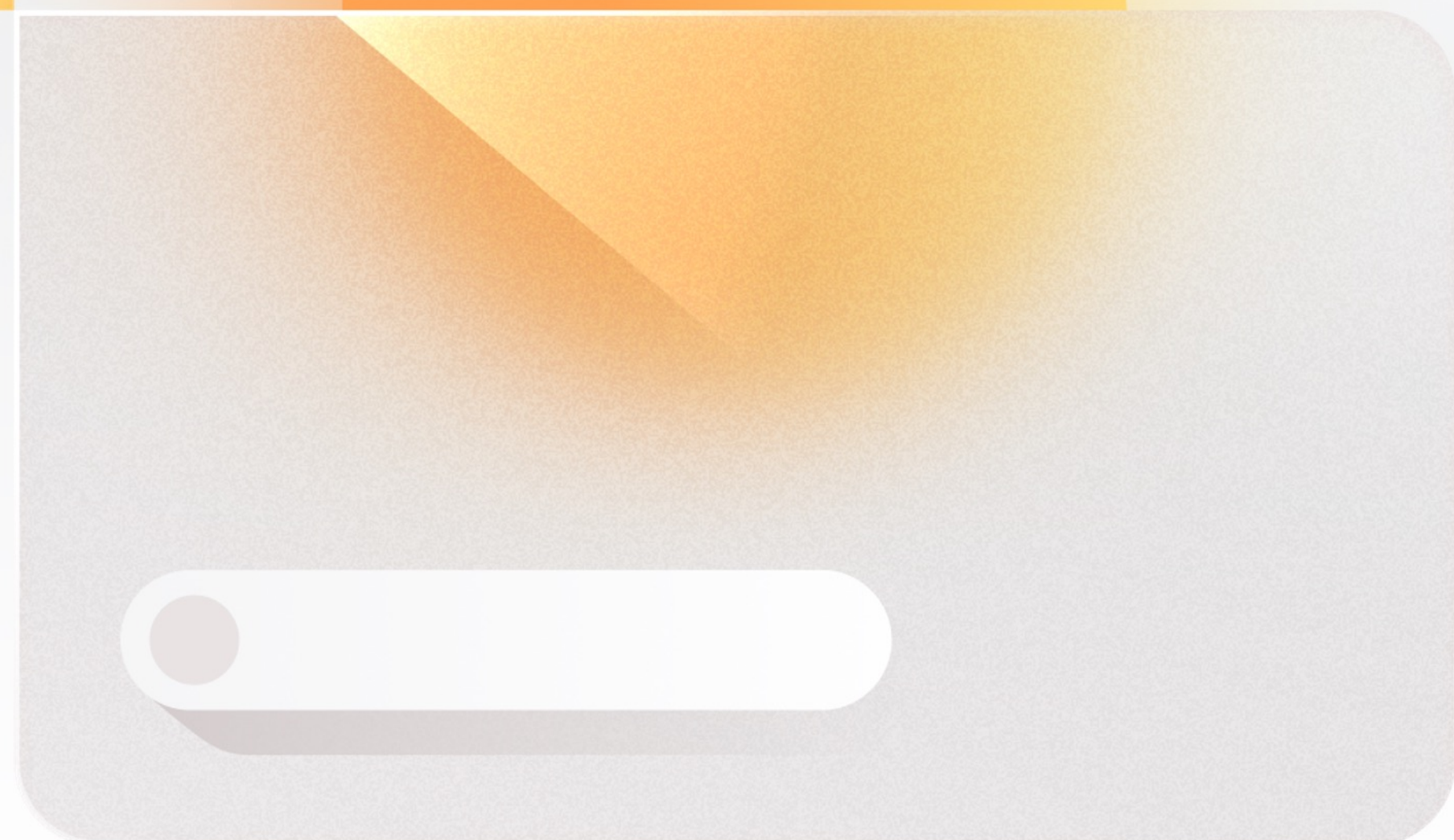
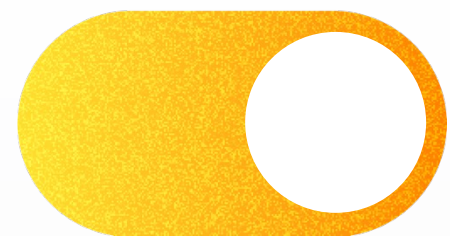


**Контракт — суперсила,  
а не бюрократия**



# Дмитрий Кузнецов



архитектор

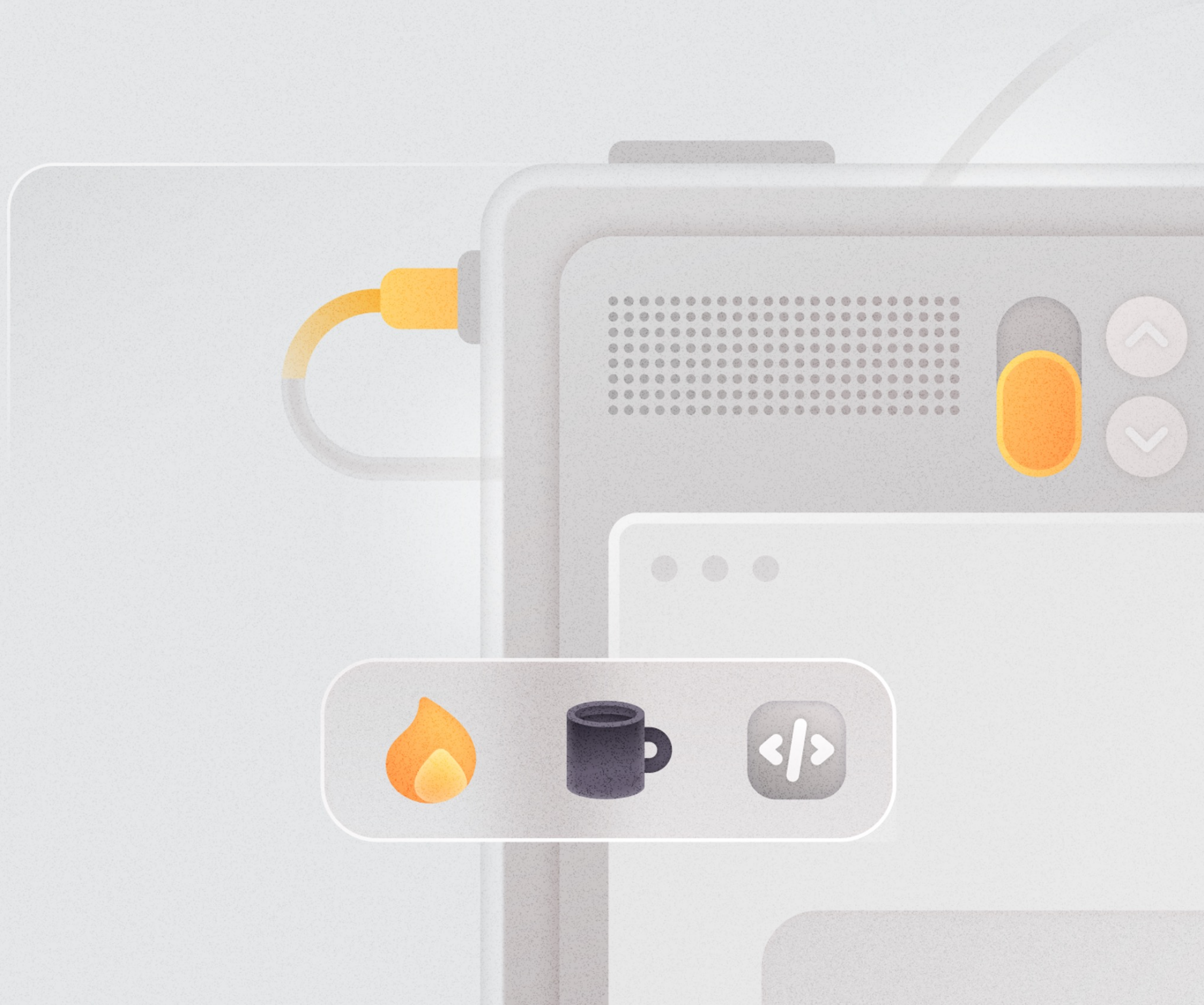
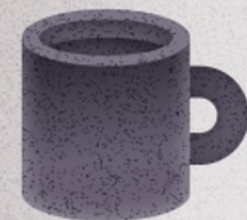
Email [d.o.kuznetsov@tbank.ru](mailto:d.o.kuznetsov@tbank.ru)

Telegram [@diokuz](https://t.me/@diokuz)



01

**Что такое  
контракт?**



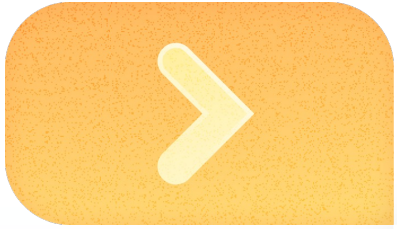


Договор

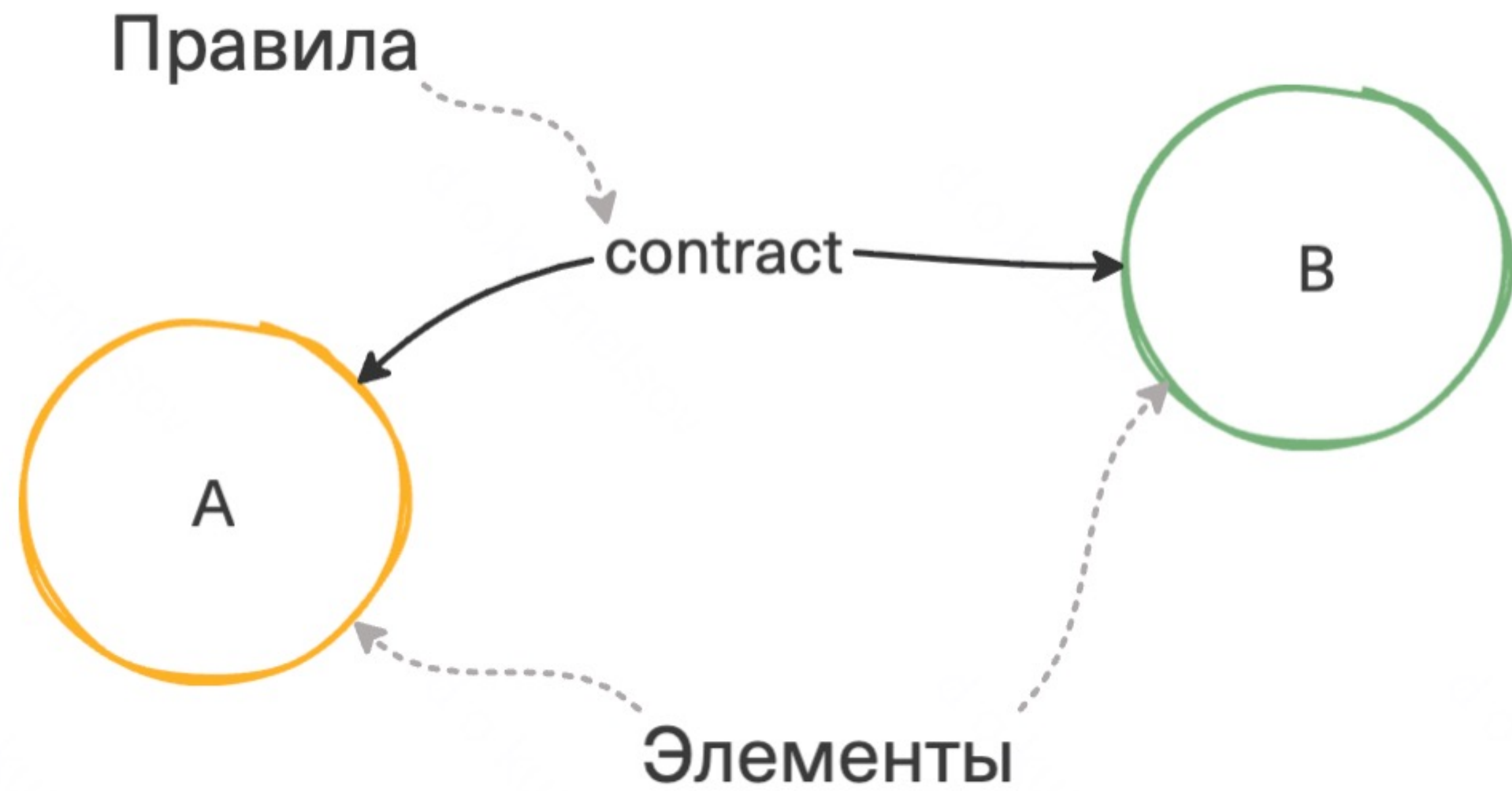
Договор между людьми о том,  
как будут взаимодействовать  
программы

# Что будем называть контрактом?

- ✓ Всем понятен
- ✓ Формален
- ✓ Между разными командами



# Система



# JSON Schema



Описывает любые объекты



Колоссальная экосистема

Document1 x

```
# JSON Schema для интерфейса User
type: object
description: Пользователь системы
required:
  - id
  - name
  - email
properties:
  id:
    type: string
    description: Уникальный идентификатор пользователя
  name:
    type: string
    description: Полное имя пользователя
    minLength: 1
  email:
    type: string
    format: email
    description: Электронная почта пользователя (уникальная)
additionalProperties: false
```

# Open API

- ✓ Использует JSON Schema
- ✓ Для http-взаимодействий
- ✓ Postman, Swagger

```
Document1 x
paths:
  /users/{id}:
    get:
      responses:
        '200':
          description: A user object
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/User'
```

# Async API



Использует JSON Schema



Для брокеров: Kafka, RabbitMQ

Document1 x

```
asyncapi: 2.6.0
info:
  title: User Event API
  version: 1.0.0
  description: API для событий, связанных с пользователями

servers:
  production:
    url: broker.example.com:9092
    protocol: kafka
    description: Kafka брокер в продакшене

channels:
  user.created:
    subscribe:
      message:
        $ref: '#/components/messages/UserCreatedMessage'

components:
  messages:
    UserCreatedMessage:
      name: UserCreated
      summary: Событие о создании пользователя
      payload:
        $ref: '#/components/schemas/User'
```

# TypeScript API

✓ Стандарт для фронтенда

✓ Объекты, методы, классы

Document1 x

```
/**
 * Представляет пользователя в системе.
 */
interface User {
  /**
   * Уникальный идентификатор пользователя.
   * Обычно в формате UUID или строковом ID.
   */
  id: string;
  /**
   * Полное имя пользователя.
   */
  name: string;
  /**
   * Электронная почта пользователя.
   * Используется для входа и связи.
   */
  email: string;
}
```

# Другие

Также контракты описываются:

- GraphQL, Protobuf
- Java интерфейсы, C header files, etc.
- Avro, WSDL, XML Schema,
- Naming rules, formats, regex...

