

# Как закрывать агентами весь SDLC

Что нужно, чтобы агент закрывал весь цикл разработки  
до деплоя в продакшен

# Павел Федотовский

AI Engineering Lead в Яндексе

- **Больше 10 лет в .NET**  
на разных инженерных позициях.
- **2 года в AI-кодинге.**
- **Продвигаю AI-инструменты**  
и строю бенчмарки агентов.



# Все инженерные роли → AI-лид



Если ваша роль заканчивается на коде — агент скоро вас заменит.

# Для кого доклад и где применить

## Для кого доклад

**Начинаете или используете агента только для кода**

Поймёте, как пройти путь от задачи до production.

**Уже используете агента во всём SDLC**

Сверите подход и найдёте улучшения.

## Где применить

**Проще начать с сервиса или небольшого набора микросервисов**

- Понятный контекст
- Быстрые проверки
- Независимый деплой

Другие системы — тоже можно, но сложнее

# Эксперимент: сервис целиком через агента

## Условия

Создать сервис с нуля

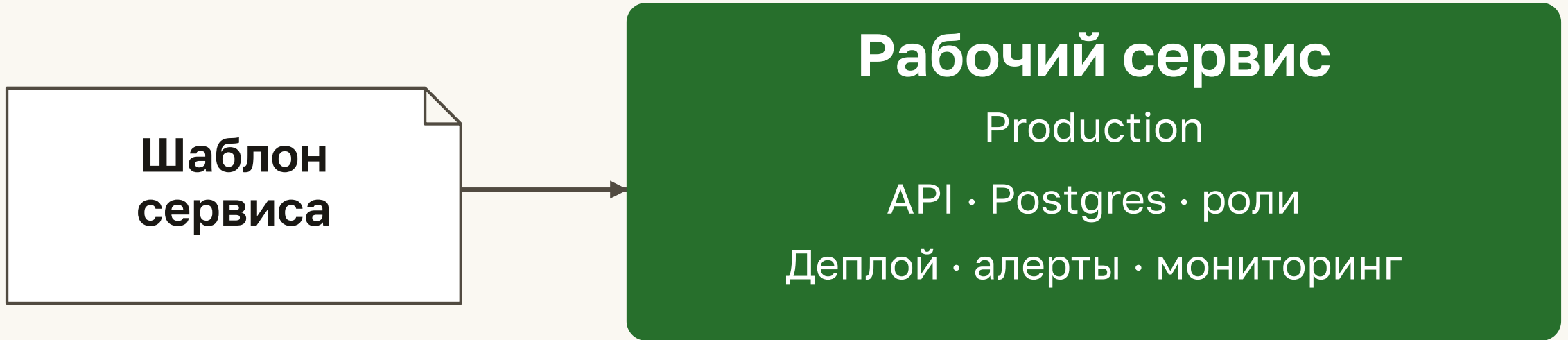
- только через промпты агенту
- незнакомый язык и инфраструктура

## Агент сам

- проектирует и пишет код
- разворачивает сервис и базу
- настраивает логи и алерты
- разбирается с проблемами в production

Май 2026

# Агент прошёл весь путь



реальная задача агенту

```
> I've got multiple alerts on PROD,  
please investigate.
```

# Результаты

2–3 недели

≈ 20%

работа над  
сервисом

≈ 80%

среда вокруг агента

локальный запуск

инфраструктурные  
ошибки

документация

skills

≈ 10 задач инфраструктурным командам

# План доклада

**01** Инфраструктура для агента

**02** Распространение в команде

**03** Лайфхаки: когда код становится дешёвым

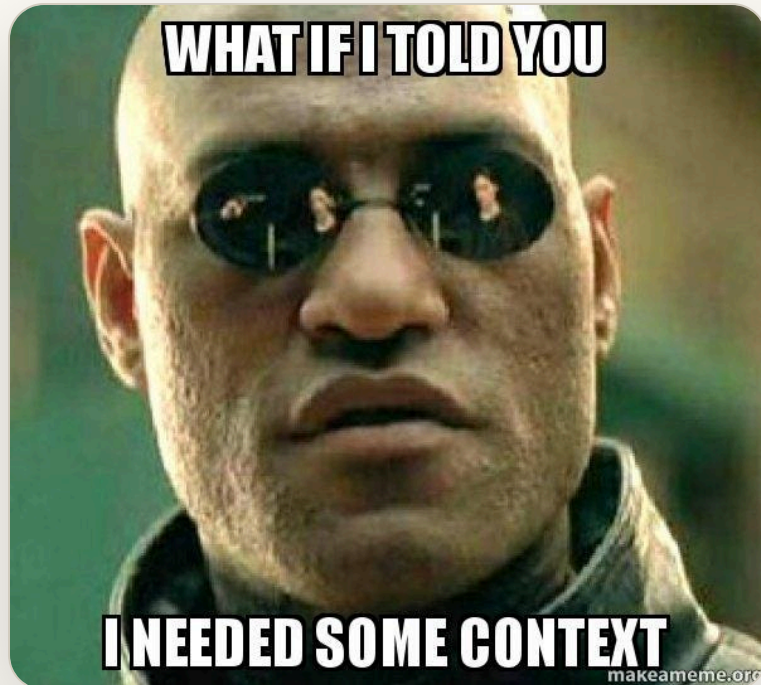
Часть 1 из 3

# Инфраструктура для агента

Как дать агенту контекст и инструменты для работы

01 / 03

01



# Всё начинается с контекста

- что это за сервис и как он устроен;
- какие решения уже приняты;
- как его запускать и проверять;
- где искать похожие PR, задачи и обсуждения.

