



**Большие миграции
для больших проектов**

Алексей Охрименко



t.me/obenjiro



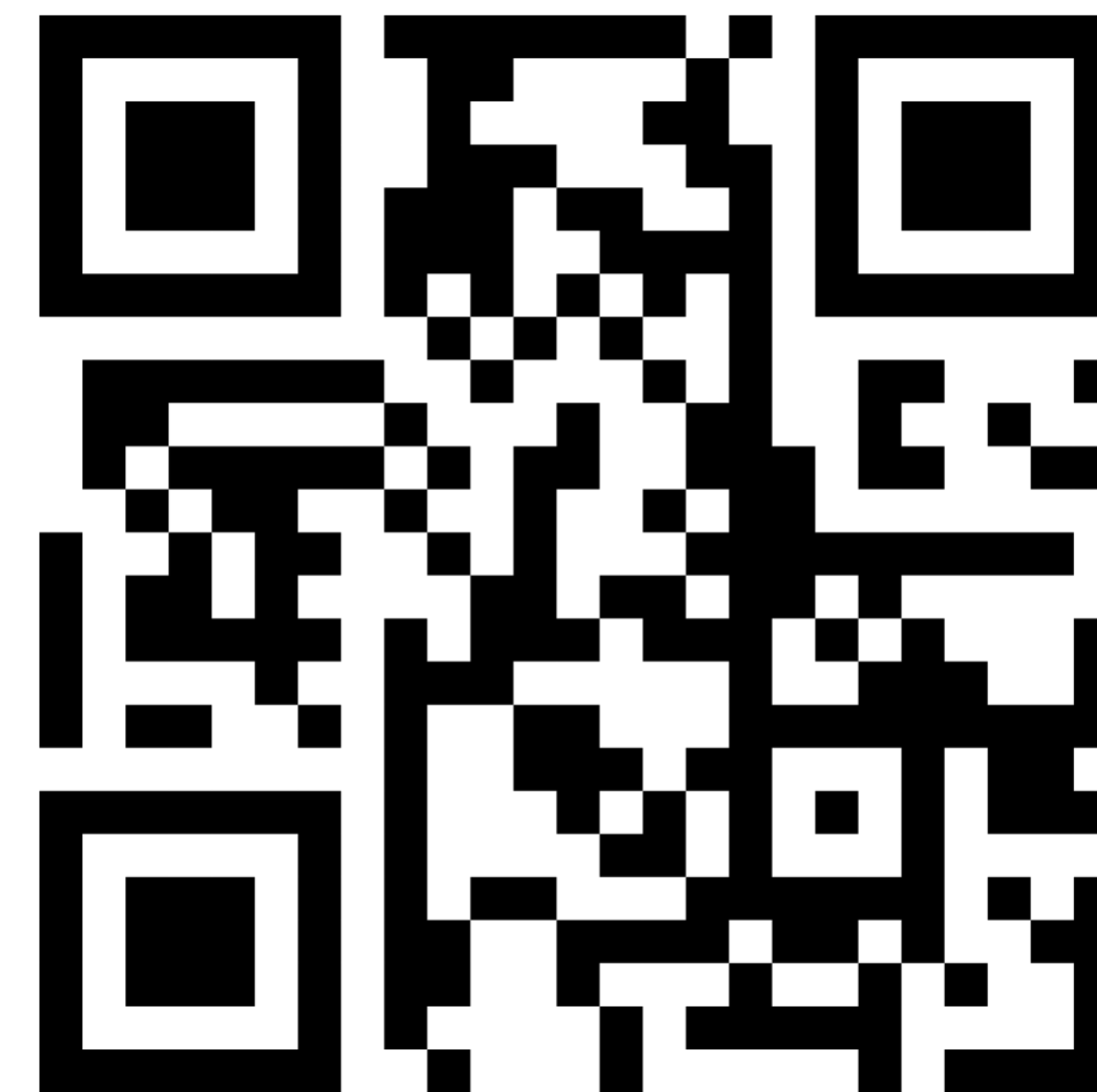
Yandex Tracker



t.me/obenjiro



t.me/tensorflow_js



t.me/tensorflow_py





- Все сказанное — это мое видение
- Как о больших, так и о малых миграциях
- Основано на опыте работы в разных компаниях
- Все иллюстрации сгенерированы нейронками

А в чем проблема?

История Tracker

Взрослый проект (> 12 лет)

Yandex Tracker Дашборды ▾ Доски ▾ Очереди ▾ Задачи ▾ Создать задачу 🔔² ? ⚙️ 👤⁹⁹ Стеша Колпакова

⚙️ Поддержка 🚩

Задачи ³⁹ Проекты Компоненты Версии Статистика Диаграмма Гантта Описание

Все Без резолюции С резолюцией Я автор Я исполнитель Я наблюдатель Избранное Без группировки По версии По исполнителю По проекту По статусу ... ▾ Выбр

без резолюции автор: kolpakova x по исполнителю x 🗑️

Алексей Кустов ¹³

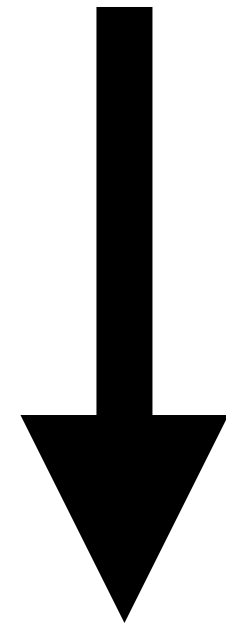
<input type="checkbox"/> SLA ▾	<input type="checkbox"/> Ключ ▾	Задача	Исполнитель ▾	Статус ▾	Автор	Дата начал
<input type="checkbox"/> 🕒 Таймер на паузе Новое правило	• 💬 SUPPORT-51	📄 Предложение по улучшению Стеша Колпакова kolpakova@tram.yasonect.com tech_support	Алексей Кустов	В работе	Стеша Колпакова	—
<input type="checkbox"/> 🚫 Превышено на 1408 ч Новое правило	• 💬 SUPPORT-53	📄 Все сломалось tech_support	Алексей Кустов	Открыт	Стеша Колпакова	—
<input type="checkbox"/> 🚫 Выполнено за 1478 ч Новое правило	↑ 💬 SUPPORT-22	📄 не работает сайт tech_support	Алексей Кустов	Открыт	Стеша Колпакова	22 мар 2019
<input type="checkbox"/> 🚫 Выполнено за 1710 ч Новое правило	↑ 💬 SUPPORT-10	📄 не работает сайт tech_support	Алексей Кустов	Открыт	Стеша Колпакова	22 мар 2019
<input type="checkbox"/> 🚫 Выполнено за 264 ч 3 мин Новое правило	• 📄 SUPPORT-47	📄 обращение tech_support	Алексей Кустов	Открыт	Стеша Колпакова	—
<input type="checkbox"/> 🚫 Выполнено за 677 ч 52 мин Новое правило	• 📄 SUPPORT-36	SUPPORT-5: Запрос 📄 Запрос 2 tech_support	Алексей Кустов	Открыт	Стеша Колпакова	—
<input type="checkbox"/> 🚫 Выполнено за 938 ч Новое правило	↑ 📄 SUPPORT-28	📄 Заявка от Алексей Алексеич Петров tech_support	Алексей Кустов	Открыт	Стеша Колпакова	—
<input type="checkbox"/> ✅ Выполнено за 43 мин 21 с	↓ 💬 SUPPORT-5	📄 Запрос	Алексей Кустов	Открыт	Стеша Колпакова	—