# Очевидные бонусы

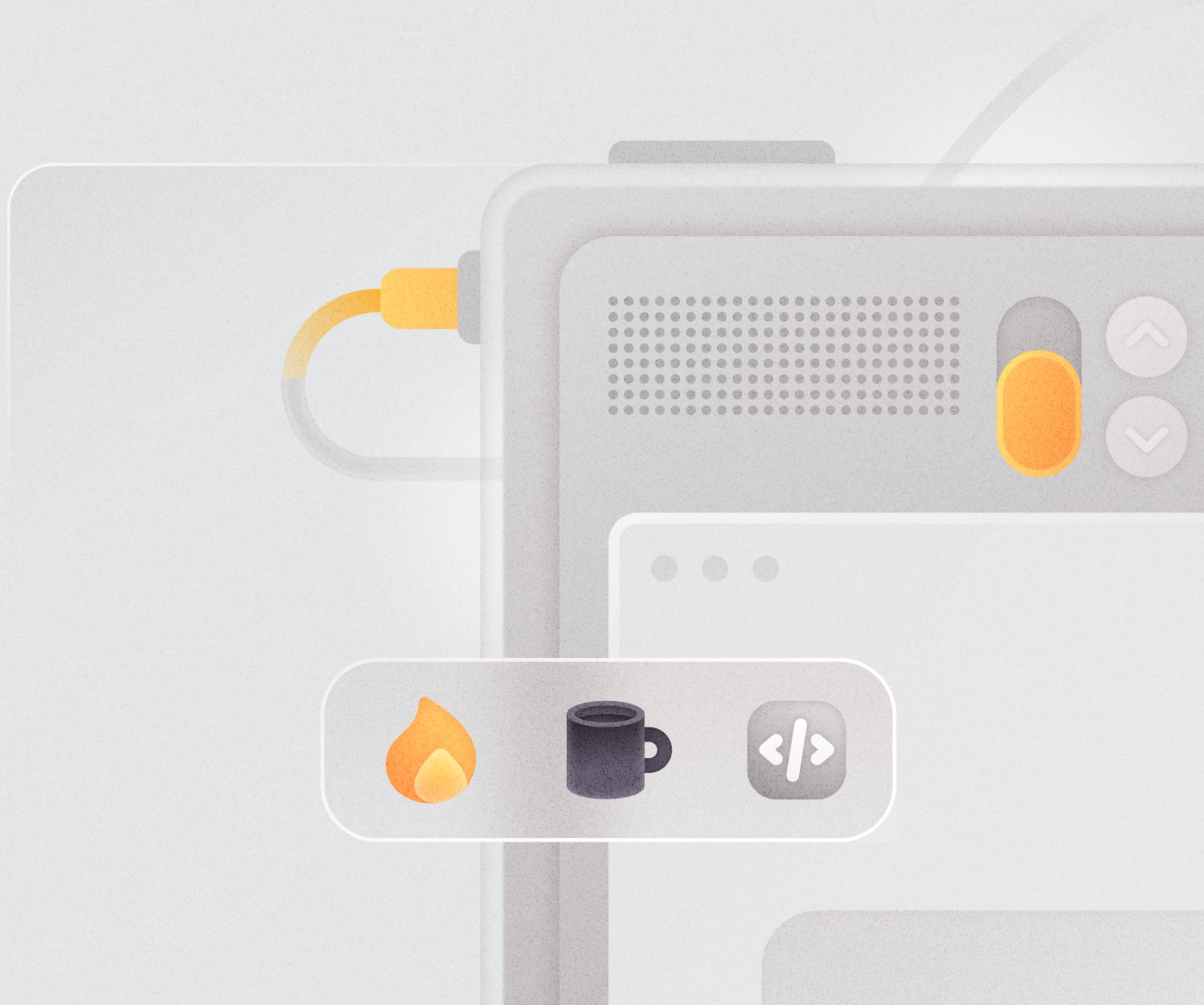
## Что даёт контракт?

- Типизацию
- Подсказки в IDE
- Документацию
- Синтетические моки
- Кодогенерацию

Здорово, а зачем?

02

Кейсы

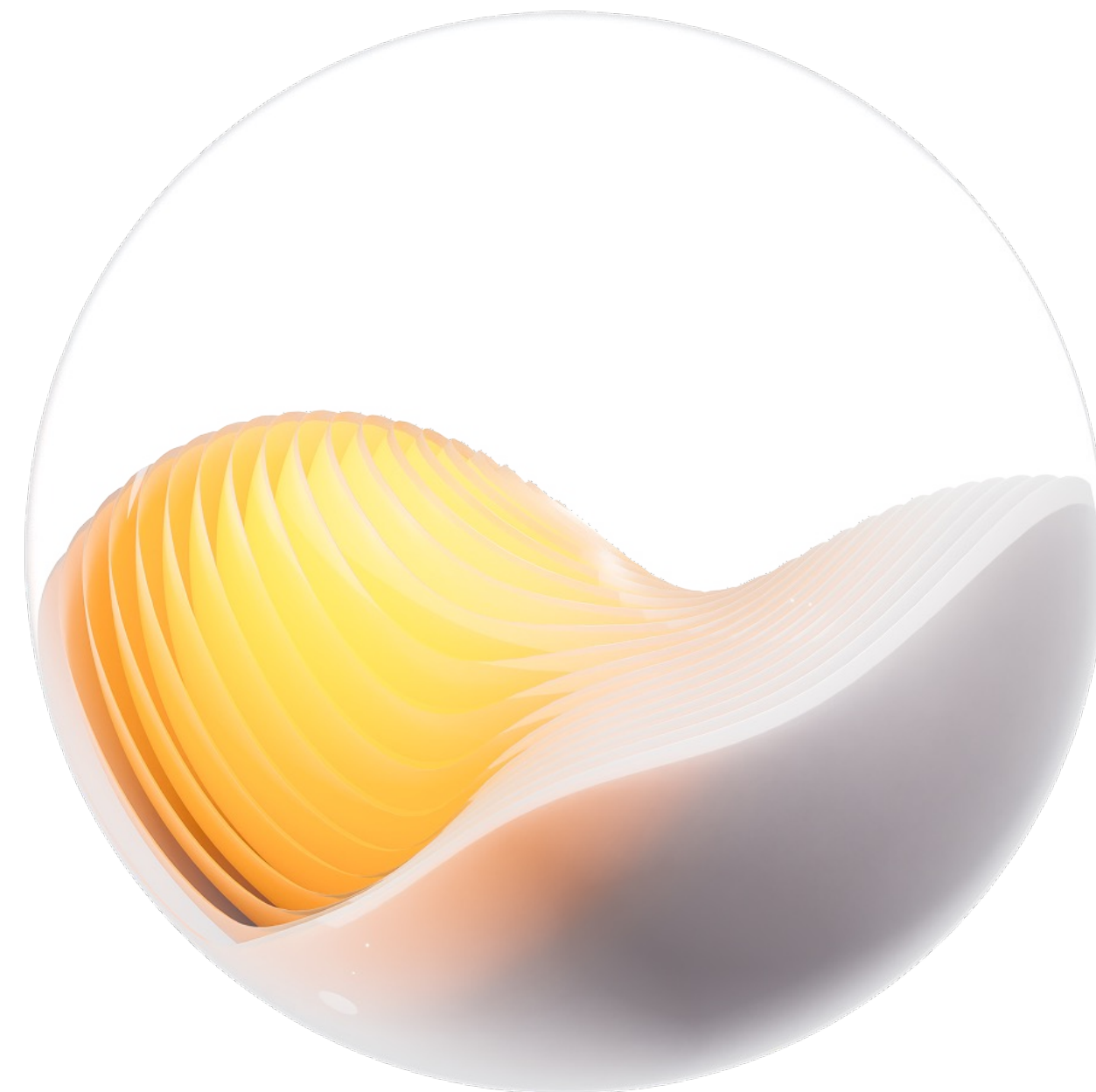


# Кейсы

- 01** Качество коммуникаций между командами
- 02** Скорость отладки
- 03** Тяжелые нестабильные тесты
- 04** Оценка тестового покрытия
- 05** Порог входа в архитектуру проекта
- 06** Совместимость с другим окружением
- 07** Риски при оценке сроков
- 08** Устаревание контракта

# Команды не понимают друг друга

- Перетягивание одеяла, холивары
- Глухой телефон
- Скрытые баги, риски



# Заявочная форма

## Закажите сим-карту

**Безлимит на всё — 490 ₽/мес**



ГБ и минуты на номера любых операторов по России



Перенесите номер до 30 апреля, чтобы сохранить цену тарифа

Фамилия, имя, отчество\*

+20%

Контактный телефон\*

+20%

Хочу бесплатно перенести номер



И сохранить цену тарифа

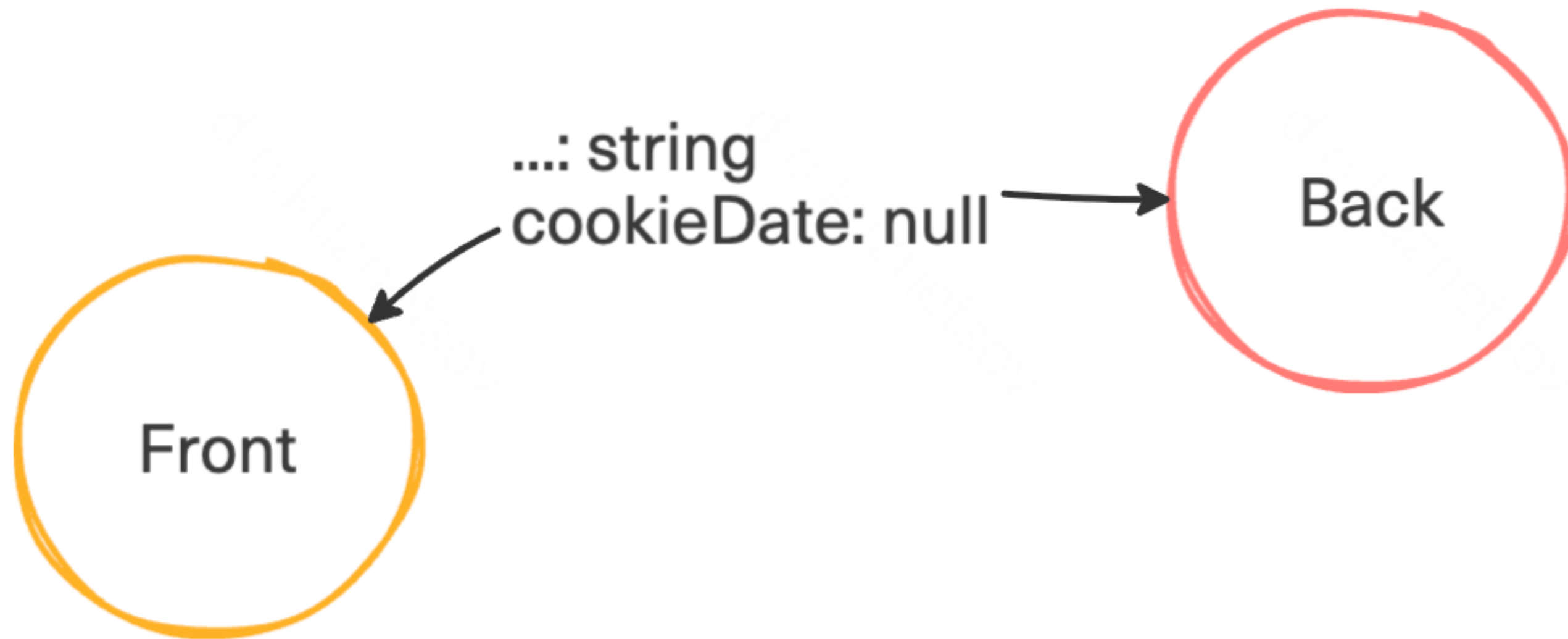
Уже заполнено

**20%**



+60% за контактную информацию

# Бэк не понял фронт



# Оптимизация холивара

- ✓ Закончили созвон
- ✓ Фронты ушли работать
- ✓ Бэки ушли прикручивать кодогенерацию

≡ JSON Schema ×

```
title: Record<string, any>
description: Объект со значениями любого типа.
type: object
additionalProperties: {}

# Мы, команда бэкенда,
# обязуемся принимать поля любого типа
```

# Куда заводит баг?

- Раньше работало
- Теперь не работает
- Баг в приложение?
- Нет!

The image shows two overlapping windows. The top window is a code editor titled 'TypeScript' containing the following code:

```
{  
  "type": "close",  
  "value": null  
}
```

Below the code is a comment: `// Поле value обязательно!`

The bottom window is a browser window titled 'NaChat' with the address bar showing `www-test.tbank...`. The browser console displays an error:

```
https://www-test.tbank.ru/mybank/nachat/  
@coretech-frontend/webview-bridge:  
1.130.1  
no window...postMessage  
Query Params:  
bridge (через веб сдк)  
theme (тема light / dark)  
mock_sin (включает моки на уровне...
```

A red 'X' icon is visible in the top-left corner of the browser window, indicating an error or crash.