Без контекста агент начинает с  
расспросов и угадывания.

# Контекст хранится рядом с кодом

```
my-service/  
├─ AGENTS.md    ← входная точка  
├─ ARCHITECTURE.md ← устройство проекта  
├─ CHANGELOG.md ← что менялось  
└─ docs/  
    ├─ plans/    ← планы решений  
    ├─ local-run.md ← инструкция  
    └─ pitfalls.md ← подводные камни
```

Все знания должны оседать в текстовом виде

# AGENTS.md: точка входа

## AGENTS.md

|                          |                                                                                                                                                         |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>## Required Route</b> | Use \$presentation-maintainer for every deck edit.<br>slides.md = order; pages/*.md = editable sources.<br>Show Pavel an exact proposal before editing. |
| <b>## Source Map</b>     | current-state.md → narrative and order<br>presentation-requirements.md → stable rules<br>todo.md → open feedback<br>sources.md → sources and claims     |
| <b>## Boundaries</b>     | Preserve work; never create a Git stash proactively.<br>Keep preview on loopback.                                                                       |

Примеры: [Codex](#) · [Temporal](#)

# Контекст за пределами репозитория

## Знания

**Вики, тикеты,  
внутренняя  
документация**

## CI

**Сборка, тесты,  
деплой**

## Мониторинг

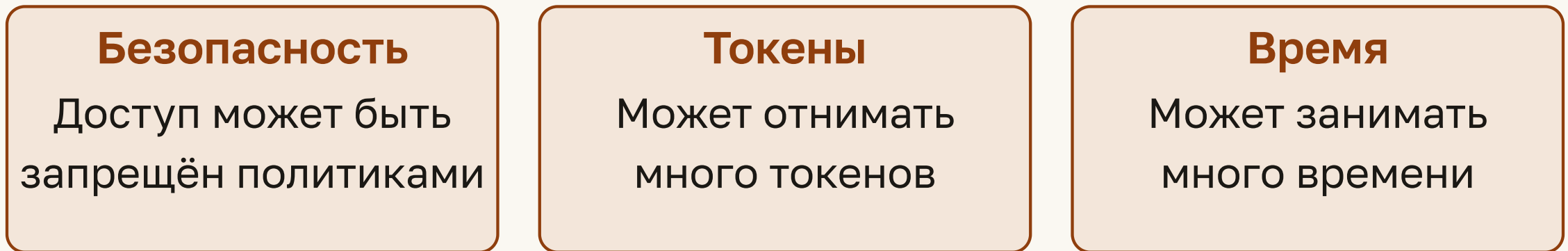
**Логи, алерты,  
метрики — как  
сервис живёт**

Если агент не может сам получить эти факты, человек передаёт их вручную и становится узким местом.

# Агент может делать всё через браузер



## Минусы



# MCP & Skills

В Яндексе

**100+**

МСП

**10000+**

Скиллов

# Что дают МСР и скиллы

**01**

## Доступ к знаниям

Поиск по коду, документации, вики и задачам дал агенту контекст для работы.

**02**

## Инфраструктура

Собрать сервис, выкатить его в testing, проверить статусы и разобраться с проблемами деплоя.

# Оформление рабочего сценария в скилл

Подготовка release notes

Вручную

**≈ 1,5 часа**

С агентом

**15–20 мин**

Co skill

**≈ 5 мин**

# Базовый набор скиллов

## **arc**

Ветки, коммиты и работа с изменениями

---

## **tracker**

Задачи, комментарии и статусы

---

## **wiki**

Чтение и обновление страниц вики

---

## **docs**

Чтение документации сервисов

---

## **intrasearch**

Поиск по внутренним источникам

---

## **infractl**

Описание и развёртывание инфраструктуры

---

# Готовые MCP и skills

| Инструмент                            | MCP                        | Skills                                     |
|---------------------------------------|----------------------------|--------------------------------------------|
| <a href="#"><u>Grafana</u></a>        | <a href="#"><u>MCP</u></a> | <a href="#"><u>grafana/skills</u></a>      |
| <a href="#"><u>GitLab</u></a>         | <a href="#"><u>MCP</u></a> | <a href="#"><u>для glab</u></a>            |
| <a href="#"><u>Octopus Deploy</u></a> | <a href="#"><u>MCP</u></a> | <a href="#"><u>в Claude Agent Step</u></a> |

Ещё MCP: [GitHub](#) · [TickTick](#)

Skills команды .NET: [github.com/dotnet/skills](#)

# Доступ к БД

## PostgreSQL

CLI · [psql](#)

Skill · [Neon Agent Skills](#)

MCP · [Postgres MCP Pro](#)