- Техдолг**
- Хочется новенького и т.п**
- Больше не поддерживается**
- Большое обновление**


BEM (enb, borshik, bem.js)

BEM (enb, borshik, bem.js)



Gravity UI (Webpack, React, TS)

Build modern interfaces with the Gravity design system and libraries

 [How to Start](#) [GitHub](#)

Recent updates

3 DAYS AGO

We've published our [illustration library](#) in Figma community.

1 WEEK AGO

We've added two new components: [Palette](#), which allows you to display a grid of icons and emojis, and [Divider](#), which simply divides things.

3 WEEKS AGO

The [illustrations library](#) has been released, providing 21 default images in two themes for the placeholder component.



Built on real-life experience

Conceived as an in-house solution in response to real developers' needs, we released Gravity to the open-source community.



First class design

Experienced designers curate the look of our libraries, ensuring that components are stylish and consistent, with support for dark mode and high-contrast themes.



An evolving ecosystem

Regular feedback from our community of developers allows us to continuously improve our libraries, and break compatibility only when necessary.

Гибкая вариативность

GravityUI Libraries Components Design Icons Telegram ↗ Figma ↗ GitHub ↗

Search by component name

UIKit ^

- Alert
- ArrowToggle
- Breadcrumbs
- Button
- Card
- Checkbox
- DropDownMenu
- Icon

Pagination [Github](#)

The image displays three different pagination UI variants. The first variant is compact, showing navigation arrows, the current page (1), and a page size selector (10). The second variant includes 'First', 'Previous', 'Next', and 'Page #' labels, along with a page size selector. The third variant is a simplified version with 'First', 'Previous', 'Next', and 'Page #' labels and a page size selector.

page: 1

pageSize: 10

total: 30

compact:

showInput:

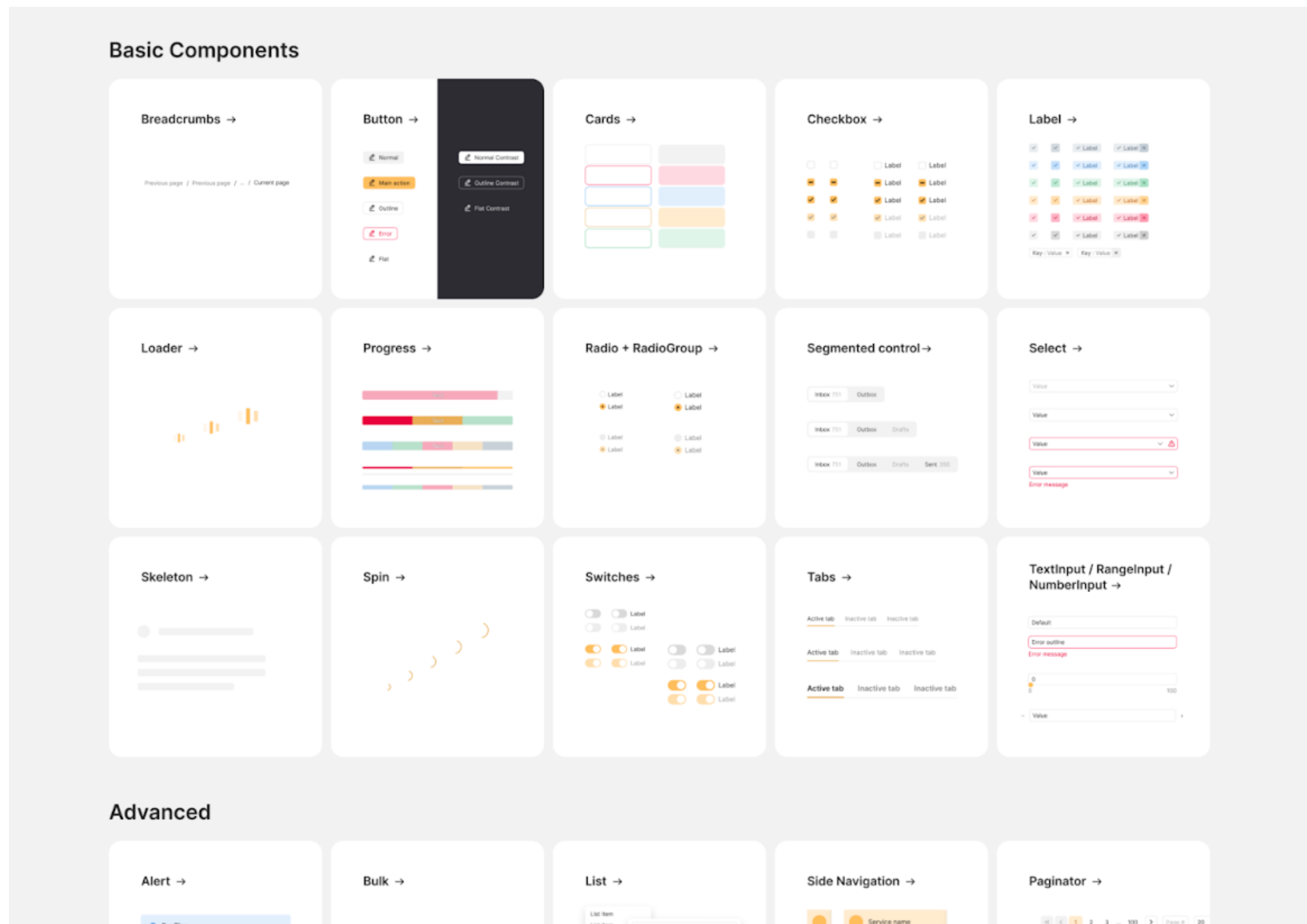
showPages:

> 100 сервисов > 150 контрибьюторов

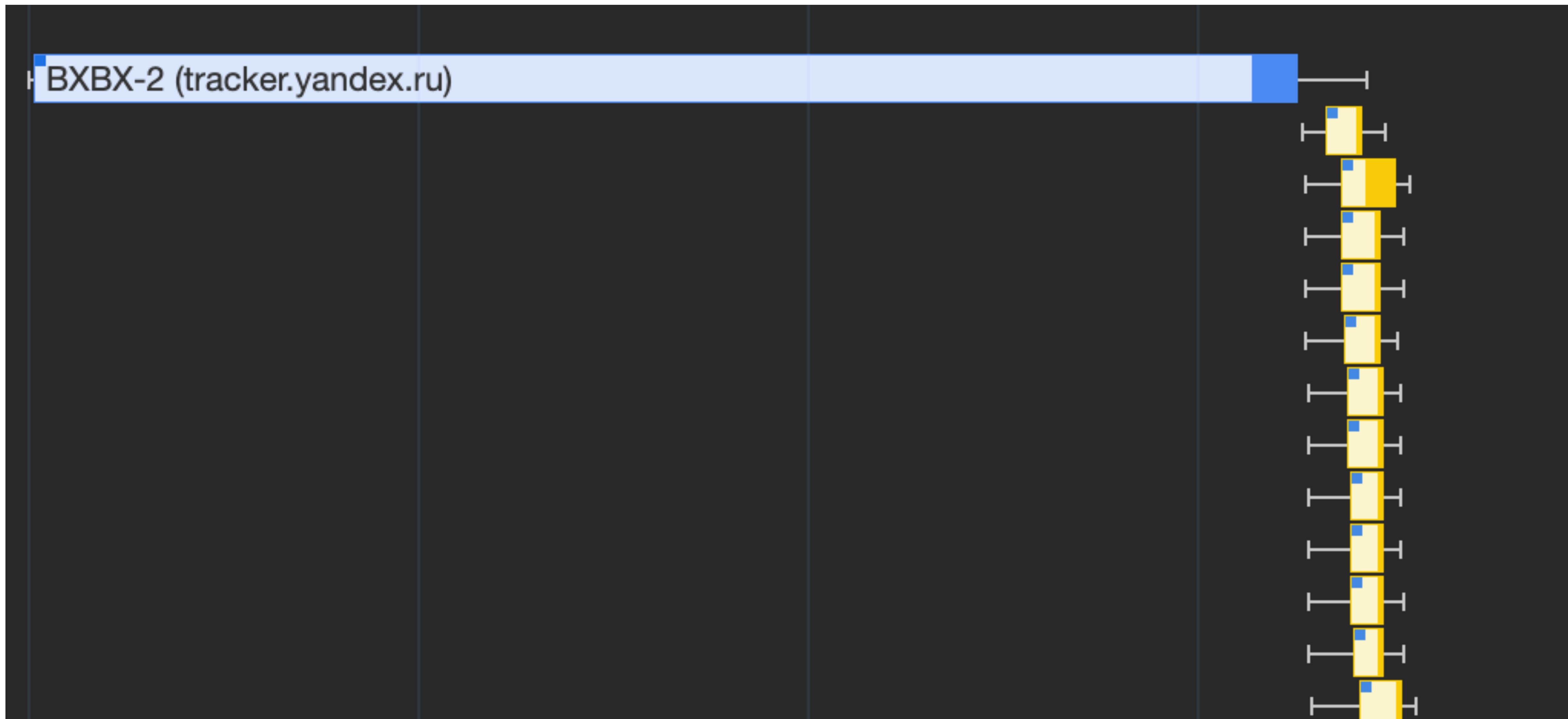
Yandex@Cloud NEBIUS  DoubleCloud  YTsaurus