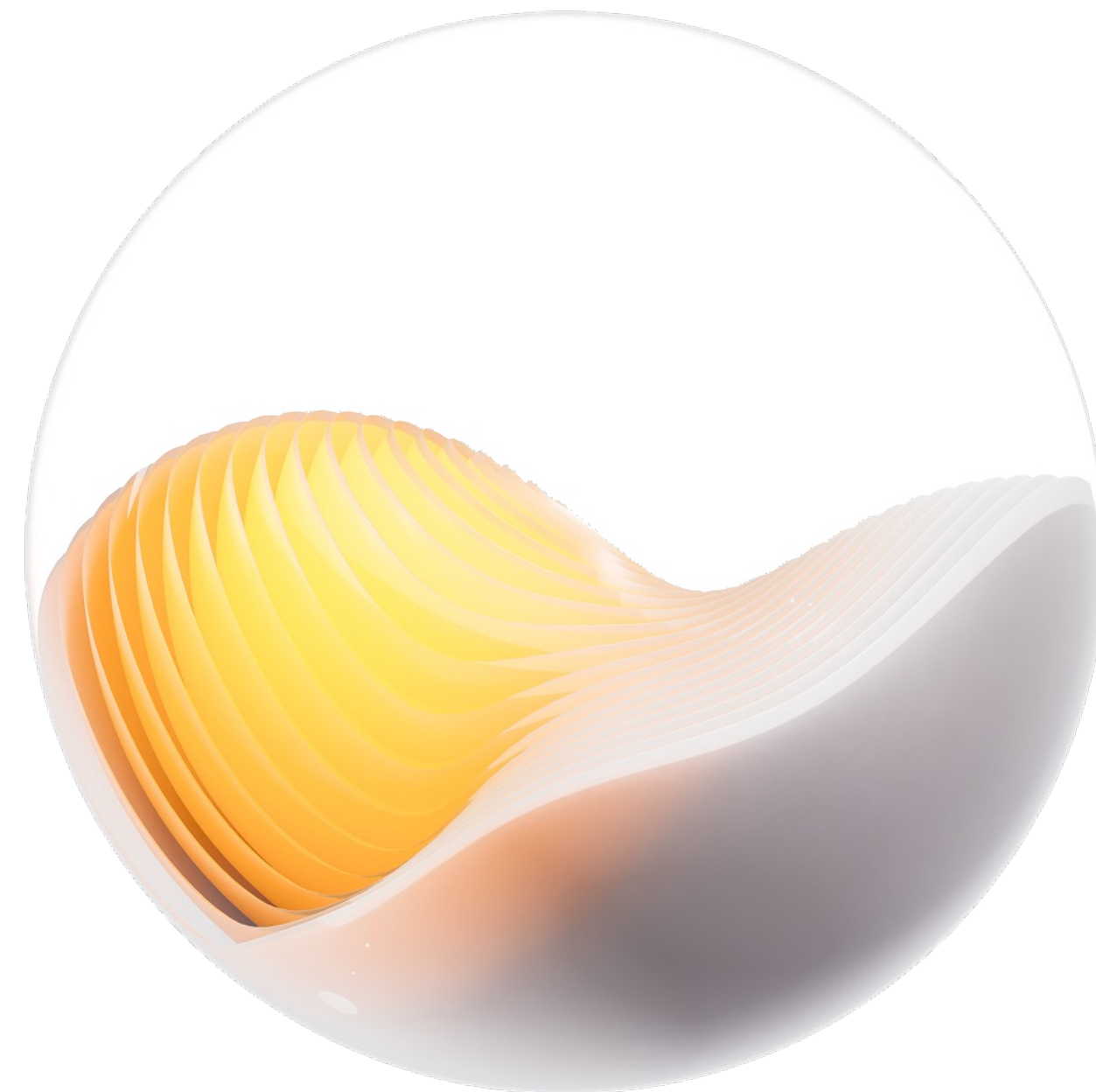
# Контракт это обязательство



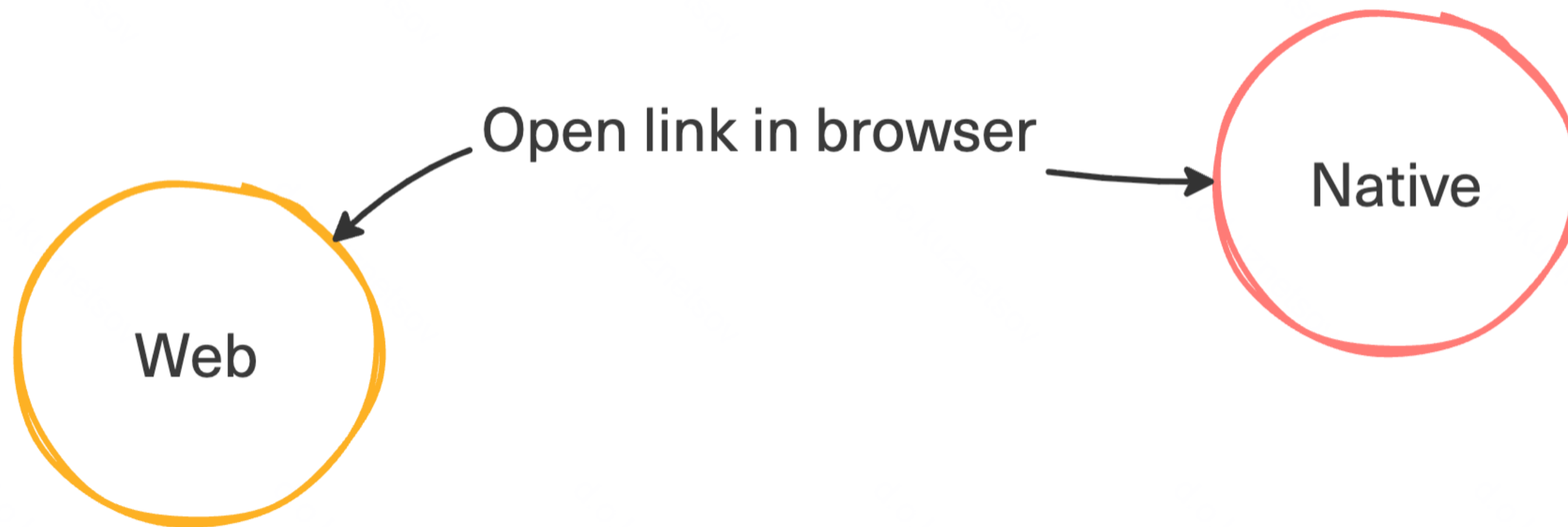
- ✓ Регламентирует границы зон ответственности между командами
- ✓ Отменяет холивары и созвоны
- ✓ Отвечает на вопрос — кому и что делать
- ✓ Цементирует понятия и ускоряет коммуникации

# Утомительный дебаг

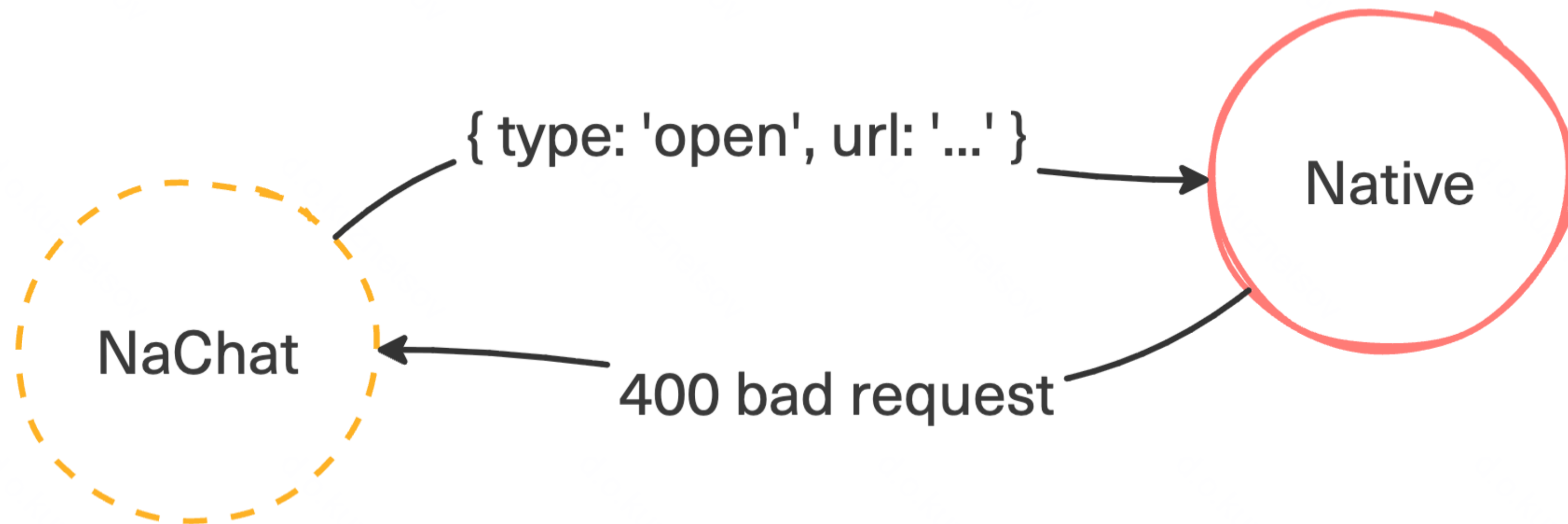
- Много кода, реп, языков
- Никто не разбирается во всём
- Дорогое тестовое окружение



# Ссылка не открывается

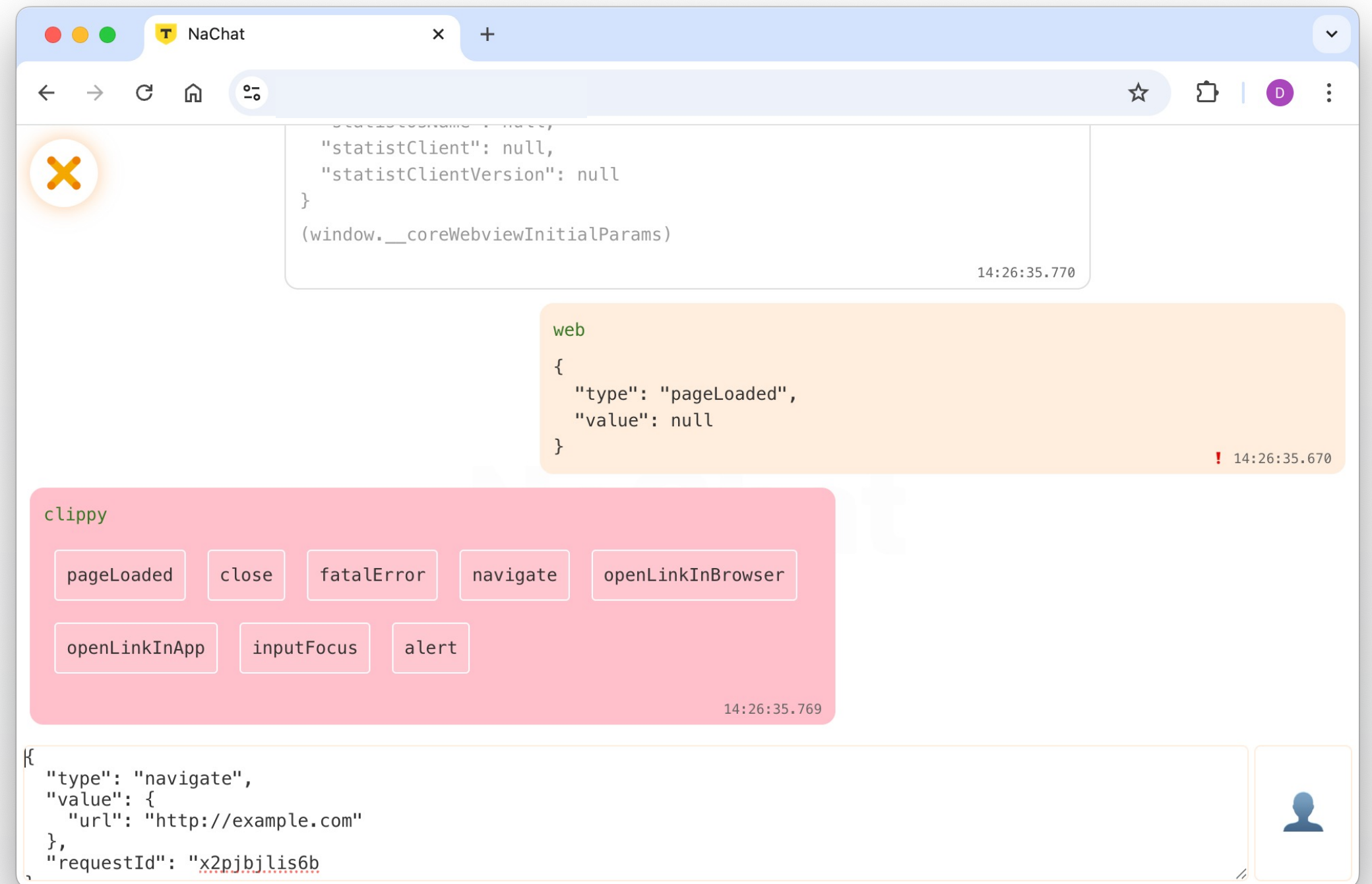


# Смотрим на взаимодействие!



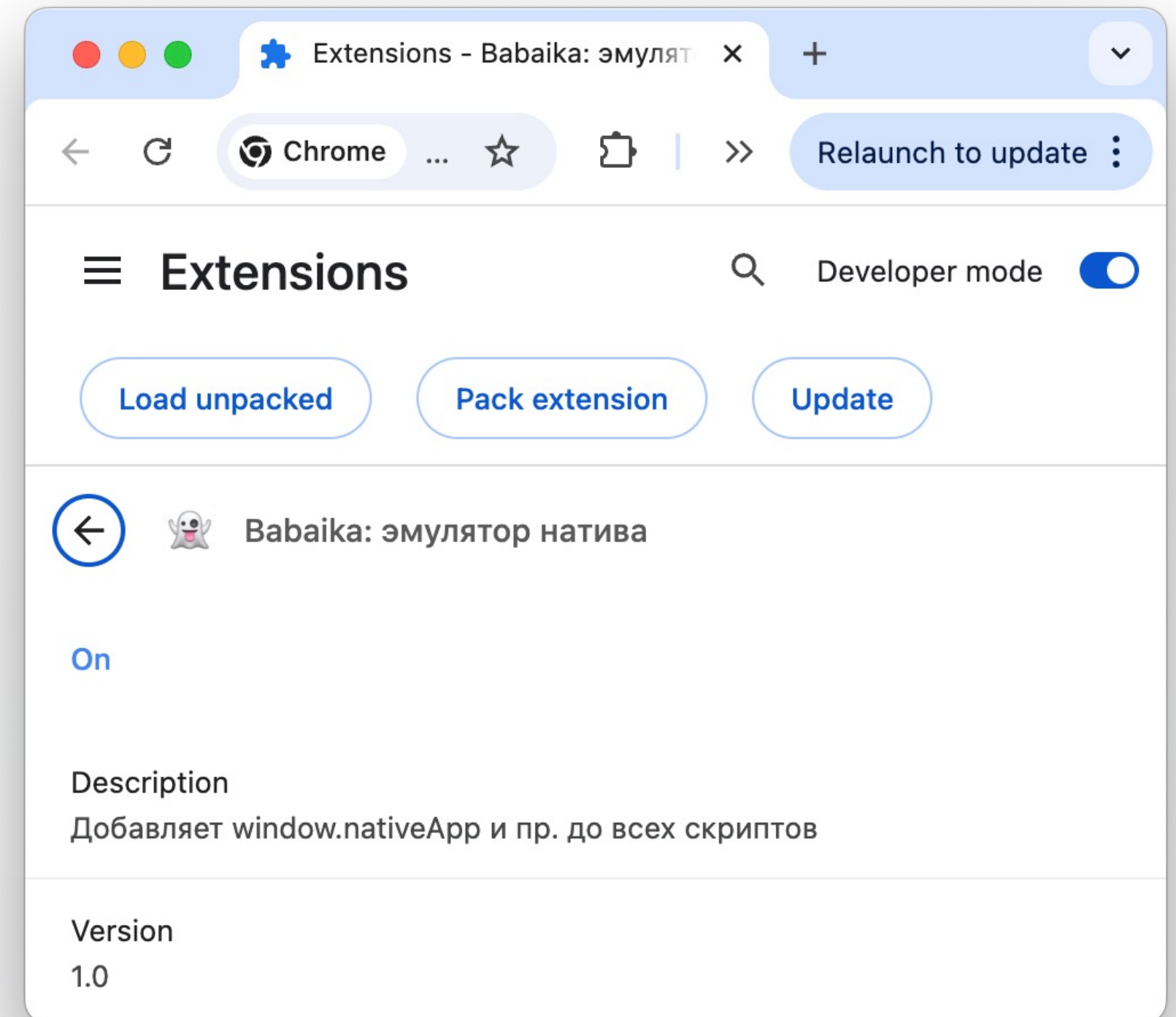
# В NaChat воспроизводится?

- ✓ Мок веба → дебажим без веба
- ✓ Строго по контракту
- ✓ Видим всё, что проходит границу



# Мок приложения

- ✓ Без Test Flight, Xcode и Android Studio
- ✓ Соблюдение контракта, наблюдаемость
- ✓ Можем тестировать в сторибуке



# Мок микрофронтенда

- ✓ Соблюдение контракта, наблюдаемость
- ✓ Не нужно хранить и актуализировать обфусцированный код

The screenshot shows the component inspector for a component. It displays the following information:

- Renders**: 1
- Локальный счетчик**: 0
- Запускался ли useEffect?**:
- test**: test-block 1.1 project
- Props**: {"foo": "bar"}
- State**: false, **Баланс**: 100500 P
- Router**: /mybank/foo/bar, **navigate** ← MIF\_ROUTER
- Manager**: setWebViewMFready() ← MICRO\_DI\_WEBVIEW\_MANAGER
- Bridge**: {"type": "pageLoaded", **send()** ← MIF\_WEBVIEW\_BRIDGE
- Warning**: Не могу работать без MICRO\_DI\_NOTIFIER

# Мок хостового приложения

- ✓ Соблюдение контракта, наблюдаемость
- ✓ Можно тестировать в сторибуке

The screenshot shows a web browser window with the following elements:

- Browser Tab:** header / \_\_stories\_\_ / block - x
- Address Bar:** localhost:3030/?pa...
- Page Title:** Тестируем жизненный цикл Header
- Text:** В этом сторисе мы можем проверить, что Header корректно демонтируется, а потом снова монтируется. Это важно для сценариев, когда di-контейнер блока кешируется, а потом вновь используется.
- Terminal:** A terminal window showing the lifecycle of the component:

```
packages-header-header__dark--iejWe @boxy/test-host  
10:48:15.801 on("example")  
→ 10:48:15.801 mounted  
→ 10:48:16.803 ready  
→ 10:48:16.803 full
```
- Controls:** Two toggle switches (one with a moon icon, one with a hand icon) and a 'full' label.
- Text:** Это блок Header.
- Footer:** \_\_stories\_\_/block/Lifecycle

# Диагностические инструменты

- Postman / Insomnia — изоляция сервиса
- WireMock — мок http-сервиса
- Chrome DevTools, MSW — сетевая изоляция
- Docker, Telepresence... — изоляция контейнера
- Testcontainers, H2... — изоляция БД