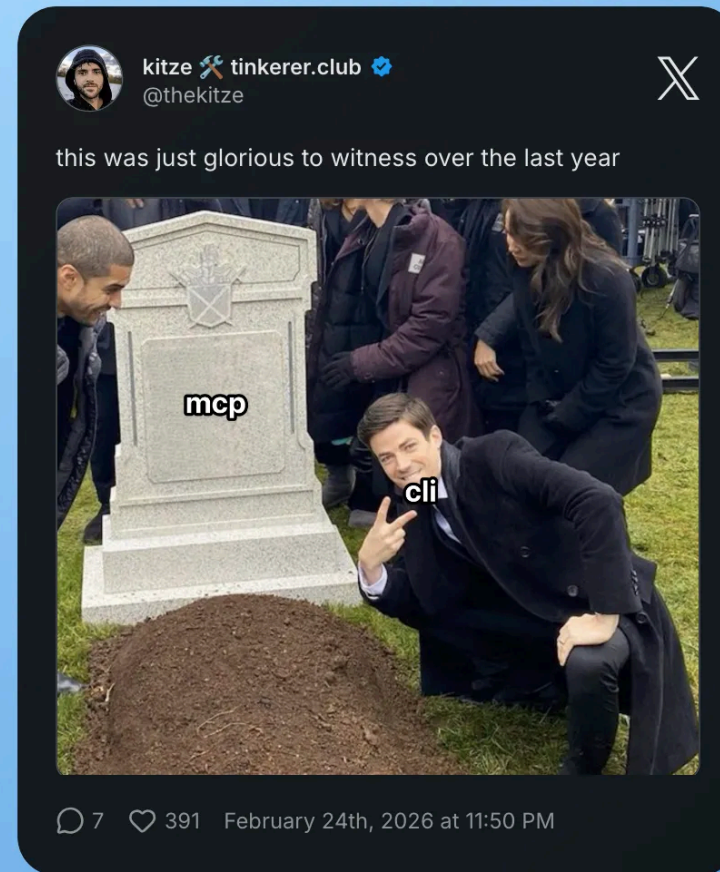
## SQL Server

CLI · [sqlcmd](#)

Skill · [EF Core](#)

MCP · [Microsoft SQL MCP](#)

# MCP IS DEAD?



# Нет. MCP продолжает развиваться

## Новая версия стандарта

Вышла 28 июля 2026 года.

## В разработке — Skills over MCP

Агент сможет получать через MCP и инструменты, и инструкции по их использованию.

# MCP МОЖНО ВЫЗЫВАТЬ ИЗ CLI

```
# посмотреть доступные инструменты
mcporter list github

# прочитать issue через GitHub MCP
mcporter call github.issue_read method=get \
  owner=openai repo=codex issue_number=123
```

mcporter · mcp2cli

# Skills + MCP = Love

Yandex Tracker stores issues in queues, projects, portfolios, and goals.

## How To Call MCP

```
... tracker_mcp --tool <ToolName>  
key:value
```

## Find my open issues

1. Filter: assignee = me(), status = open

2. Call **GetIssues**

3. Return key, summary, status

01 Описание  
с чем работает skill

02 Вызов MCP  
как вызвать инструмент

Сценарий  
03 фильтр → GetIssues →  
ответ

# Скилл для работы с сервисом - варианты

## Skill over API

Если прямой вызов разрешён

`SKILL.md` описывает API или CLI и пользовательский сценарий.

## Skill over MCP

Если нужны права и аудит

MCP управляет доступом, skill задаёт порядок действий.

Начинайте со Skill over API. Переходите к Skill over MCP, когда этого требуют безопасность или аудит.

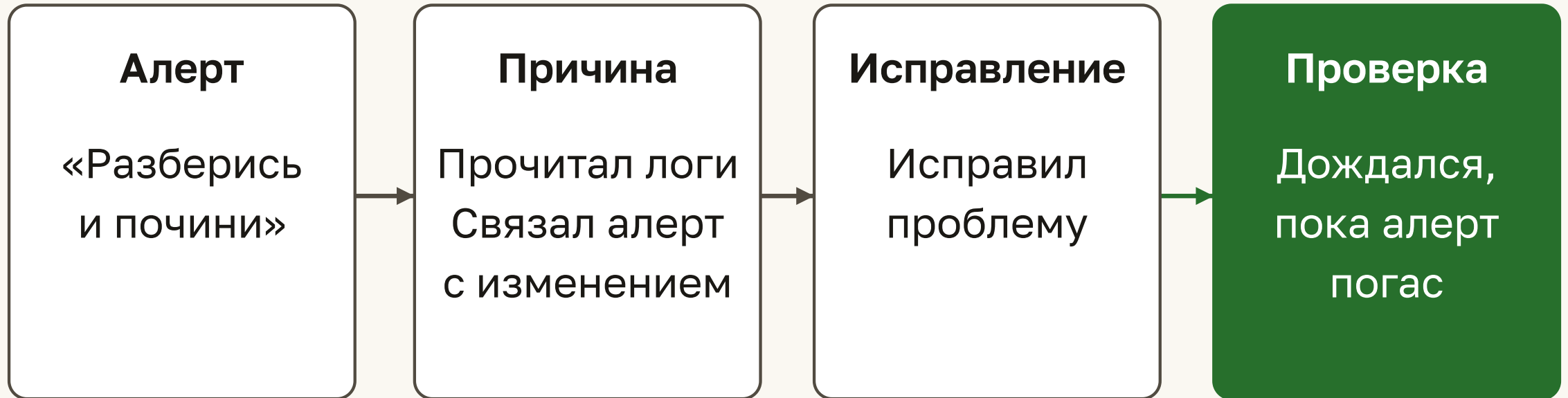
# Инфраструктура должна быть доступна агенту

| Вариант           | Вход для агента               |
|-------------------|-------------------------------|
| <u>Terraform</u>  | MCP + <u>skills HashiCorp</u> |
| <u>Crossplane</u> | Kubernetes API + skill        |
| <u>OpenTofu</u>   | CLI + skill                   |

В Яндексе — `infractl`

IaC помогает агенту понимать, как устроена инфра вокруг кода

# Идеальная картина



# Важность локальной проверки

Проверка через testing

деплой → ожидание → ошибка

≈ 15 минут

Локальная проверка

одна команда + тестовые данные

запуск → healthcheck →  
smoke-test

≈ 1 минута

---

Хорошая среда разработки = хорошая среда для агента.