 Diplodoc  DataLens  YDB

Дизайн-гайды



HTML streaming

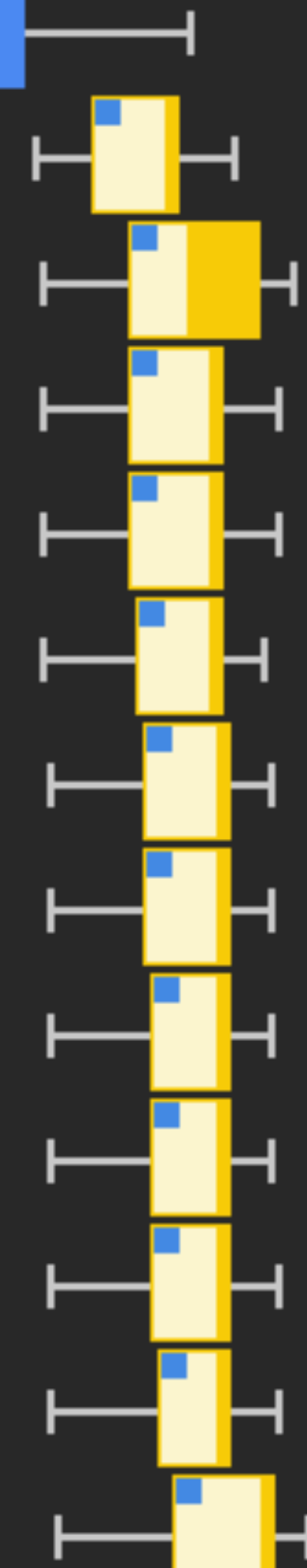


HTML streaming

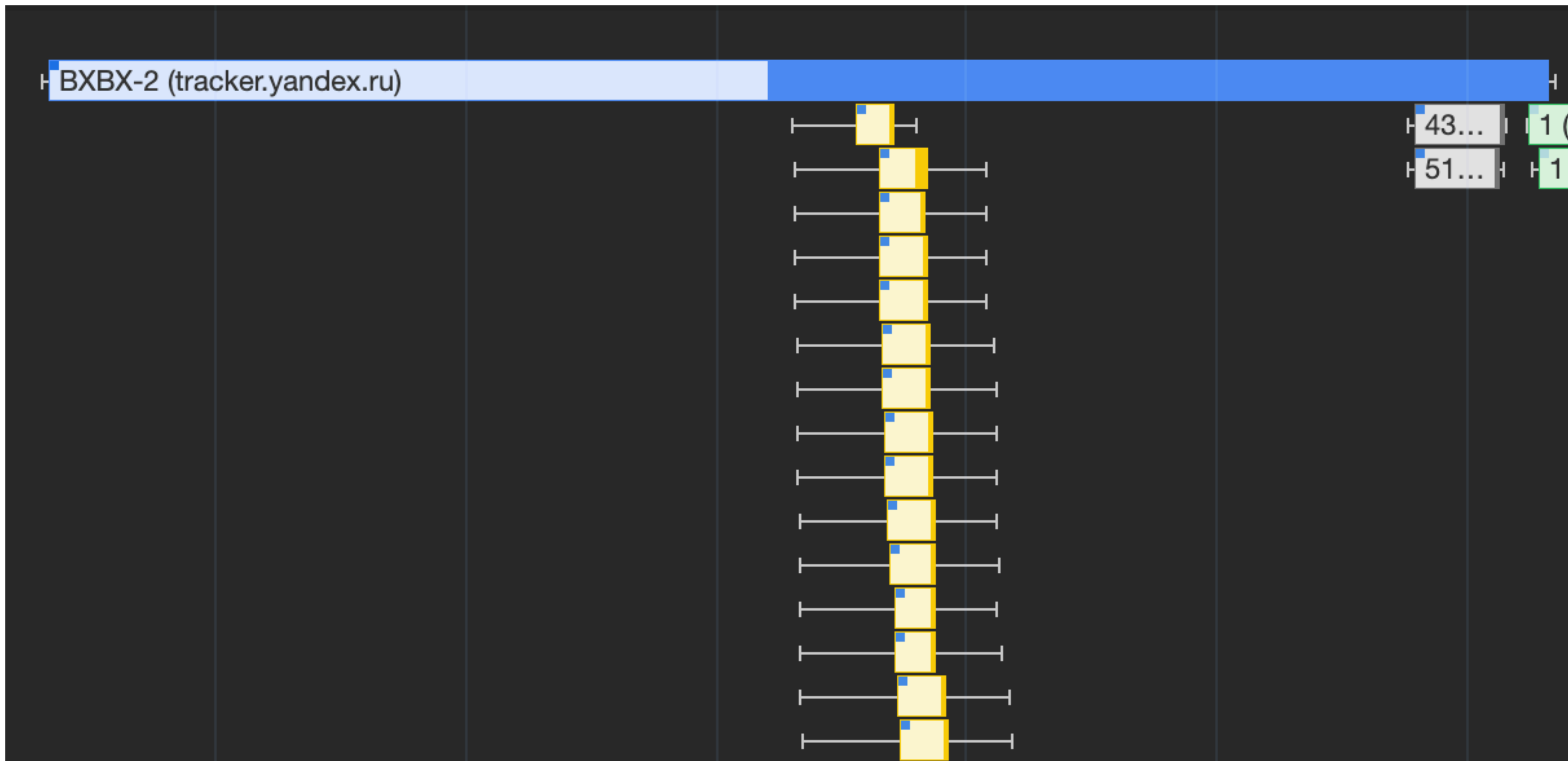
BXBX-2 (tracker.yandex.ru)