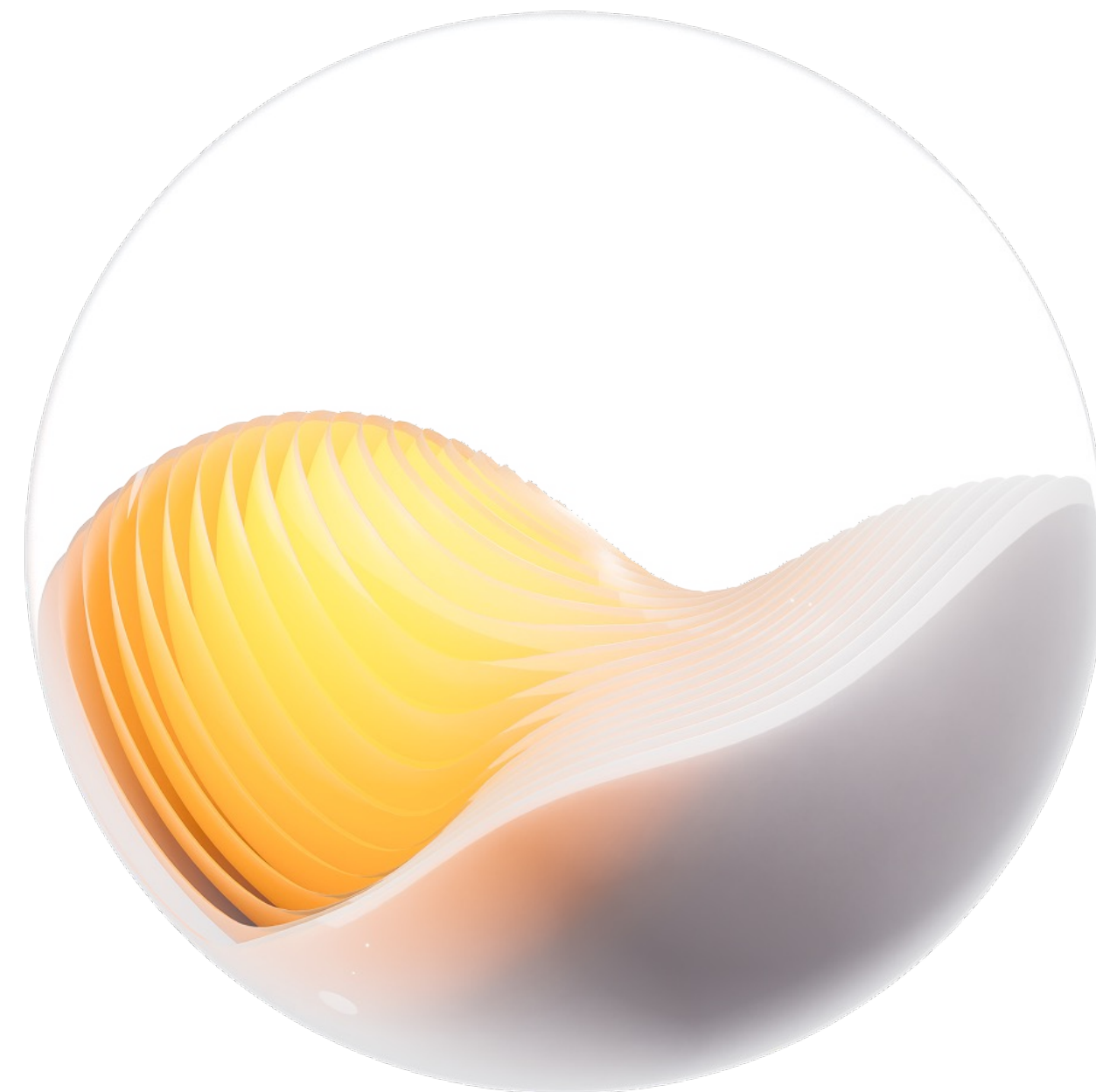
# Контракт — детектор нарушений



- ✓ Показывает, что проходит через (архитектурную) границу
- ✓ Позволяет (при помощи инструментов) эмулировать окружение,  
и не вникать в код

# Тяжелые тесты

- Долго проходят
- Нестабильные
- Сложно разбирать

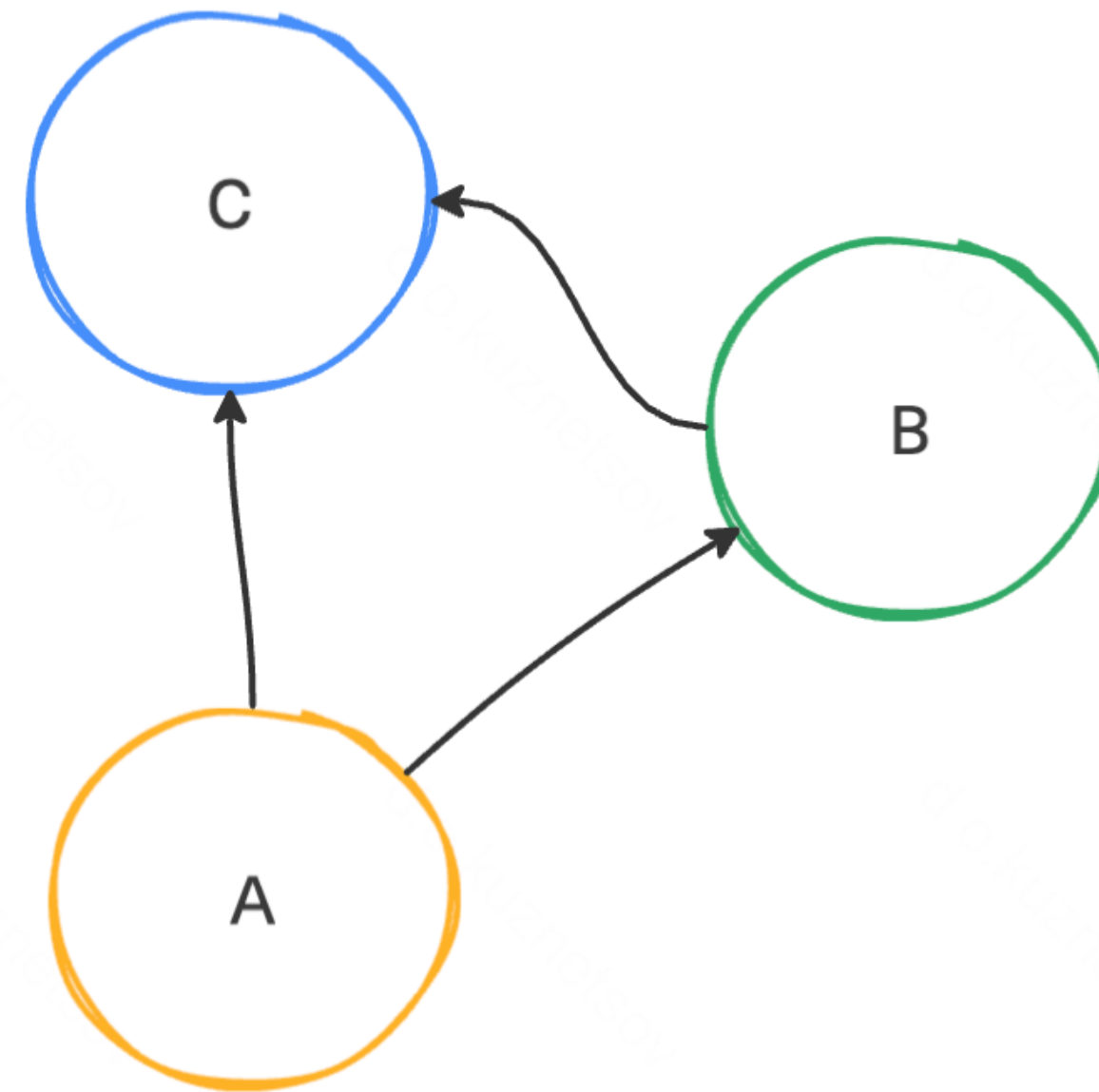


# Возможность изоляции

Монолит

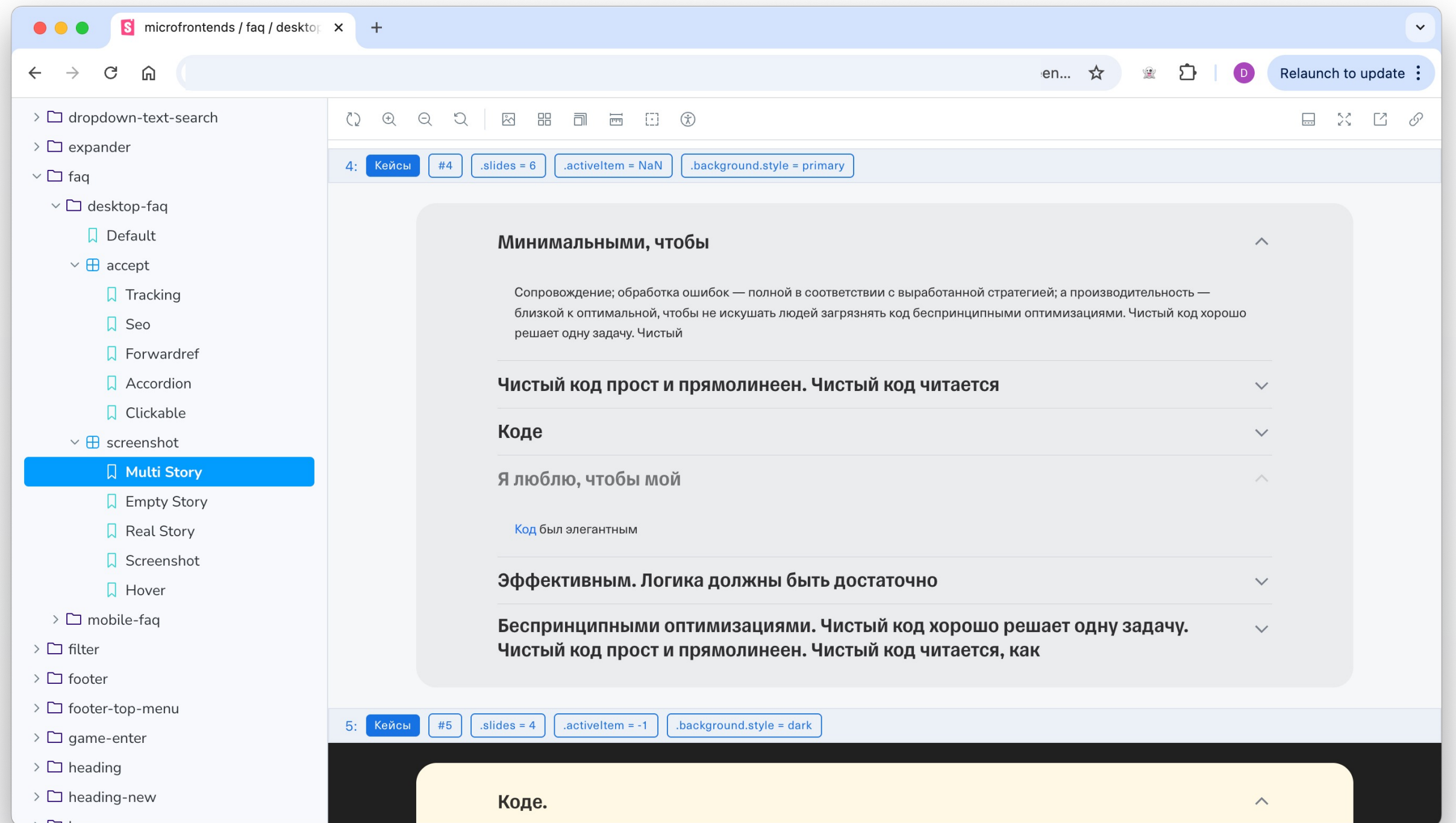


Система



# Тестирование визуального компонента

- ✓ Генерируем кейсы по контракту
- ✓ Знаем все краевые кейсы
- ✓ Pair-wise





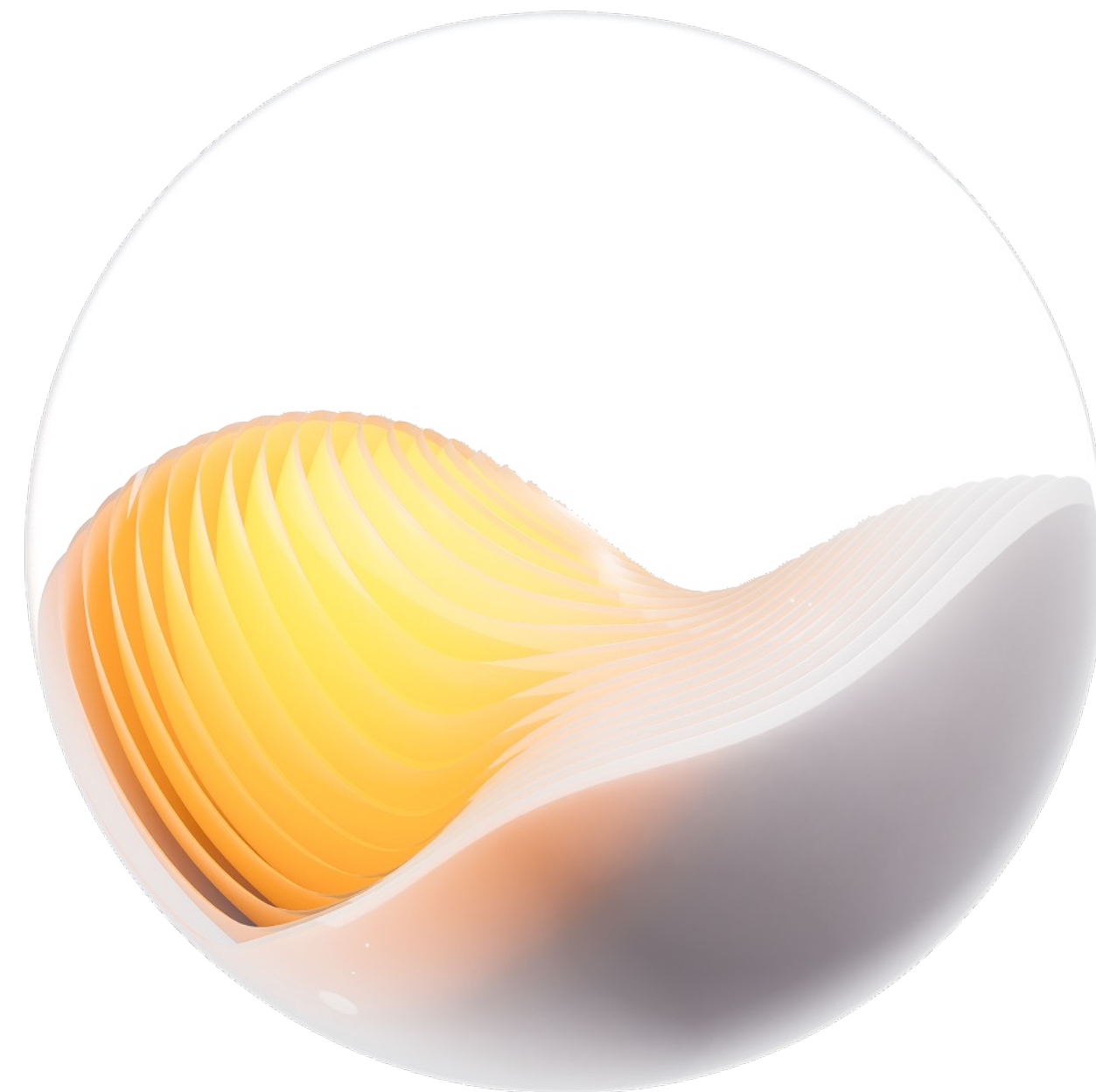
# Контракт — ключ к пирамиде



- ✓ Контракт это указатель на архитектурную границу
- ✓ Нет контракта — нет изоляции
- ✓ Без изоляции невозможна пирамида

# Оценка покрытия

— «Вы, конечно, молодцы, но учитывать ваши юниты, в тестовом покрытии, я не буду» ©



# Пирамиду сложно применять

## Много причин не применять пирамиду:

- ➔ Сложно понять связь юнитов с покрытием
- ➔ Проще написать пару системных тестов
- ➔ Нет требований к отдельным частям системы
- ➔ Нет доверия мокам