# Настройка требует времени

**01**

Настроить

Команда запуска,  
локальные  
зависимости, БД и  
тестовые данные.

**02**

Описать

Docs, skill и  
однозначный  
критерий успешной  
проверки.

**03**

Поддерживать

Обновлять вместе с  
сервисом и сохранять  
рабочий smoke-test.

# Зачем проверять skills

## Без эвалов

Узнаём об ошибках  
от пользователей.

## С эвалами

Проверяем изменения  
на известных сценариях.

---

Eval  $\approx$  юнит-тест для skill.

# Уровни проверки skill

| Уровень               | Что проверяем                              |
|-----------------------|--------------------------------------------|
| <b>L0 · Структура</b> | Корректно ли оформлен skill?               |
| <b>L1 · Активация</b> | Выбирает ли агент его для нужного запроса? |
| <b>L2 · Сценарий</b>  | Решает ли агент задачу с его помощью?      |

# Пример eval

```
{
  "prompt": "Покажи основную информацию по
пулреквесту 999: заголовок, автора и
текущий статус.",
  "expectations": [
    "Ответ называет автора PR (login
alice)"
  ],
  "mock": {
    "dir": "scripts/evals"
  }
}
```

## Реальная задача

Агент читает страницу  
через mock.

## Наблюдаемый результат

LLM-as-a-Judge проверяет  
ответ агента.

# HTML-отчёт: сравнение версий

Пример: skill для создания дашбордов.

Skill Eval Report

8/26/2026, 11:03:25 AM

lightdarkauto

13/13 vs 11/13

QUALITY - A +2 PASS VS B

-461.0k -27.8%

SOLVER TOKENS - A - B

-\$0.025 -18.6% - est.

SOLVER COST - A - B

Usage & cost

all runs · 5.6M tokens · ~\$0.4657

| RUN                             | SOLVER                     | SOLVER DELTA                             | AVG SOLVER / COMPLETED    | COST / SUCCESSFUL          | JUDGE OVERHEAD             | TOTAL BUDGET             |
|---------------------------------|----------------------------|------------------------------------------|---------------------------|----------------------------|----------------------------|--------------------------|
| A<br>monium-dashboard-create    | 1.2M tokens<br>~\$0.1106   | -461.0k · -27.8%<br>-0.0252 USD · -18.6% | ~\$0.0085<br>13 completed | ~\$0.0085<br>13 successful | 829.0k tokens<br>~\$0.0628 | 2.0M tokens<br>~\$0.1734 |
| B<br>monium-dashboard-create    | 1.7M tokens<br>~\$0.1358   | reference                                | ~\$0.0104<br>13 completed | ~\$0.0123<br>11 successful | 813.0k tokens<br>~\$0.0593 | 2.5M tokens<br>~\$0.1951 |
| baseline<br>baseline · no skill | 376.6k tokens<br>~\$0.0421 | not compared                             | ~\$0.0032<br>13 completed | ~\$0.0210<br>2 successful  | 769.3k tokens<br>~\$0.0551 | 1.1M tokens<br>~\$0.0972 |

Solver delta is A - B; judge overhead is excluded. Changes are highlighted at |change| ≥ 5%. Cost never changes PASS/FAIL.

Pairwise deltas

3 comparisons · right side is reference

| PAIR                                                          | QUALITY                   | SOLVER TOKENS      | SOLVER COST                |
|---------------------------------------------------------------|---------------------------|--------------------|----------------------------|
| A - B<br>monium-dashboard-create - monium-dashboard-create    | +2 pass<br>13/13 vs 11/13 | -461.0k<br>-27.8%  | -\$0.025<br>-18.6% · est.  |
| A - baseline<br>monium-dashboard-create - baseline · no skill | +11 pass<br>13/13 vs 2/13 | +820.7k<br>+218.0% | +\$0.069<br>+163.1% · est. |
| B - baseline<br>monium-dashboard-create - baseline · no skill | +9 pass<br>11/13 vs 2/13  | +1.3M<br>+340.4%   | +\$0.094<br>+223.0% · est. |

Без skill

2 из 13

Предыдущая версия

11 из 13

Новая версия

13 из 13

Пример одного запуска на 13 задачах.

# Эвалы скиллов в Яндексе

В общем репозитории скиллов

## 01 · Автор

Добавить файлы

```
activations.json  
scenarios.json
```

## 02 · Мердж-чеки

При изменении  
skill

Автоматический  
прогон проверок на  
разных агентах.

## 03 · Результат

HTML-отчёт в PR

Ошибки и сравнение  
версий.

# Выводы

- Дайте агенту контекст проекта и доступ к инфраструктуре
- Закрепите повторяемую последовательность действий в skill
- Проверяйте изменения скилла на эвалах