Request to backend

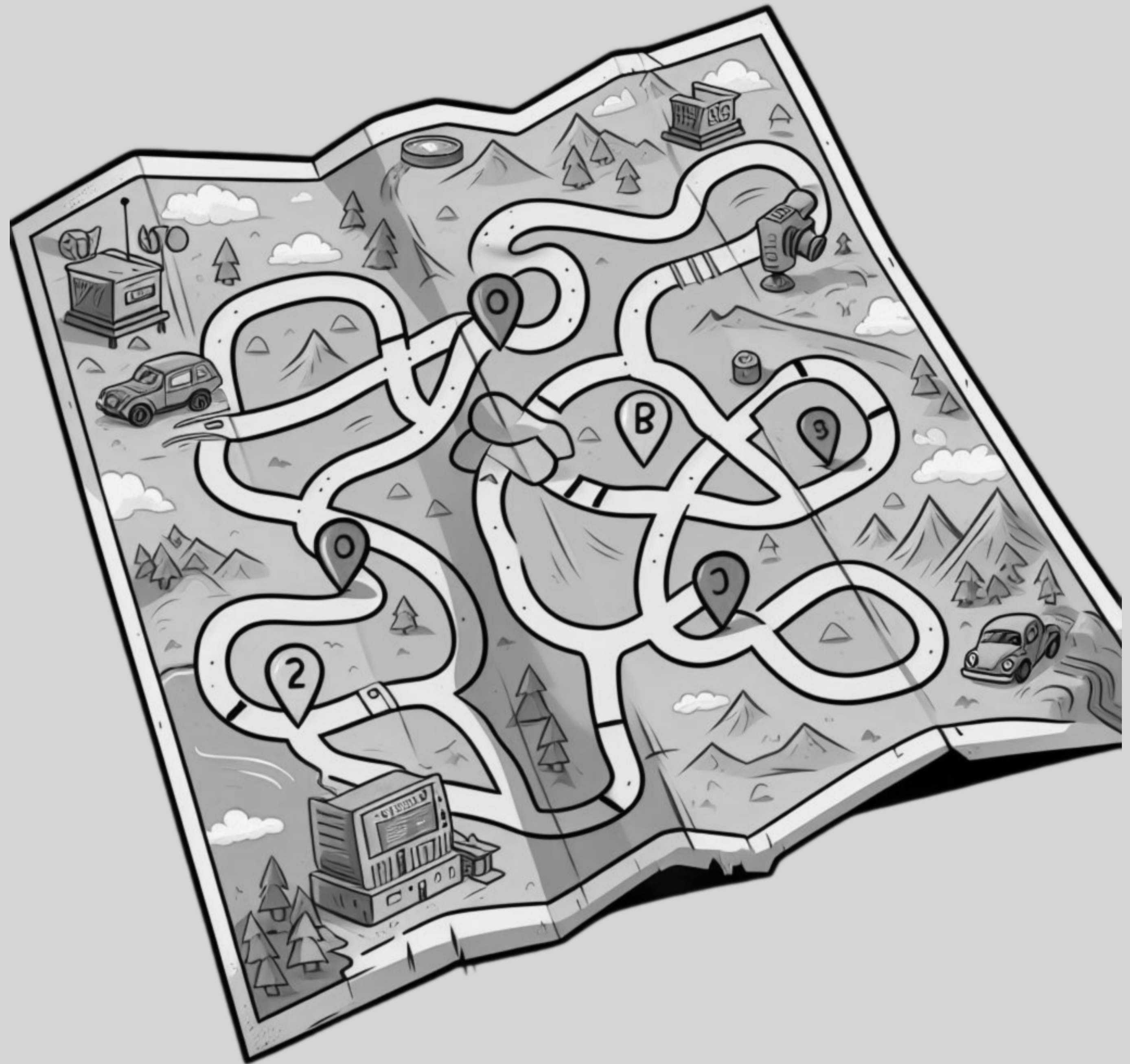