# ЖИВЫЕ МОКИ

- ✓ Тестируем бэк
- ✓ Вырезаем поля
- ✓ Видим полноту моков

```
TS get-city-names.test.ts x
> TS get-city-names.test.ts > test('Проверка контракта с passportApi API - метод getCityNames - неуспешный ответ: не заполнено поле prefix') са... >
57 test('Проверка контракта с passportApi API - метод getCityNames - неуспешный ответ:
58 passportApi,
59 auth,
60 storybook,
61 testApiBlock,
62 stubit,
63 }) => {
64   await auth.fastAuth();
65   await storybook.goToStory(getStoryUrl());
66
67   await passportApi.getCityNames.startWaitForResponse();
68
69   await testApiBlock.clickOnButton();
70
71   const responseJson = await passportApi.getCityNames.getResponse();
72
73   await test.step('Проверяем, что реальный ответ соответствует актуальному контракт
74     const strippedJson = motherApiResponseErrorSchema.parse(responseJson);
75
76     /**
77      * Сохраняем реальный ответ, отфильтрованный по схеме, в файл-стаб
78      * (если выставлен env.UPDATE_STUBS=1)
79      */
80     await stubit(
81       strippedJson,
82       'mocks/get-city-names/stubs/real-get-city-names-error-no-prefix.json'
83     );
84   });
85 });
```

# Живые моки



Тестируем бэк



Вырезаем поля



Видим полноту моков

```
TS get-city-names.test.ts x
> TS get-city-names.test.ts > test('Проверка контракта с passportApi API - метод getCityNames - неуспешный ответ: не заполнено поле prefix') са... >
57 test('Проверка контракта с passportApi API - метод getCityNames - неуспешный ответ:
58 passportApi,
59 auth,
60 storybook,
61 testApiBlock,
62 stubit,
63 }) => {
64   await auth.fastAuth();
65   await storybook.goToStory(getStoryUrl());
66
67   await passportApi.getCityNames.startWaitForResponse();
68
69   await testApiBlock.clickOnButton();
70
71   const responseJson = await passportApi.getCityNames.getResponse();
72
73   await test.step('Проверяем, что реальный ответ соответствует актуальному контракту
74     const strippedJson = motherApiResponseErrorSchema.parse(responseJson);
75
76   /**
77    * Сохраняем реальный ответ, отфильтрованный по схеме, в файл-стаб
78    * (если выставлен env.UPDATE_STUBS=1)
79    */
80   await stubit(
81     strippedJson,
82     'mocks/get-city-names/stubs/real-get-city-names-error-no-prefix.json'
83   );
84 };
85 );
```

# ЖИВЫЕ МОКИ

- ✓ Тестируем бэк
- ✓ Вырезаем поля
- ✓ Видим полноту моков

```
TS get-city-names.test.ts x
> TS get-city-names.test.ts > test('Проверка контракта с passportApi API - метод getCityNames - неуспешный ответ: не заполнено поле prefix') са... >
57 test('Проверка контракта с passportApi API - метод getCityNames - неуспешный ответ:
58 passportApi,
59 auth,
60 storybook,
61 testApiBlock,
62 stubit,
63 }) => {
64   await auth.fastAuth();
65   await storybook.goToStory(getStoryUrl());
66
67   await passportApi.getCityNames.startWaitForResponse();
68
69   await testApiBlock.clickOnButton();
70
71   const responseJson = await passportApi.getCityNames.getResponse();
72
73   await test.step('Проверяем, что реальный ответ соответствует актуальному контракт
74     const strippedJson = motherApiResponseErrorSchema.parse(responseJson);
75
76     /**
77      * Сохраняем реальный ответ, отфильтрованный по схеме, в файл-стаб
78      * (если выставлен env.UPDATE_STUBS=1)
79      */
80     await stubit(
81       strippedJson,
82       'mocks/get-city-names/stubs/real-get-city-names-error-no-prefix.json'
83     );
84   });
85 }
```

# Полнота моков

```
src > prod > community > api-clients > passport-api > __api__ > real-request > get-popular-names > TS get-popular-names.test.ts > ...  
27 const TEST_FIRST_NAME_LETTERS = 'алекс';  
28  
29 > test.describe('Проверка контракта с passportApi API – метод getPopularNames – успешные ответы', () => {  
67   });  
68  
69 > test('Проверка контракта с passportApi API – метод getPopularNames – неуспешный ответ: не заполнено поле type', async ({  
99   });  
100  
101 > test('Проверка контракта с passportApi API – метод getPopularNames – неуспешный ответ: не заполнено поле name', async ({  
131   });  
132
```

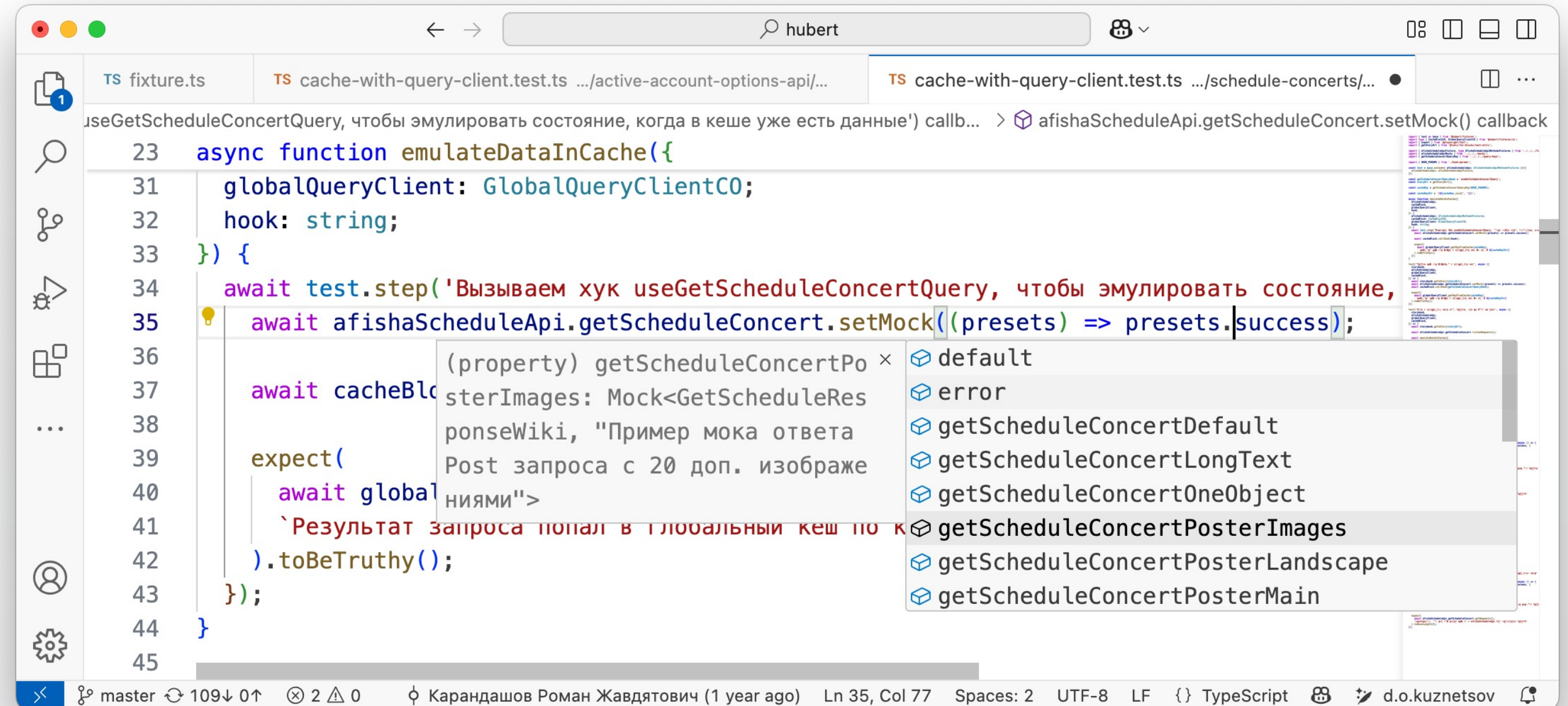
master 109↓ 0↑ 0 0 0  
Прилепин Виктор Анатольевич (1 year ago) Ln 131, Col 4 Spaces: 2 UTF-8 LF {} TypeScript d.o.kuznetsov

# Используем живые моки

✓ Моки реальны и актуальны

✓ Протестили бэк

✓ Видно пробелы в покрытии



```
TS fixture.ts TS cache-with-query-client.test.ts .../active-account-options-api/... TS cache-with-query-client.test.ts .../schedule-concerts/...
useGetScheduleConcertQuery, чтобы эмулировать состояние, когда в кеше уже есть данные') callb... > afishaScheduleApi.getScheduleConcert.setMock() callback
23 async function emulateDataInCache({
31   globalQueryClient: GlobalQueryClientC0;
32   hook: string;
33 }) {
34   await test.step('Вызываем хук useGetScheduleConcertQuery, чтобы эмулировать состояние,
35   await afishaScheduleApi.getScheduleConcert.setMock((presets) => presets.success);
36
37   await cacheBusterImages: Mock<GetScheduleRes
38   expect(
39     await global
40     `Результат запроса попал в глобальный кеш по к
41   ).toBeTruthy();
42   });
43 }
44 }
45
```

(property) getScheduleConcertPo ×

- default
- error
- getScheduleConcertDefault
- getScheduleConcertLongText
- getScheduleConcertOneObject
- getScheduleConcertPosterImages
- getScheduleConcertPosterLandscape
- getScheduleConcertPosterMain

master 109↓ 0↑ 2 ⚠ 0 Карандашов Роман Жавдятович (1 year ago) Ln 35, Col 77 Spaces: 2 UTF-8 LF {} TypeScript d.o.kuznetsov

# Неосознанное

- ✘ Мокируем всё, что попало
- ✘ Какие-то моки, полученные в какой-то фазе луны
- ✘ Моки могут нарушать контракт
- ✘ Моки из эпохи динозавров, правленные напильником

VS

# Осознанное

- ✔ Точно знаем, что мокируем и как
- ✔ Выбираем нужные нам классы эквивалентности
- ✔ Уверены в исполнении контракта
- ✔ Видим дату актуализации моков, уверены в отсутствии ручных правок

# Неосознанное

- ✗ Мокируем всё, что попало
- ✗ Какие-то моки, полученные в какой-то фазе луны
- ✗ Моки могут нарушать контракт
- ✗ Моки из эпохи динозавров, правленные напильником

VS

# Осознанное

- ✓ Точно знаем, что мокируем и как
- ✓ Выбираем нужные нам классы эквивалентности
- ✓ Уверены в исполнении контракта
- ✓ Видим дату актуализации моков, уверены в отсутствии ручных правок



# Контракт обеспечивает качество моков



- ✓ Это база для доверия изолированным тестам
- ✓ Позволяет анализировать как полноту самих моков, так и полноту использования моков

# Генерируем тест-кейсы

```
Document1 x Document2

interface Bar {
  foo: string
}
```

- ```
Document1 Document2 x
```
- Проверка наличия свойства foo
  - Тип foo – только string
  - Запрет присвоения не-string в foo
  - Объект без foo не должен проходить валидацию
  - Пустая строка в foo допустима
  - Дополнительные поля разрешены при совместимости

# Генерируем тест-кейсы лучше

Document1 × Document2

```
/**
 * Номер телефона для пользователей из России.
 *
 * Поддерживаемые форматы:
 * – Начинается с '+7' и содержит ещё 10 цифр (например, +79123456789)
 * – Начинается с '8' и содержит ещё 10 цифр (например, 89123456789)
 * – Начинается с '9' и содержит 10 цифр (считается, что это номер в формате без
кода; подразумевается +7, например, 9123456789)
 *
 * Запрещены:
 * – Пробелы, дефисы, скобки и другие символы – только цифры и '+' в начале
 * – Более или менее 10 цифр после префикса
 * – Коды стран, кроме +7 или эквивалентов (8, 9 как внутренние)
 *
 * Валидные примеры:
 * +79123456789
 * 89123456789
 * 9123456789
 *
 * Невалидные примеры:
 * 79123456789 (без + у 7)
 * +7 912 345 67 89 (есть пробелы)
 * +7912345678 (9 цифр)
 * +11234567890 (не РФ)
 */
interface Baz {
    phone: string;
}
```

Document1 Document2 ×

- Поле phone обязательно
- phone – только строка
- Номер начинается с +7 и содержит 11 цифр
- Номер начинается с 8 и содержит 11 цифр
- Номер начинается с 9 и содержит 10 цифр (автопрефикс +7)
- Разрешены только цифры и + в начале
- Запрещены пробелы, скобки, дефисы
- Длина после форматирования – ровно 11 цифр
- Номера с кодом страны ≠ +7 не проходят
- Примеры валидных форматов проходят проверку
- Примеры невалидных форматов отклоняются

# Тест-кейсы для потребителей

```
18 describe('Транспорт: Взаимодействие между вебом и мобильным приложением', () => {
19   it('SYS.TRANS.001 Успешный запрос от веба', () => {
20     allure.description(dedent(`
21       🎯 Цель: Проверить успешную отправку и обработку запроса от веба к мобильному приложению
22
23       🛠 Шаги:
24       1. Веб вызывает window.nativeApp.postMessage с валидным JSON-объектом.
25       2. Объект содержит: type (строка), requestId (уникальная строка), value: null.
26       3. Мобильное приложение получает сообщение, парсит его.
27       4. МП обрабатывает запрос и отправляет ответ через window.__sendToWeb.
28
29       ✅ Ожидаемый результат:
30       - Мобильное приложение возвращает ответ с тем же requestId.
31       - status: 200 (OK).
32       - Ответ содержит корректные поля: type, requestId, status, value.
33       - Ответ возвращается **всегда**, даже если нет данных для передачи.
34
35       💡 Ценность теста:
36       Это базовый сценарий, подтверждающий работоспособность RPC-механизма.
37       Гарантирует, что веб может инициировать запрос и получить подтверждение его обработки.
38     `));
39   });
40
41   it('SYS.TRANS.002 Запрос с объектом в value', () => {
42     allure.description(dedent(`
43       🎯 Цель: Убедиться, что веб может передать структурированные данные в запросе.
44
```



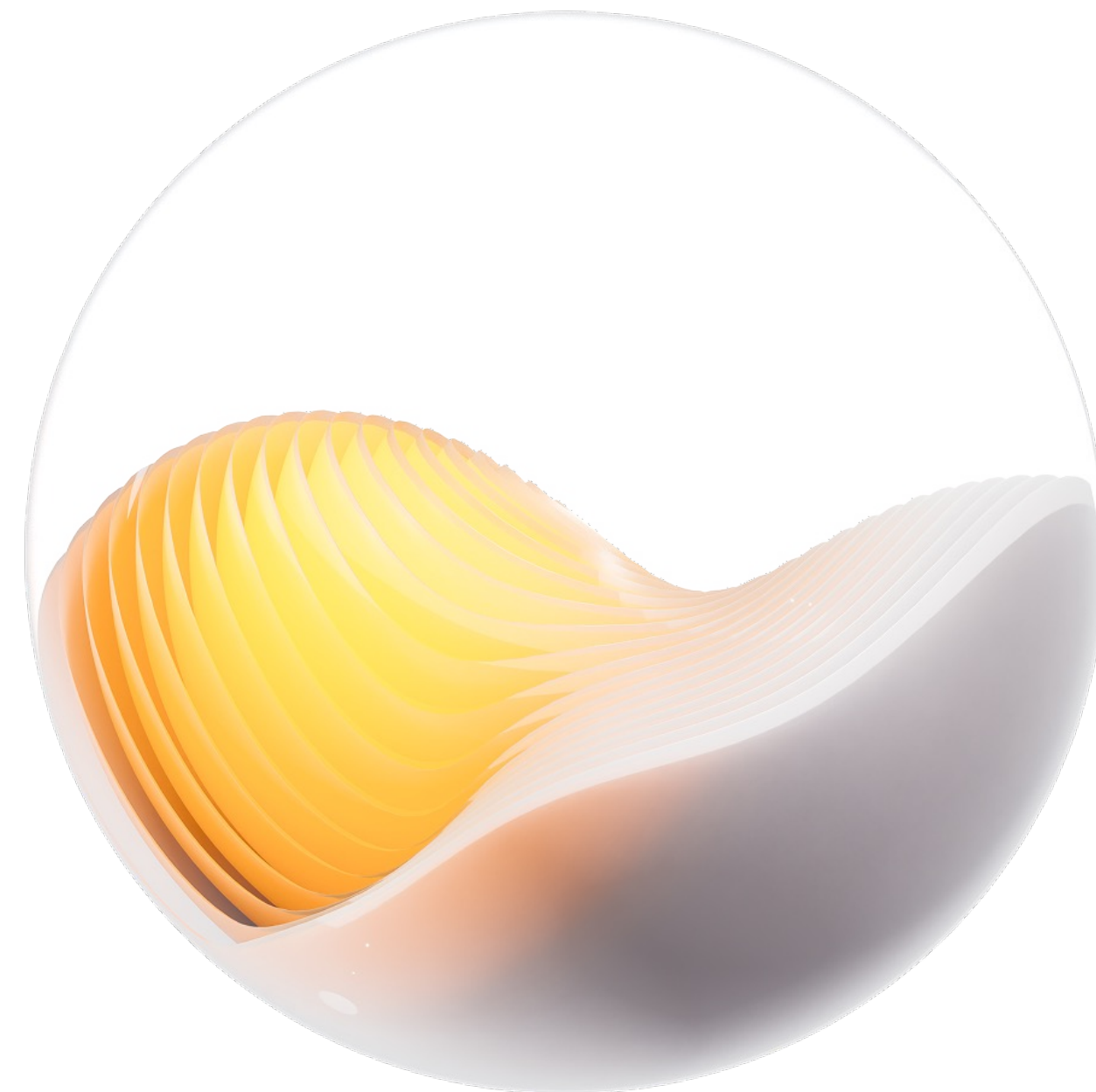
# Контракт — источник для тест-кейсов



- ✓ LLM генерирует тест-кейсы
- ✓ Больше доки → лучше тест-кейсы
- ✓ LLM может оценить полноту покрытия
- ✓ LLM валидирует совместимость тестов и контракта

# Высокий порог входа в систему

- Системы большие и сложные
- В разных репах и на разных языках
- Много легаси



# Генерируемая документация



В синке с контрактом



Просачивается в IDE



Доступна не разработчикам



Документирование — first class citizen

Протокол общения | Webview Contract

coretech-frontend.pages.devplatform.tcsbank.ru/webview-contract/docs/...