Часть 2 из 3

# Распространение в команде

Как находить, подключать и поддерживать общий набор инструментов

02 / 03

02

# Единый каталог МСР-серверов

1. Найти сервер
2. Посмотреть инструменты
3. Открыть инструкцию подключения

## Карточка сервера

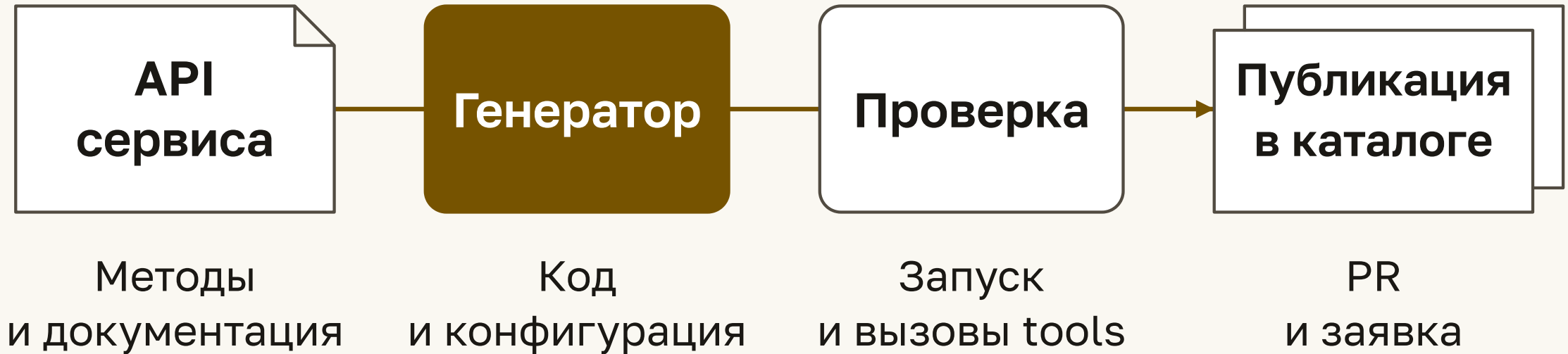
Описание      Документация

Исходники      Контакты команды

---

Команды публикуют интеграции в одном месте.

# Инфра для создания МСР в Яндексе



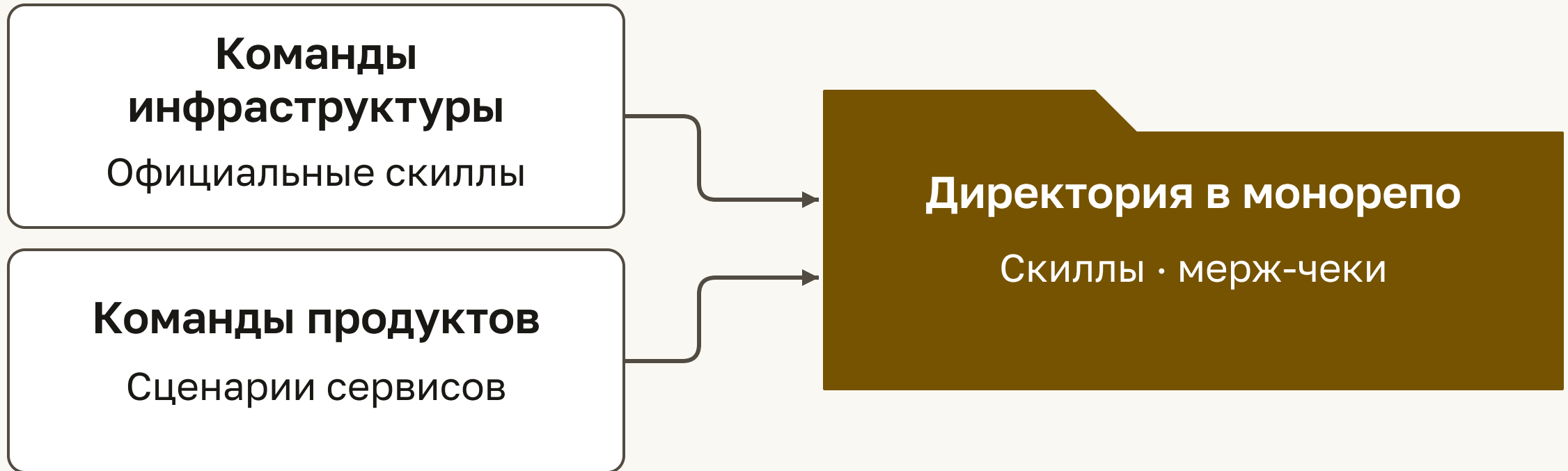
---

Типовую обвязку МСР собирает агент.

# Скиллов много. Как выбрать?

- Одну задачу могут решать несколько похожих скиллов.
- Непонятно, какой выбрать и как его проверяли.

# Скиллы команд — в общем репозитории



# Каталог помогает выбрать скилл

UI Skill Store

Поиск · авторы · рейтинги · кнопка фидбэка

Основа — директория скиллов в монорепо

## Эвалы

Что проверяли  
и с каким  
результатом.

## Использование

Насколько скилл  
востребован.

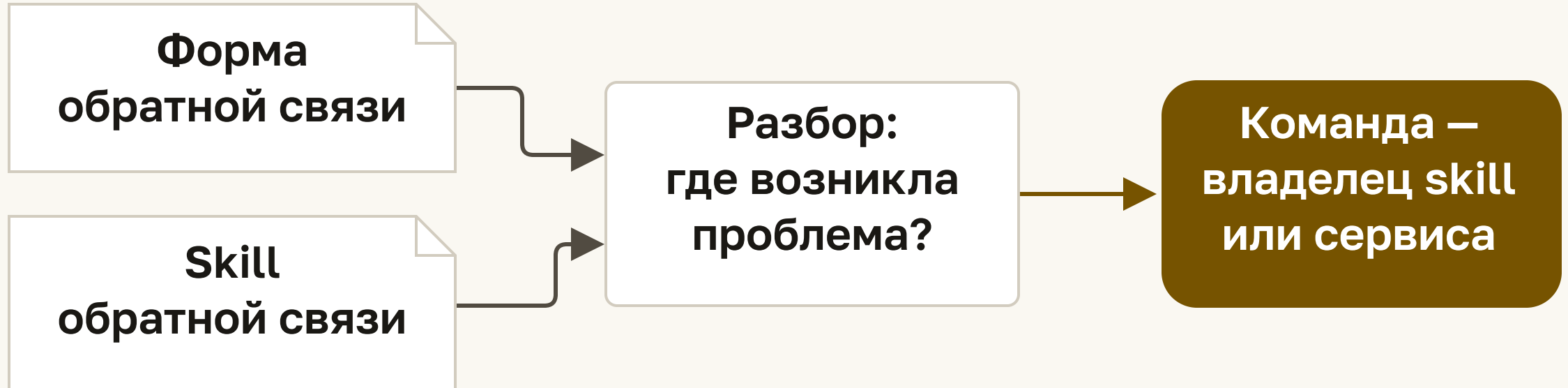
## Рейтинги

Как его оценивают  
пользователи.

