали от инструмента

.ai-kit

Было	Стало
~~figmaNodeId~~	nodeId
~~figmaLayer~~	layer
~~figmaVariant~~	variantProperty
~~figmaState~~	stateProperty
~~@figma* в JSDoc~~	@design*

kit.json `designTool: "pixso"`

figma-page-gen → pixso-page-gen

screenshot → `design_to_code` (справка) → маппинг → `get_variants` → код

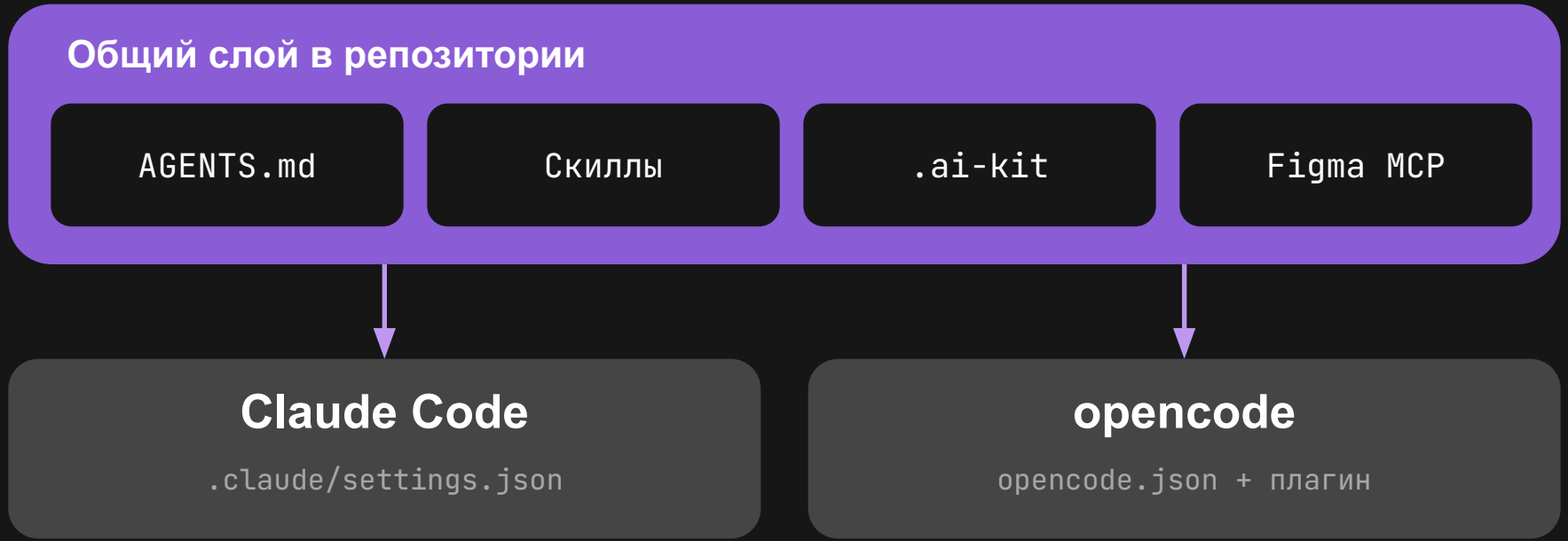
sync-ai-map

читает только `get_variants`. Typography и ButtonGroup пропускаются без вызова MCP


DEV
opencode

Один контракт — два агента

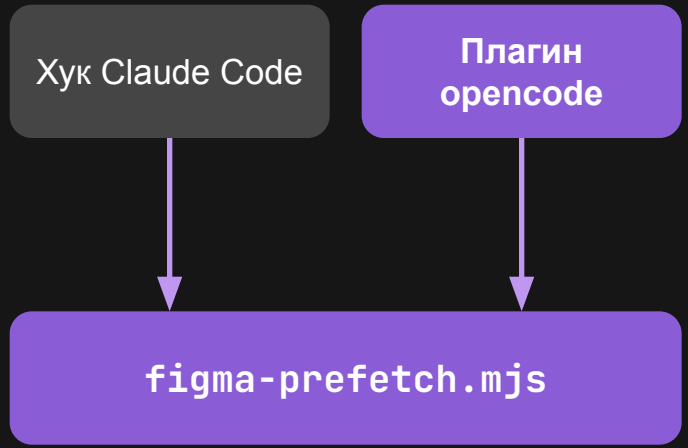
Всё, что знает про проект, лежит в репозитории и не зависит от агента.



Расходится только тонкий адаптер

Что отличается и как адаптировали

Параметр	Claude Code	opencode
Конфиг	.claude/settings.json	opencode.json
Хуки	JSON-конфиг	**JS-плагин**
Вызов скилла	имя и аргументы	**только имя**



Логика не дублируется: меняется только адаптер, а общий скрипт и контракт остаются одни.

Выводы

Что же у нас получилось?

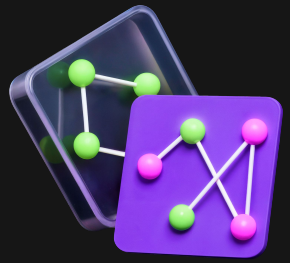


Итого



AI следовал инструкциям

Не галлюцинировал, выдавал production-ready результат



Для разных проектов

Подходит как большим компаниям так и командам по 5-10 человек



Целевая схема заработала

Ctrl + C → Ctrl +V подход в действии


AI
Разработчики

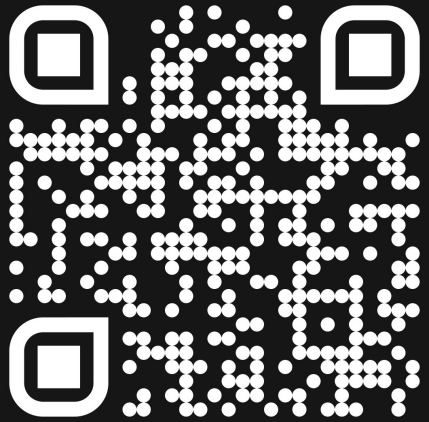
Открытые вопросы

- Адаптивная верстка
- Few-shot компонентный подход
- Абсолютное позиционирование
- Анимации





Спасибо!
Вопросы?



Подпишись
TG @kovalev_frontend

Ковалёв Кирилл

Frontend-разработчик