Webview Contract Документация Search Nestor search

Введение  
adrs  
meetings  
Web SDK  
Bridge Fixture  
Web SDK для MB  
webview-bridge-mock  
webview-contract  
Добавить в контракт  
Cookie параметры  
GET параметры  
Начальные клиентские параметры  
Нативные сообщения  
Статичные параметры  
Протокол общения  
Экранные сообщения  
Веб-сообщения

webview-contract > Протокол общения

## Протокол общения

Давай ответим на вопрос: где в `window` находится метод для отправки сообщений, и в каком формате их слать?

Протокол общения задаёт общий контракт общения натива с вебом. Все сообщения, включая частные, должны передаваться в этом формате.

Протокол не решает проблемы унификации платформ, совместимости со старыми версиями, и удобства использования (promise-like фасад для продуктовой разработки) — эти проблемы решаются абстракциями на уровне выше (в частности, в web-bridge sdk).

Тезисы:

1. Натив ожидает, что метод `window.__sendToWeb` всегда есть.
2. Для отправки сообщений в натив используется методы `window.nativeApp.postMessage` (Android) и `window.webkit.messageHandlers.nativeApp.postMessage` (iOs)
3. Эти два метода используются как для сообщений-запросов, так и для сообщений-ответов.
4. У всех ответов есть поле `status`, этим они отличаются от запросов.
5. В сообщениях-ответа используется `value` с полем `error`, если

Interfaces

- BasicMessage
- ErrorValue
- RequestMessage
- ResponseMessage
- WindowWithPostMessageAndroid
- WindowWithPostMessageIos

Type Aliases

- Message
- UUIDv4

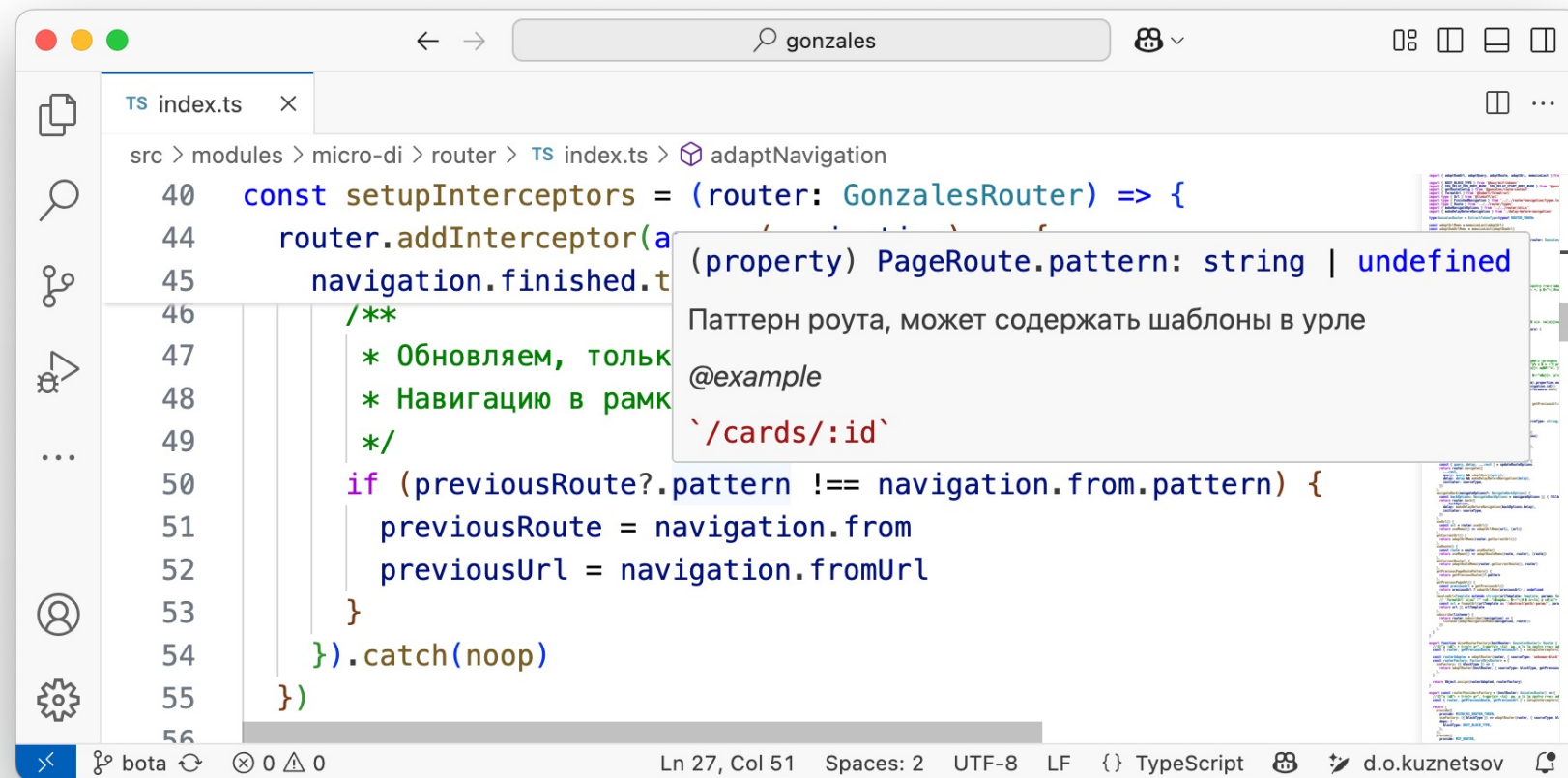
Window

- WindowWithTransport

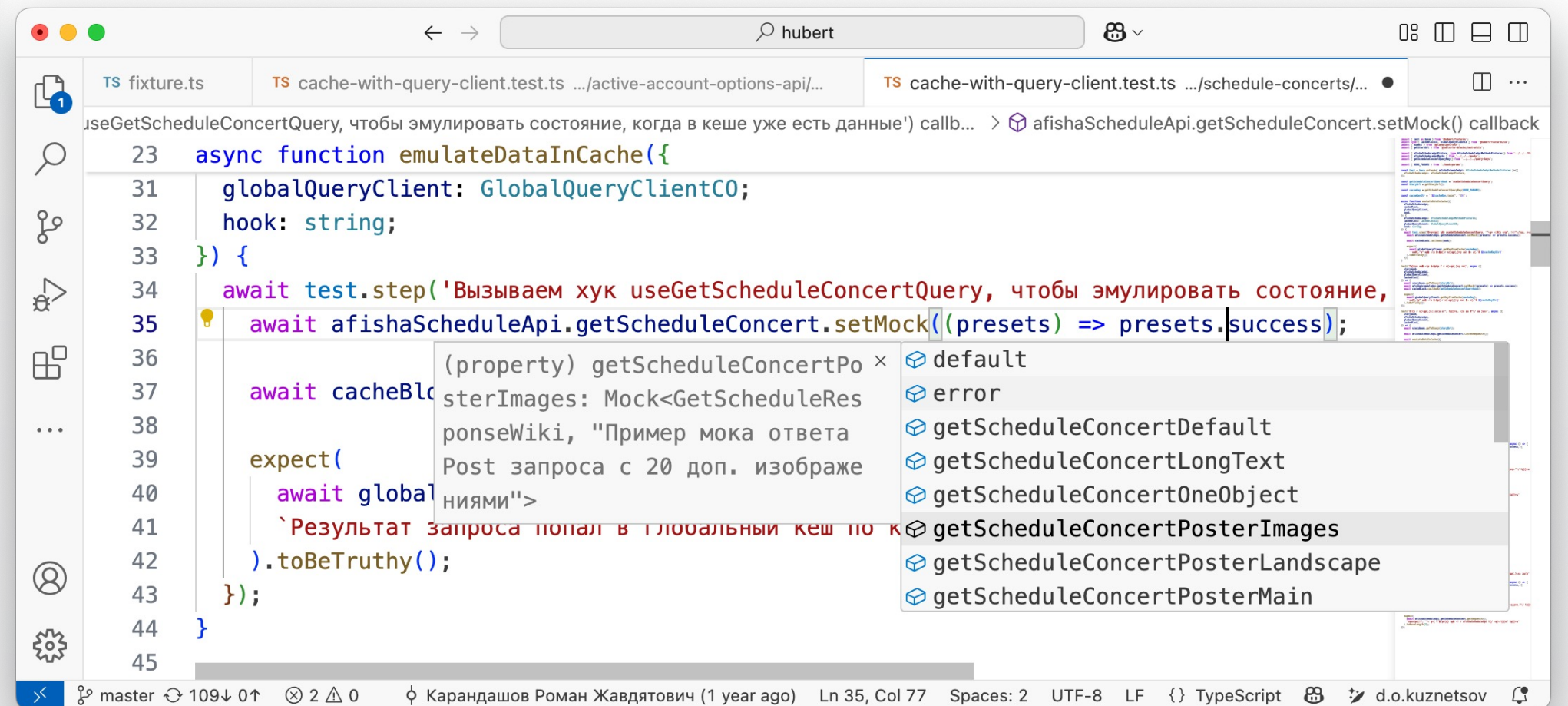
Методы

- PostMessage()
- SendToWeb()

# Примеры интеграций в IDE



The screenshot shows a VS Code editor window with the file path `src > modules > micro-di > router > TS index.ts`. The code defines a function `setupInterceptors` that takes a `GonzalesRouter` and returns a function. It uses `router.addInterceptor` to add a navigation interceptor. A tooltip is displayed over the `navigation.from.pattern` property, showing its type as `(property) PageRoute.pattern: string | undefined` and a description: "Паттерн роута, может содержать шаблоны в урле". An example is provided: `@example '/cards/:id'`. The code includes comments in Russian: `/** * Обновляем, только * Навигацию в рамках */`. The status bar at the bottom shows "Ln 27, Col 51", "Spaces: 2", "UTF-8", "LF", and "TypeScript".



The screenshot shows a VS Code editor window with the file path `TS cache-with-query-client.test.ts .../schedule-concerts/...`. The code defines an `async function emulateDataInCache` that uses `globalQueryClient` and `afishaScheduleApi.getScheduleConcert.setMock`. A tooltip is displayed over the `presets` argument, showing a list of mock presets: `default`, `error`, `getScheduleConcertDefault`, `getScheduleConcertLongText`, `getScheduleConcertOneObject`, `getScheduleConcertPosterImages`, `getScheduleConcertPosterLandscape`, and `getScheduleConcertPosterMain`. The code includes a comment in Russian: `await test.step('Вызываем хук useGetScheduleConcertQuery, чтобы эмулировать состояние, когда в кеше уже есть данные') callb...`. The status bar at the bottom shows "Ln 35, Col 77", "Spaces: 2", "UTF-8", "LF", and "TypeScript".