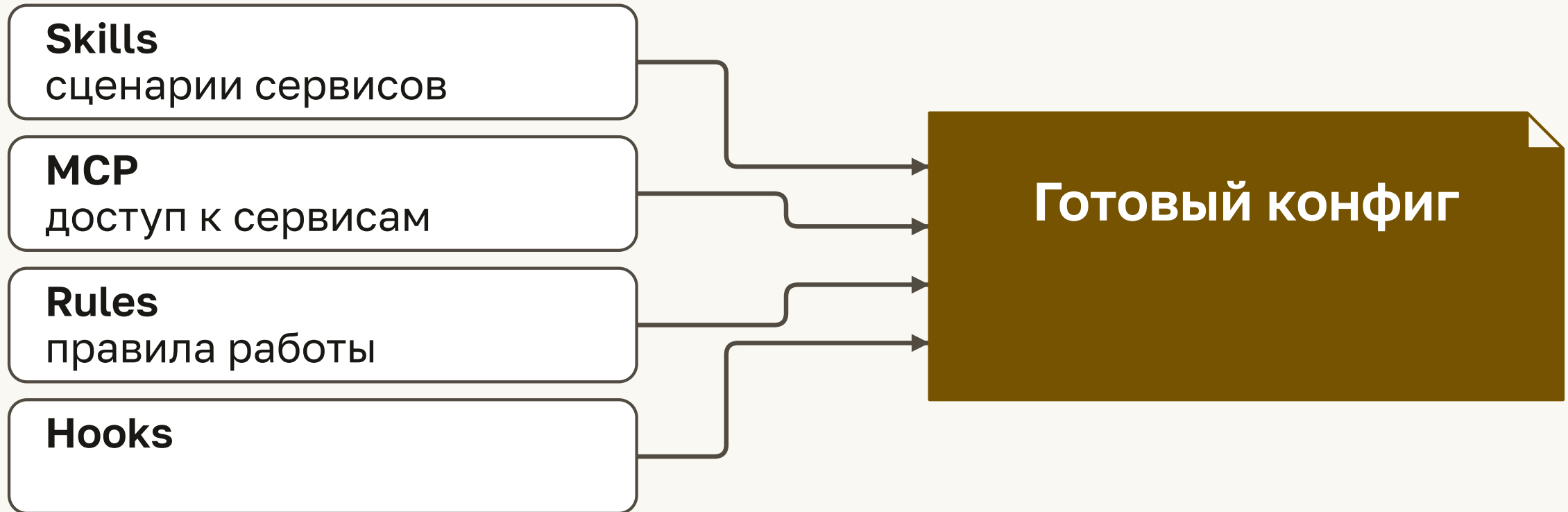
# Автоматические проверки скиллов в Pull Request

- Проверка безопасности
- Проверка структуры скилла
- Проверка зависимостей из пресетов
- Эвалы скиллов
- Поиск похожих скиллов — проверка на возможные дубли

# Процесс фидбэка на скиллы



# Как собрать harness для команды?



Нужен один способ подключить общий набор для команды.