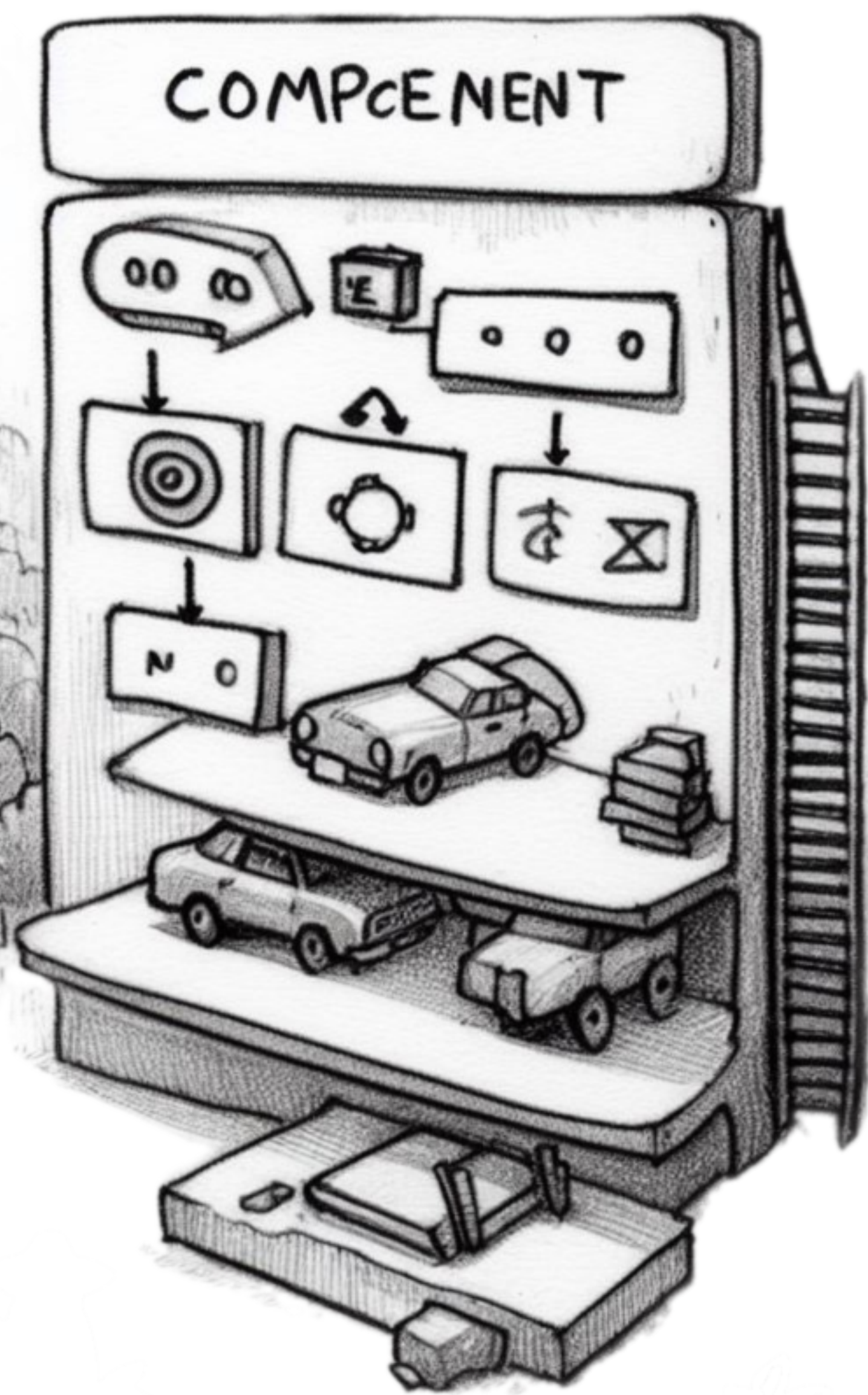
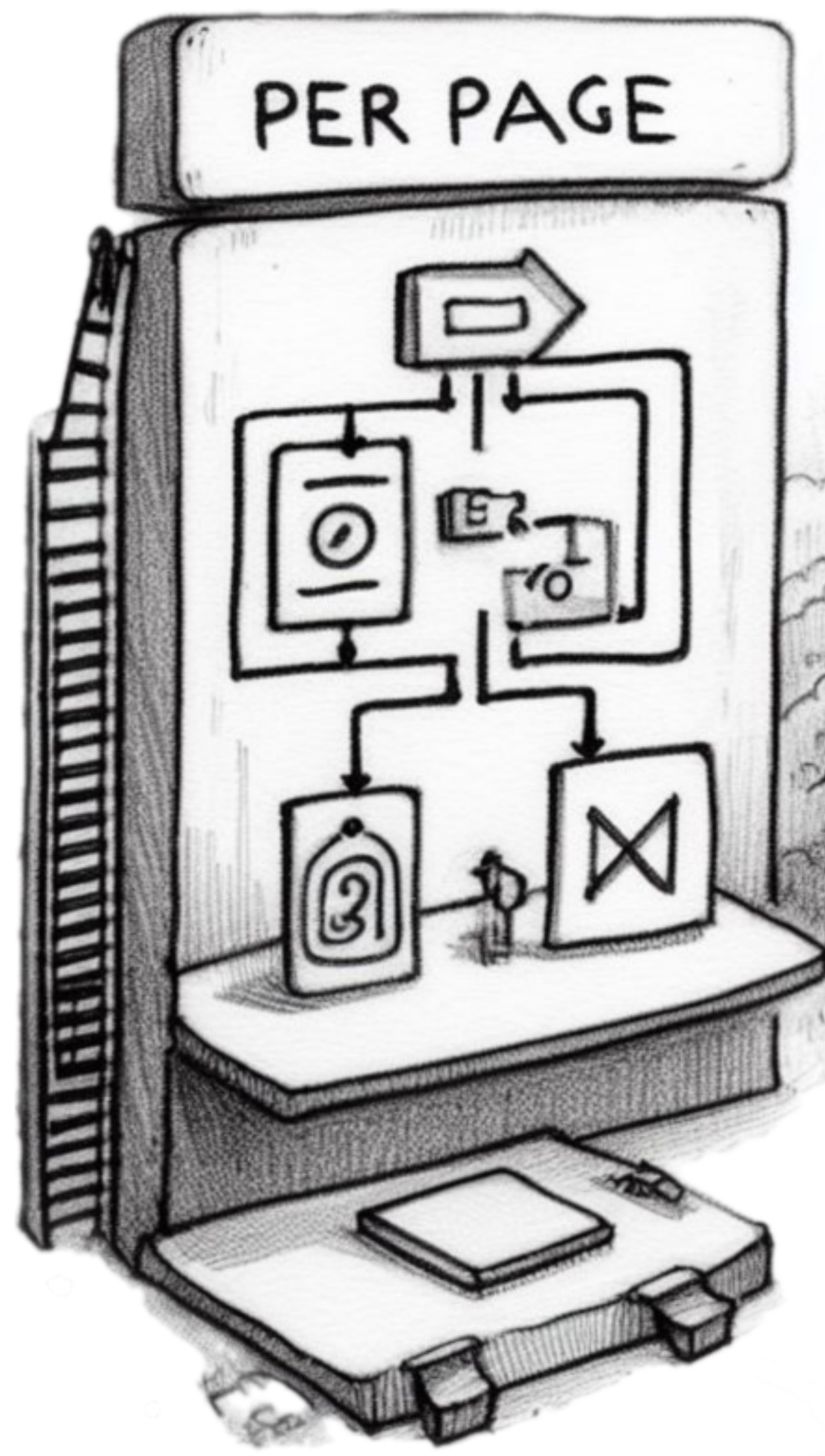
HTML streaming