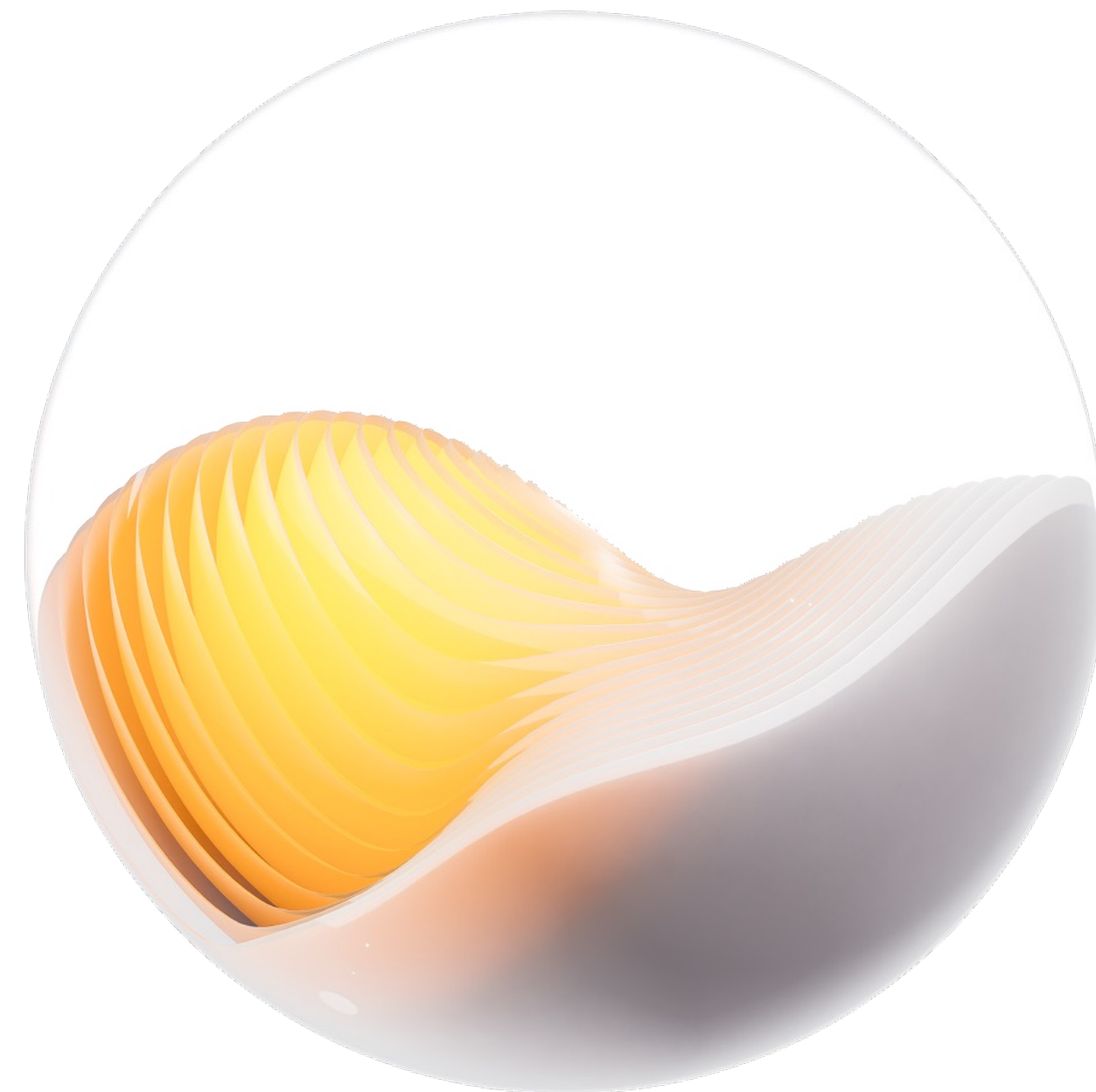
# Контракт это навигатор по системе



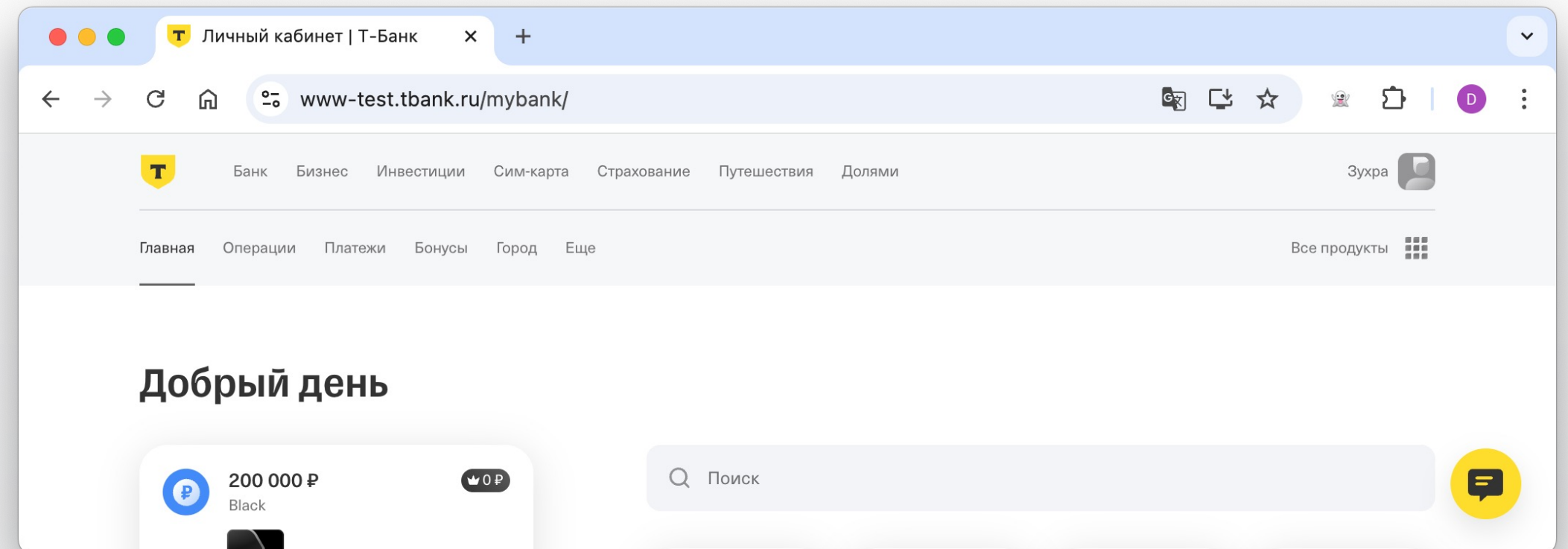
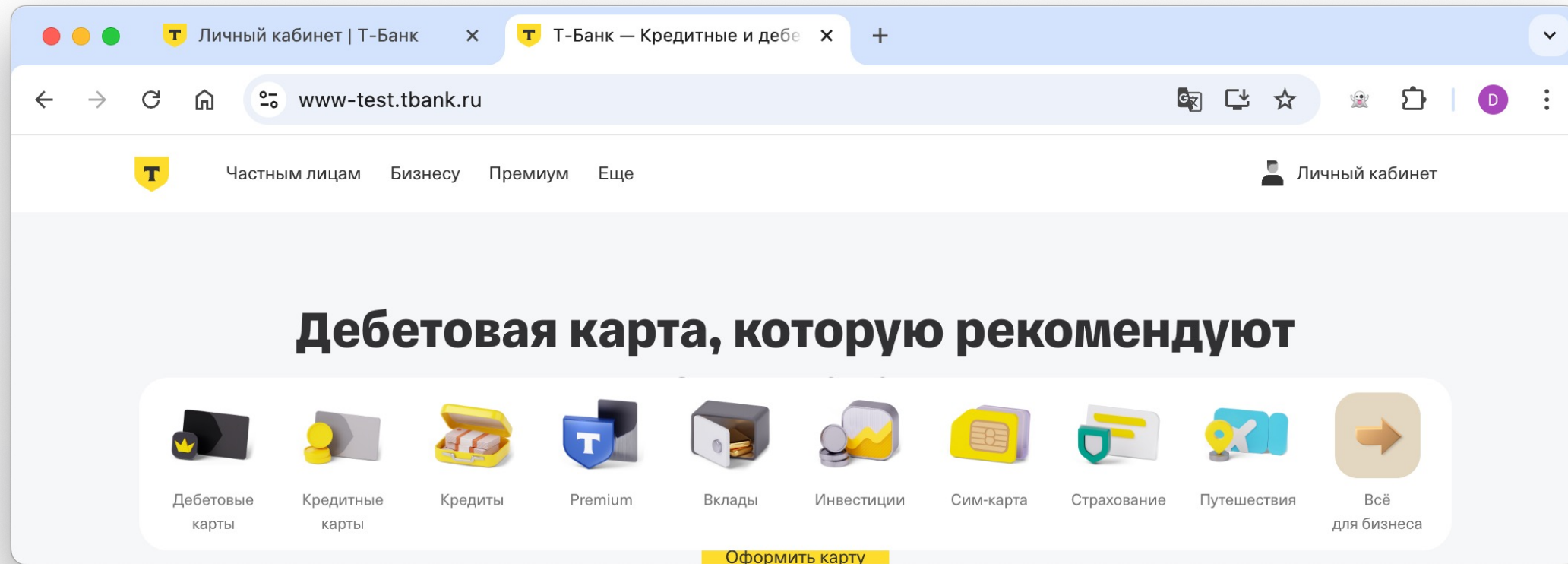
- ✓ Чтобы знать систему не нужно лезть в код
- ✓ В коде понятны смыслы

# Не работает в другом окружении

- Переиспользование — база в IT
- Совместимость трудно обеспечить
- Дорого поддерживать



# Задача: перенести хедер



# Без контракта

- ✘ Запускаем хедер как есть
- ✘ Чиним все падения
- ✘ Надеемся, что серьёзных багов не осталось
- ✘ Поддерживаем два приложения

VS

# С контрактом

- ✓ Анализируем контракты приложений
- ✓ Пишем адаптеры, добавляем сущности
- ✓ Подключаем хедер не глядя
- ✓ Подключаем ещё 110 блоков не глядя

## Без контракта

- ✘ Запускаем хедер как есть
- ✘ Чиним все падения
- ✘ Надеемся, что серьёзных багов не осталось
- ✘ Поддерживаем два приложения

VS

## С контрактом

- ✓ Анализируем контракты приложений
- ✓ Пишем адаптеры, добавляем сущности
- ✓ Подключаем хедер не глядя
- ✓ Подключаем ещё 110 блоков не глядя



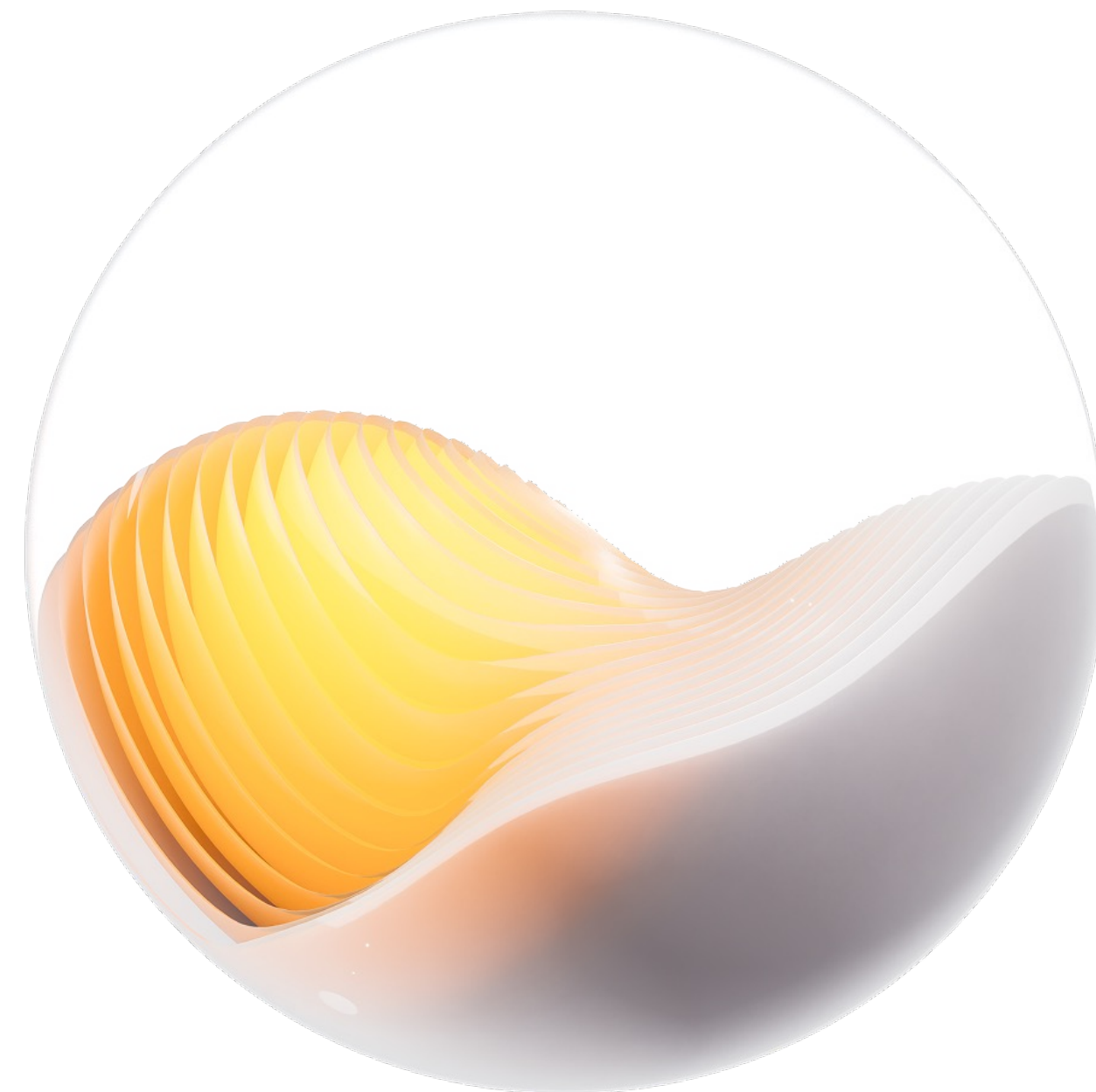
# Контракт гарантирует совместимость

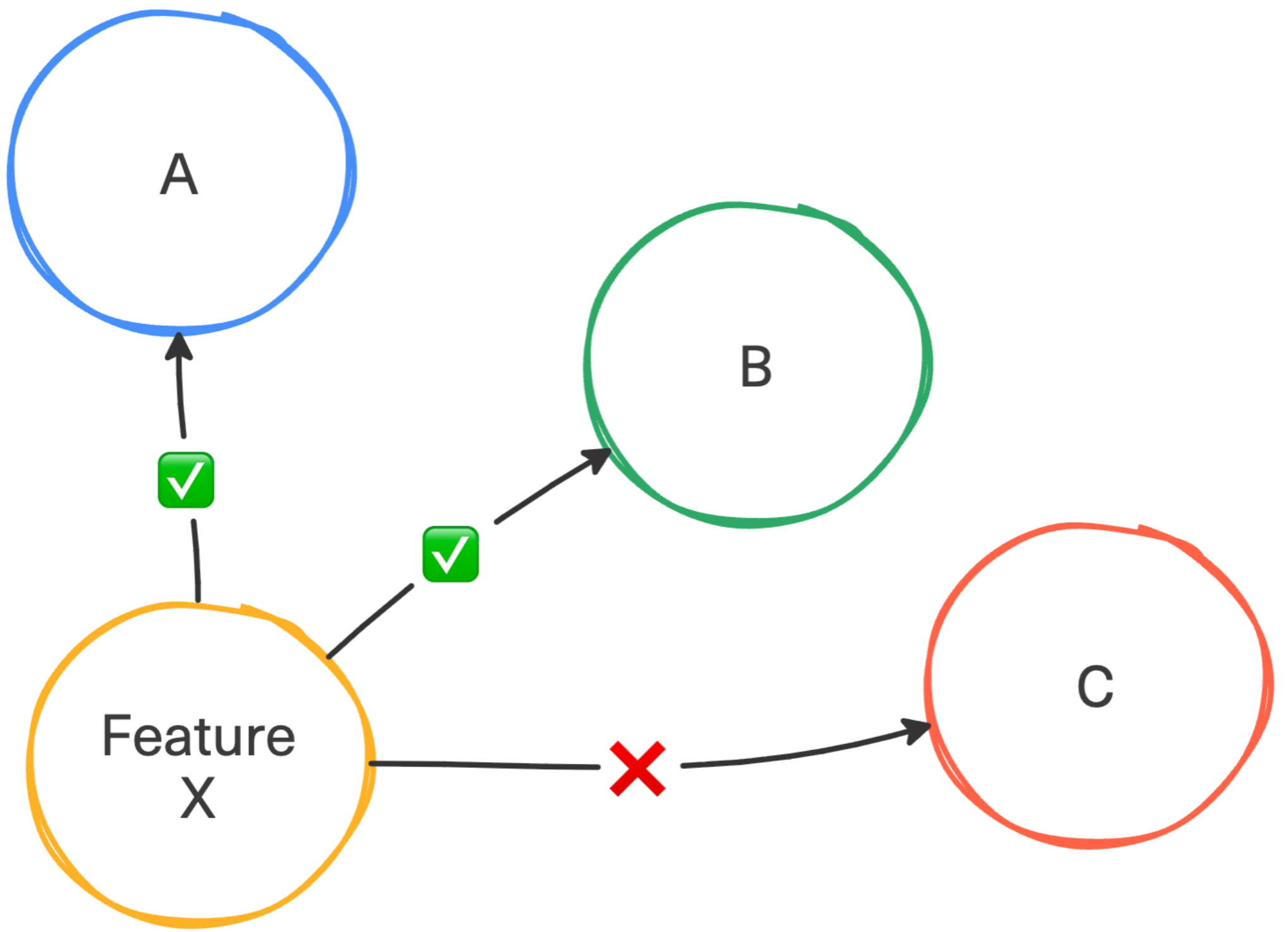
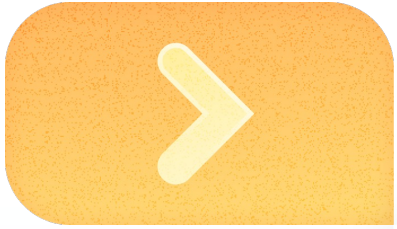


- ✓ Легко проанализировать
- ✓ При разных форматах написать адаптер
- ✓ Точно видим, чего не хватает
- ✓ Масштабируемо

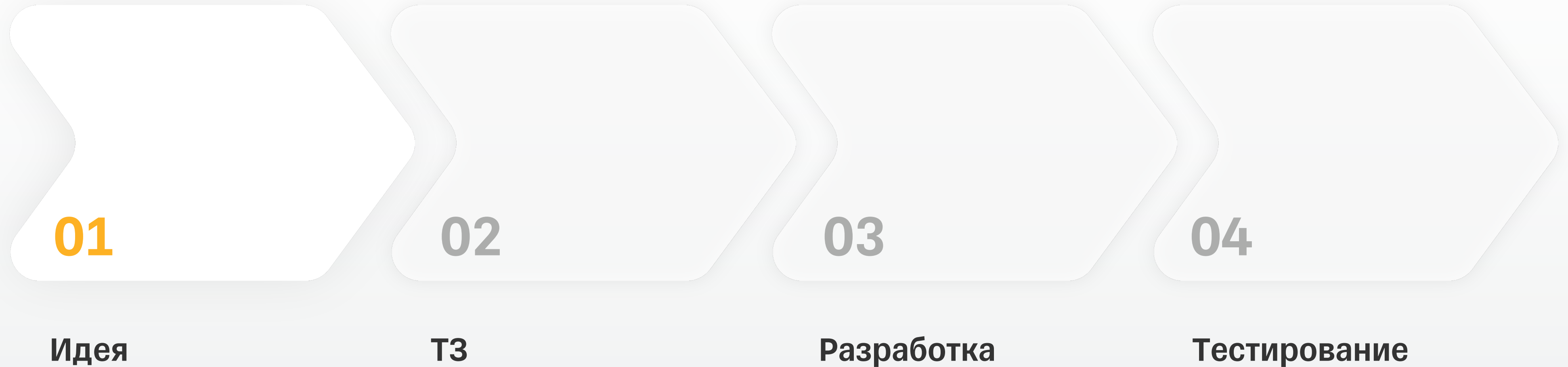
# Как оценить риски по срокам?