# aisuite собирает окружение под проект



`aisuite.yaml`

```
skills:  
- "infra/**"
```

Скиллы для работы со всей инфраструктурой

# Способы подключения

## В проекте

```
my-service/  
├─ aisuite.yaml  
├─ AGENTS.md  
├─ src/  
└─ docs/
```

Конфиг в корне проекта

## Отдельный пресет

Конфиг хранится отдельно

Один пресет подключается к разным проектам

Пресет → Проект А  
Проект Б  
Проект В

# Открытые решения - плагины

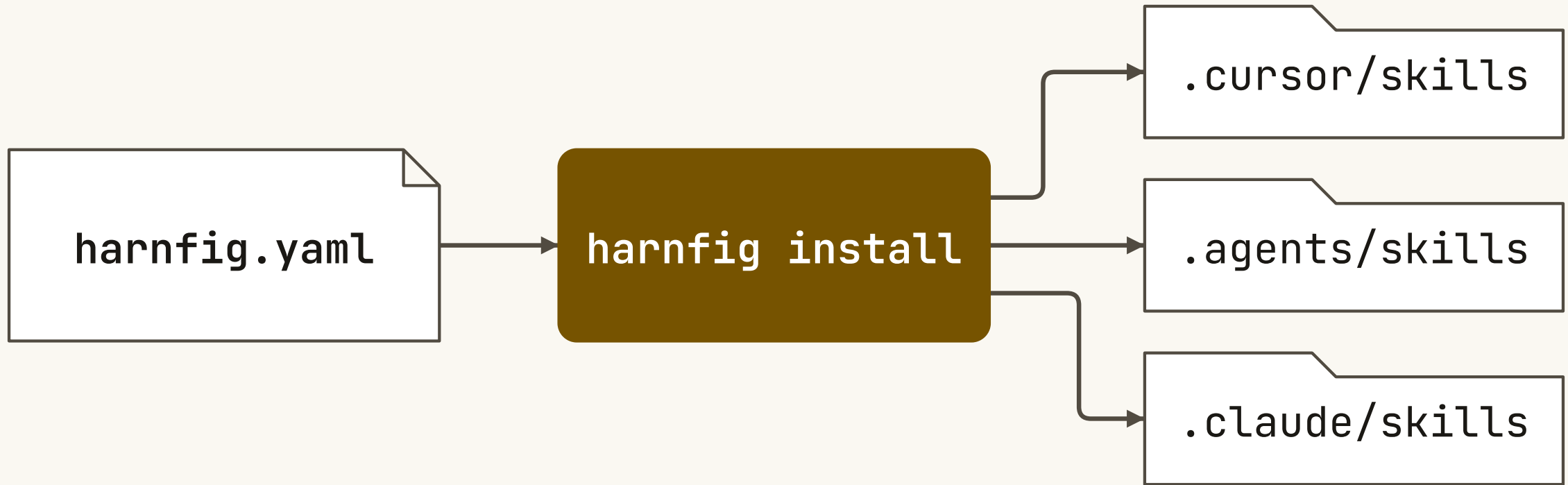
## Claude Code plugins

Skills, агенты, hooks и MCP в одном пакете

## Agent Plugins 1.0

Открытый стандарт: skills + MCP в одном пакете

# harnfig — open-source пример



[harnfig](#) · Собрал на .NET за вечер

Похожую утилиту можно быстро собрать под свою команду.

# Пример конфига

```
version: 1
scope: project
agents:
  - cursor
  - codex
  - claude
skills:
  - repo: mattpocock/skills
    path: skills/productivity/grill-me
    ref: main
```

**agents**

Куда поставить

**repo + path**

Откуда взять skill

**ref**

Ветка, тег или commit

# Выводы

- Сделайте общий каталог скиллов и MCP
- Опишите harness для агента в виде конфига
- Настройте цикл обратной связи для фидбэка по MCP и скиллам

Часть 3 из 3

# Лайфхаки: когда код становится дешёвым

03 / 03

03

# Код как новый ассемблер

## Фокус

Не держим весь код в голове. Проверяем контракт: входы, выходы, метрики, тесты.



## На практике

LLM может написать много кода. Инженер проектирует рамку, в которой этот код можно принять.



Важнее собирать слой выше: требования, архитектуру, тесты и обратную связь.



A line of Neo soldiers from the movie The Matrix Reloaded, wearing dark suits and sunglasses, standing in a row. The image is dark and moody, with the soldiers' faces partially obscured by shadows.

# Как бороться с нейрослопом?

# Понятные описания тикетов и PR

Тикет: что нужно решить

Проблема и ожидаемое поведение.  
Ограничения и варианты решения.  
Критерии приёмки.

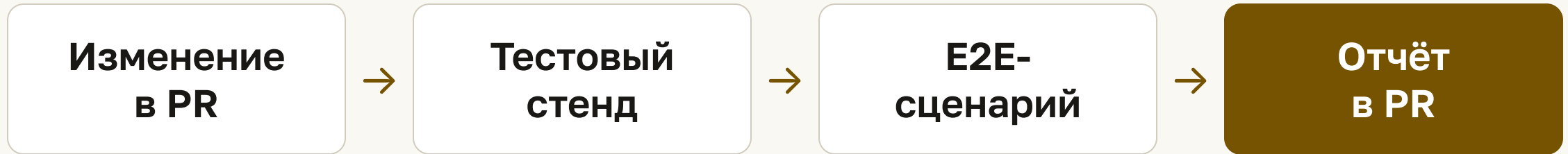
PR: что изменилось

Что сделано и почему.  
Как проверялось, с каким  
результатом.  
Риски и оставшиеся вопросы.

---

Шаблон можно закрепить в skill

# Е2Е-проверка — часть PR



## Результат проверки

- Проверен текущий коммит
- Видны ожидаемый и фактический результат
- При падении доступны trace и логи

# До реализации — ADR

ADR — запись архитектурного решения.

Для существенного изменения разработчик приносит короткое предложение:

- Что изменится
- Почему
- Какие варианты рассмотрели
- Какие последствия

Разработчик должен уметь всё объяснить.  
Команда проверяет.

# Сначала план, потом код

Уточнение плана

4–5

итераций до  
реализации

- агент собирает контекст и пишет план;
- инженер задаёт уточняющие вопросы;

Дешевле поправить план до изменений, чем переделывать

# Визуальный план проще понять

## Горы нейрослопа

Согласовать контракт фильтра.

Затем добавить параметр в API и фильтр в интерфейс.

Когда обе части готовы, проверить E2E-тестом, что фильтр меняет список.

## План в HTML



Визуализируй план в HTML: покажи этапы, зависимости и точки проверки.