Ок, мы поняли: пора переписывать

Но как?



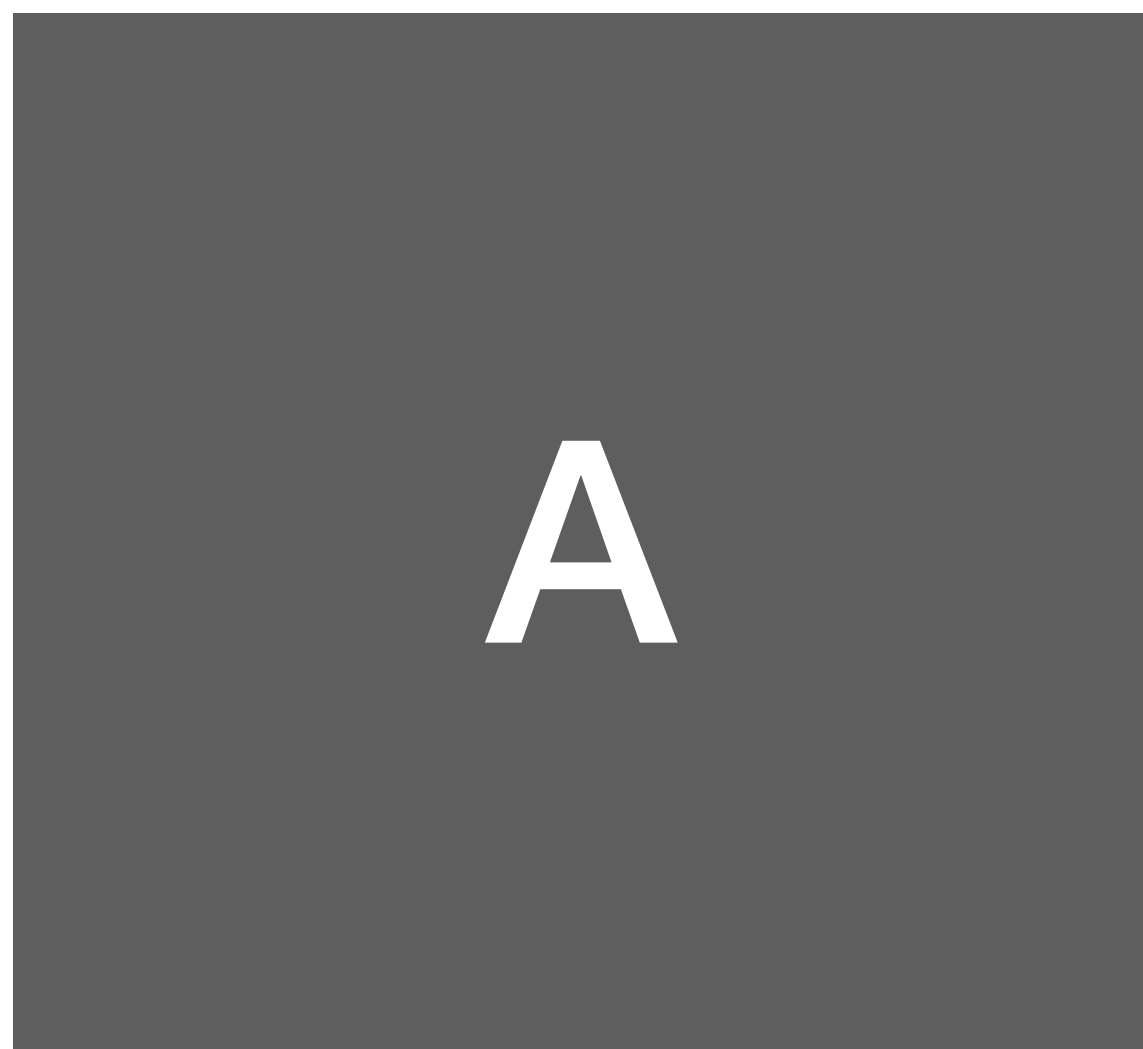


В рамках монолита

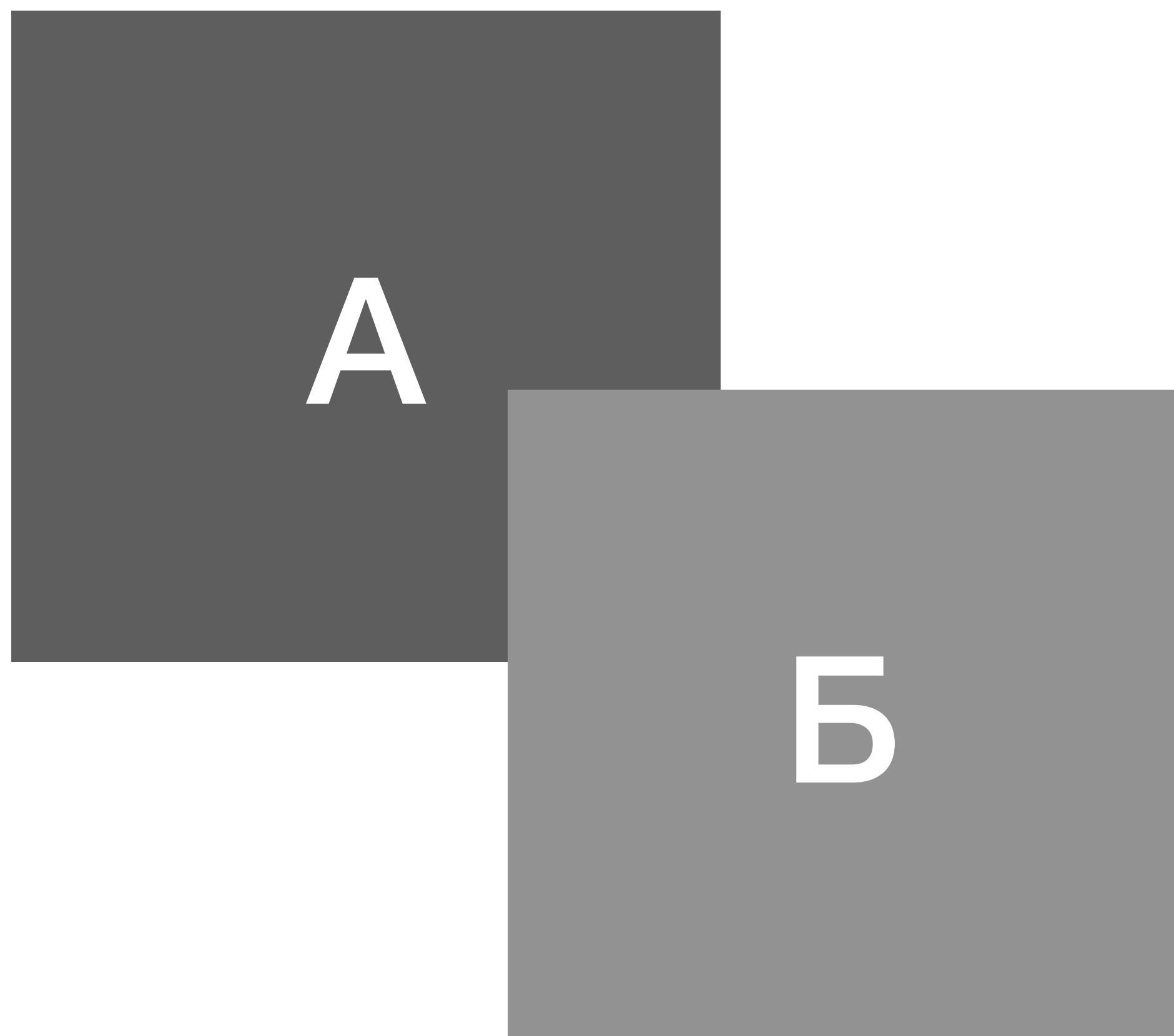
Микрофронтенд (2 варианта)