— Планировали сделать за неделю, затянулось на год





# Смещаем влево поиск рисков





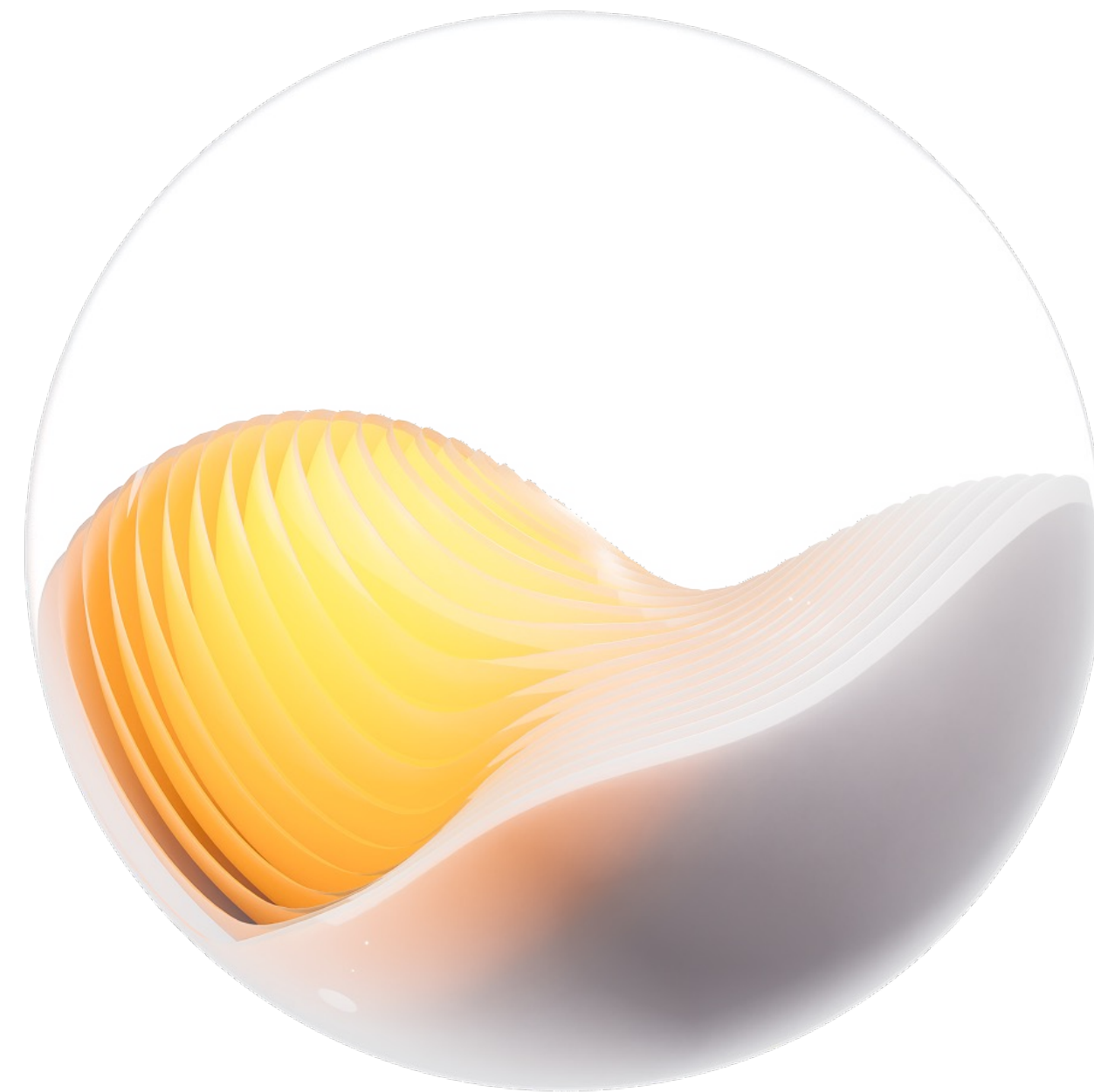
## **Контракт определяет локальность фичи**



- Если всего хватает → риски обычные
- Если чего-то не хватает → риски повышенные

# Контракт устаревает

Если за контрактом не следить, он  
порастает мхом



## Внедрим насильно

- ✘ Требуется запрещающих обязательств
- ✘ Будет обходить при первой возможности
- ✘ Ушёл в отпуск — всё сломалось

VS

## Сделаем полезным

- ✔ Разработчики подсаживаются на типы и подсказки в IDE
- ✔ Тестировщики генерируют моки и тесткейсы
- ✔ Сапортеры ссылаются на документацию

## Внедрим насильно

- ✗ Требуется запрещающих обязательств
- ✗ Будет обходить при первой возможности
- ✗ Ушёл в отпуск — всё сломалось

VS

## Сделаем полезным

- ✓ Разработчики подсаживаются на типы и подсказки в IDE
- ✓ Тестировщики генерируют моки и тесткейсы
- ✓ Сапортеры ссылаются на документацию



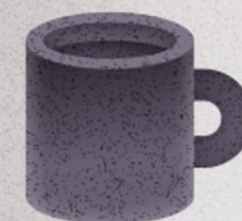
# Полезный контракт = бессмертный контракт

Потому что все хотят его обновить



03

# Практические советы



# Когда контракт не нужен?

- ✓ Не договорились с людьми
- ✓ Короткоживущий проект
- ✓ Небольшой проект для одной команды
- ✓ Хватает системного уровня тестирования, мало фич
- ✓ Нет оверхеда на взаимодействие частей системы (сетевое или ресурсного)

# Что использовать?

- ✓ TypeScript, JSON Schema
- ✓ JSDoc + TypeDoc + Docusaurus \*
- ✓ npm для поставки типов и инструментов.
- ✓ s3/pages
- ✓ LLM

*\*Meta Platforms Inc. — организация признана экстремистской, её деятельность запрещена на территории России.*

# Начни с малого!

- ✓ Выбери критичную и проблемную фичу
- ✓ Формализуй контракт
- ✓ Проверь, все ли понимают контракт одинаково
- ✓ **Договорись, что все следуют этому контракту!**
- ✓ Подумай над всем остальным: документация, тесткейсы, интеграции...

# Рецепт: как формализовать контракт?

- ✓ Звоним эксперту, записываем ответы как есть
- ✓ Скармливаем в IIm, просим сделать typescript+jsdoc

```
/**  
 * Открывает документ  
 */  
function openDocument() {  
    //  
}
```

Напиши хороший jsdoc для typescript интерфейса, для параметра appVersion по списку тезисов, которые я приведу ниже:

appVersion это версия мобильного приложения

Версионирование сквозное, одинаковое для ios и android, подробнее см.

<http://example.com/>

appVersion передается из мобильного приложения в веб, открытый через вебвью

appVersion нужен системе веб-аналитики, чтобы знать, в каких версиях и в каком

соотношении открывается наш веб

appVersion также может быть использован для таргетированного изменения логики на

стороне веба для каких-то взаимодействий — например, если мы узнали, что в

конкретной версии мобильного приложения баг, и взаимодействие в этой версии нужно

выключить

Исправить баг в мобильном приложении так просто не получится, потому что любая

конкретная версия живет в проде до 5 лет

Примеры значений: 7.19, 8.0

Параметр появился в версии 7.19

Используй ключевые слова jsdoc там, где это уместно.

09:46

# Результат

- ✓ Не терзаем разработчика
- ✓ Отделяем форму от содержания
- ✓ Лучше шарим контекст
- ✓ Находим белые пятна

```
TS params.ts x
packages > webview-contract > TS params.ts > Params
19 export interface Params {
48   webviewVersion: string
49   /**
50    * Версия мобильного приложения, из которого открывается веб-контент через вебвью.
51    *
52    * Версионирование сквозное и одинаковое для iOS и Android.
53    * Подробнее о системе версионирования: {@link https://wiki.tcsbank.ru/x/pxSoIQ |Календарь релизов мобил
54    * Продовые версии всегда состоят из 2 или 3 чисел, разделённых точкой.
55    * Тестовые версии могут иметь любую строку (см. примеры).
56    *
57    * @remarks
58    * Параметр используется системой веб-аналитики для отслеживания,
59    * в каких версиях мобильных приложений и в каком соотношении открывается наш веб.
60    *
61    * Также может применяться для таргетированного изменения логики на стороне веба,
62    * например, чтобы отключить определённое поведение, если известно,
63    * что в конкретной версии мобильного приложения есть баг.
64    *
65    * @important
66    * Исправление багов в уже выпущенных версиях мобильных приложений затруднено,
67    * так как каждая версия поддерживается в проде до 5 лет.
68    *
69    * @example
70    * Примеры корректных значений: `7.19`, `3.1.1`, `2.14.0-rc2-aa5ef9b` (тестовая сборка)
71    *
72    * @since МБ – 7.19
73    */
74   appVersion: string
```

# Рецепт: как сгенерировать тесткейсы?

- ✓ Копируем контракт as is
- ✓ Скармливаем в LLM, просим сделать X
- ✓ Показываем LLM какой X «у нас дома»
- ✓ Автоматизируем

# Где хранить контракт?

## Требования:

- ✓ Ревью и нотификации
- ✓ Автоматизация анализа
- ✓ Простой контрибутинг

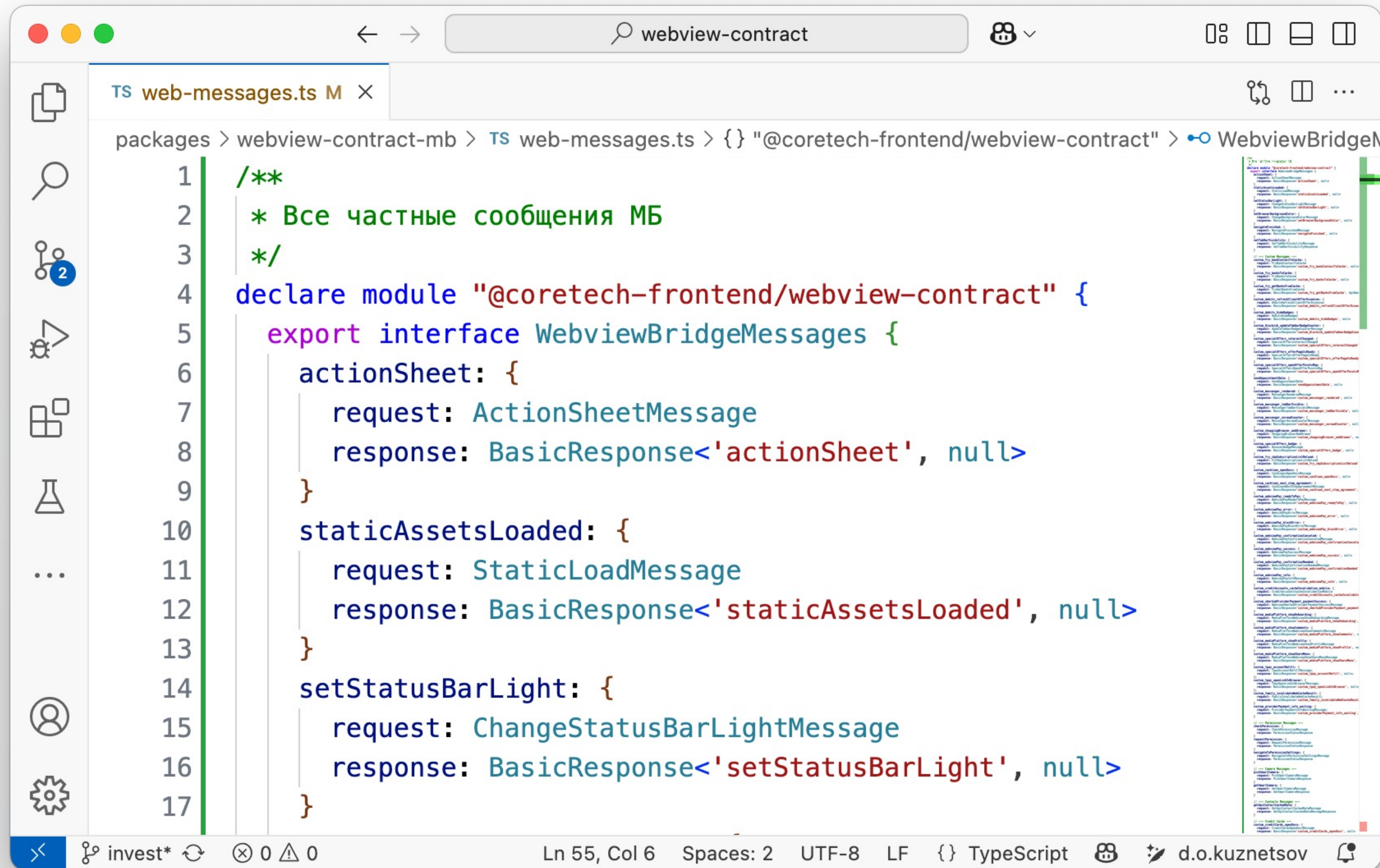
# Где хранить контракт?

## В отдельном репозитории

- ✓ Gitops: контроль версий, ревью, апрувы
- ✓ CI/CD для проверок, генерации и доставки
- ✓ Проще контрибуть
- ✓ Нет обременений

| Approvers | Approvals | Approved by |
|-----------|-----------|-------------|
| basic     | 0 of 1    |             |
| gonzales  | 0 of 1    |             |

# Как расширяем контракты



```
packages > webview-contract-mb > TS web-messages.ts > {} "@coretech-frontend/webview-contract" > WebviewBridgeM  
1 /**  
2  * Все частные сообщения МБ  
3  */  
4 declare module "@coretech-frontend/webview-contract" {  
5   export interface WebviewBridgeMessages {  
6     actionSheet: {  
7       request: ActionSheetMessage  
8       response: BasicResponse<'actionSheet', null>  
9     }  
10    staticAssetsLoaded: {  
11      request: StaticLoadMessage  
12      response: BasicResponse<'staticAssetsLoaded', null>  
13    }  
14    setStatusBarLight: {  
15      request: ChangeStatusBarLightMessage  
16      response: BasicResponse<'setStatusBarLight', null>  
17    }  
18  }
```

Ln 55, Col 6 Spaces: 2 UTF-8 LF {} TypeScript d.o.kuznetsov

# Важнейшие атрибуты контракта

- ✓ Принятое обязательство
- ✓ Типы + смыслы
- ✓ Отражает реальность

# Решаем проблемы вместе

01


Найди проблему

02

Созвонись с  
экспертом

03

Договорись с  
командами



**Разреши  
контракту быть  
полезным!**