# Визуализируйте, что происходит в системе

## Админка

состояние базы · безопасные операции

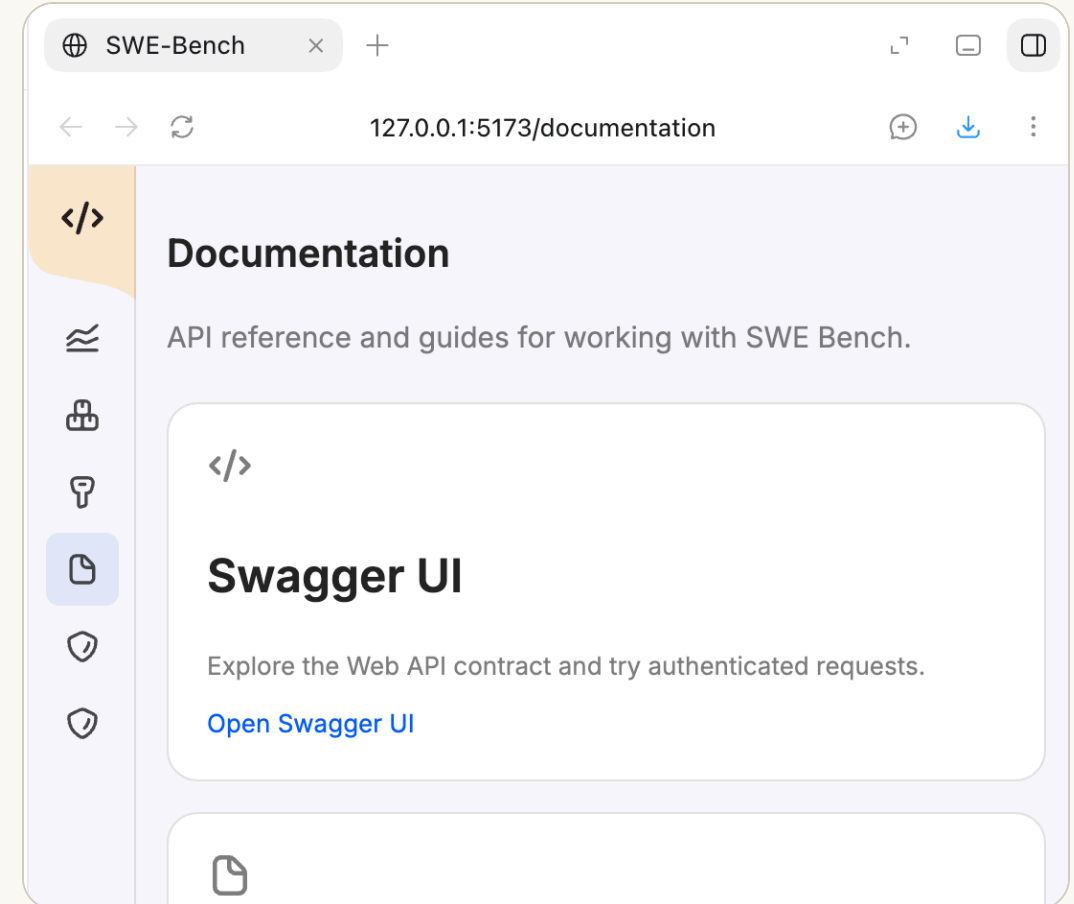
## Архитектура сервиса

компоненты · зависимости · потоки данных

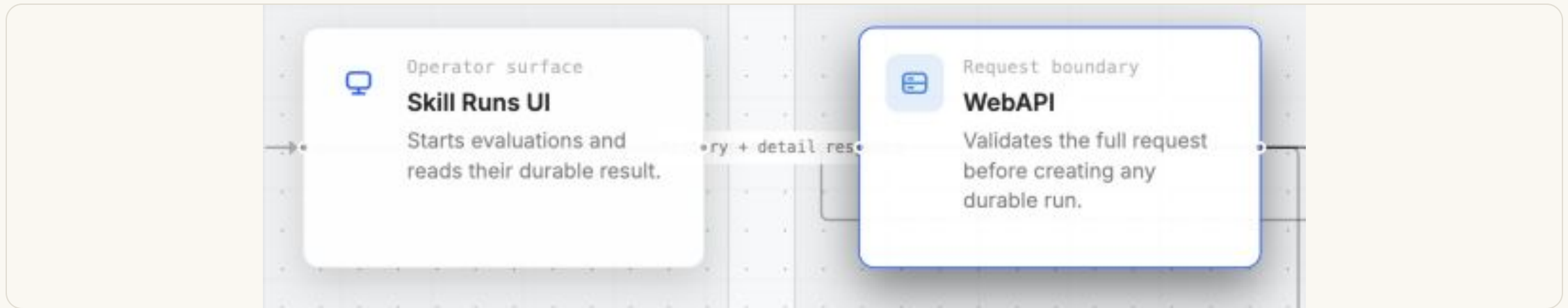
## Страница с полезными ссылками

Swagger UI · документация · инструкции

Агент может собрать такие страницы прямо внутри приложения.



# Попросите агента визуализировать архитектуру



archify — skill для интерактивных HTML-схем.

Компоненты, связи и путь запроса.

# Автоматизация обязательных шагов

## Проблема

Агент может пропустить обновление docs или форматирование кода.



## Решение

Обязательные шаги запускаем через хуки.



## Итог

- В конце ответа хук проверяет документацию по изменениям кода.
  - Если она устарела, возвращает агенту задачу обновить AGENTS.md, docs/ и другие документы.
- Проверка запускается автоматически.



# Задача инженера: создать среду для агента

80 % работы

Это настройка контура  
работы для агента



Главное — дать агенту доступ ко всему инженерному контуру:

- к внутренней документации и поиску, чтобы он сам разбирался в задачах
- деплою приложения и просмотру логов, чтобы он мог проверять гипотезы
- базе и миграциям, чтобы доводить изменения до прода
- общему контексту в репозитории: боли превращаются в docs, skills и hooks



OpenAI: Harness engineering

# SDLC агента — цикл с обратной связью



# Полезные материалы



[dotnet/skills](#)

---

Готовые skills  
для .NET



[OpenAI](#)  
[Harness engineering](#)

Как устроить среду  
для агента



[Claude Code](#)  
[Plugin evals](#)

Как проверить  
пользу плагина