За одним Proху

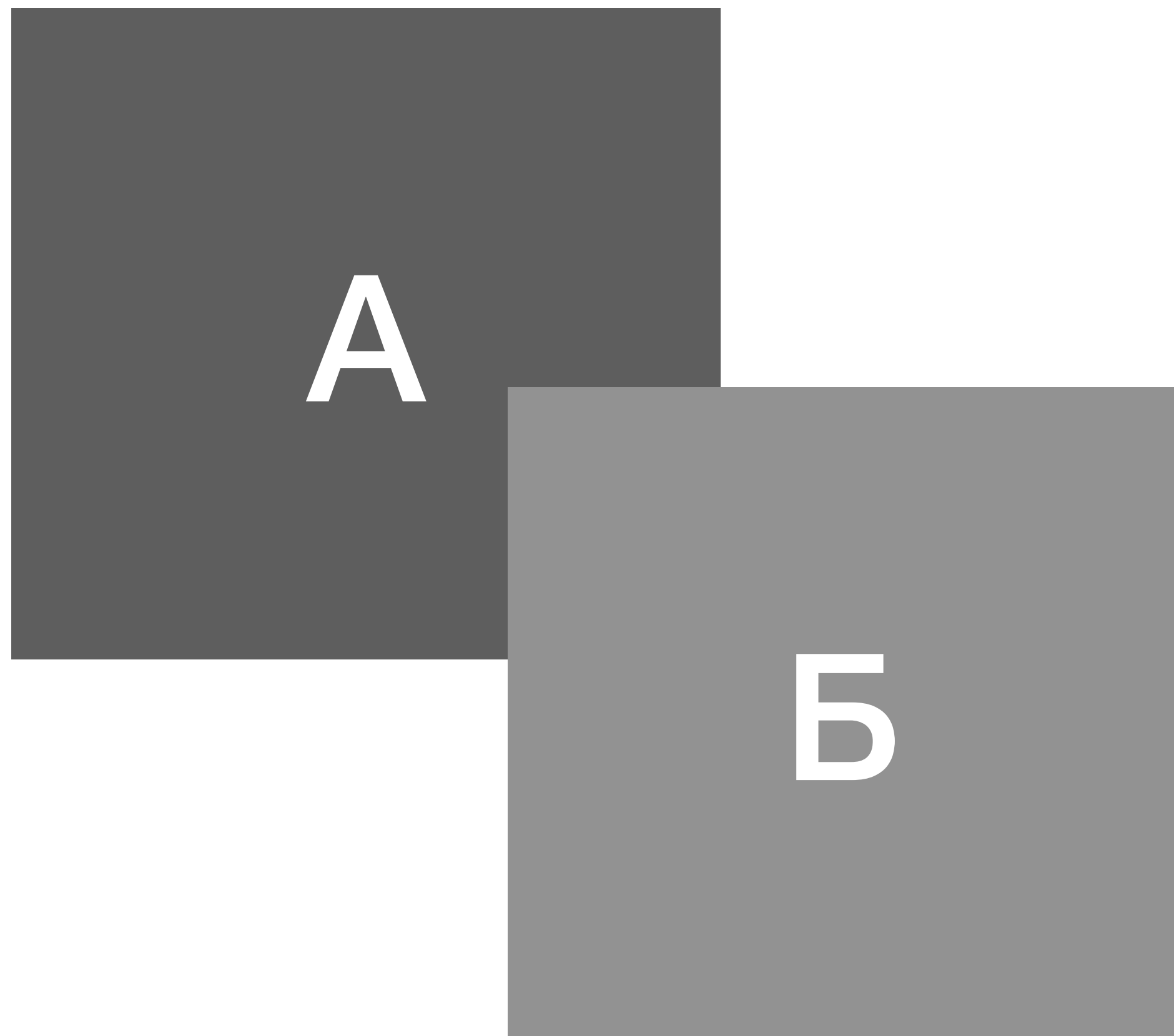
В рамках монолита



В рамках монолита



В рамках монолита



```
newAppCookie ?  
  <NewApp /> :  
  <OldApp />
```

В рамках монолита — плюсы

- Постранично, покомпонентно
- Минимальные вложения в инфраструктуру
- Переиспользование старого кода

В рамках монолита — минусы

- Циклическая связь старого кода от нового и наоборот
- Deadlock зависимостей (webpack4 / webpack5)
- Один рантайм. Упал старый — упадет и новый
- Очень сложно обновить базовую технологию (React, Angular, Vue)

Микрофронт — микро-приложения



Микрофронт — микро-приложения

```
DI.inject({ cmp: "header@1.0.3" })
```

```
DI.inject({ cmp: "body@2.0" })
```

```
DI.inject({  
  cmp: "sidebar"  
})
```

Микрофронт — микро-приложения

```
npm publish header@1.0.2
```

Микрофронт — микро-приложения

`npm publish header@1.0.2`



Verdaccio

Verdaccio Plugin

Микрофронт — микро-приложения

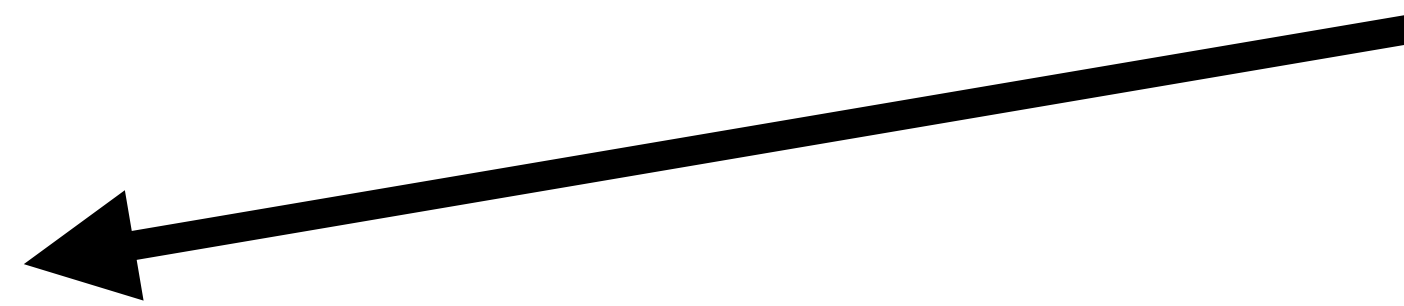
`npm publish header@1.0.2`



Verdaccio

Verdaccio Plugin

Build + SSR deploy



Микрофронт — микро-приложения

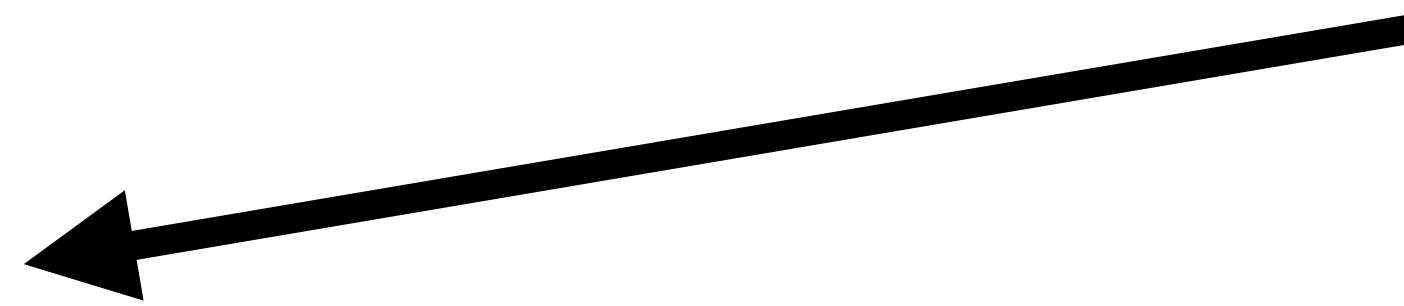
npm publish header@1.0.2



Verdaccio

Verdaccio Plugin

Build + SSR deploy



```
{
```

```
"name": "header",
```

```
"version": "1.0.2",
```

```
"client": ["https://cdn.com/bundle-a.hash.js"],
```

```
"server": "https://server.com/v/1.0.2/bundle-a"
```

```
}
```

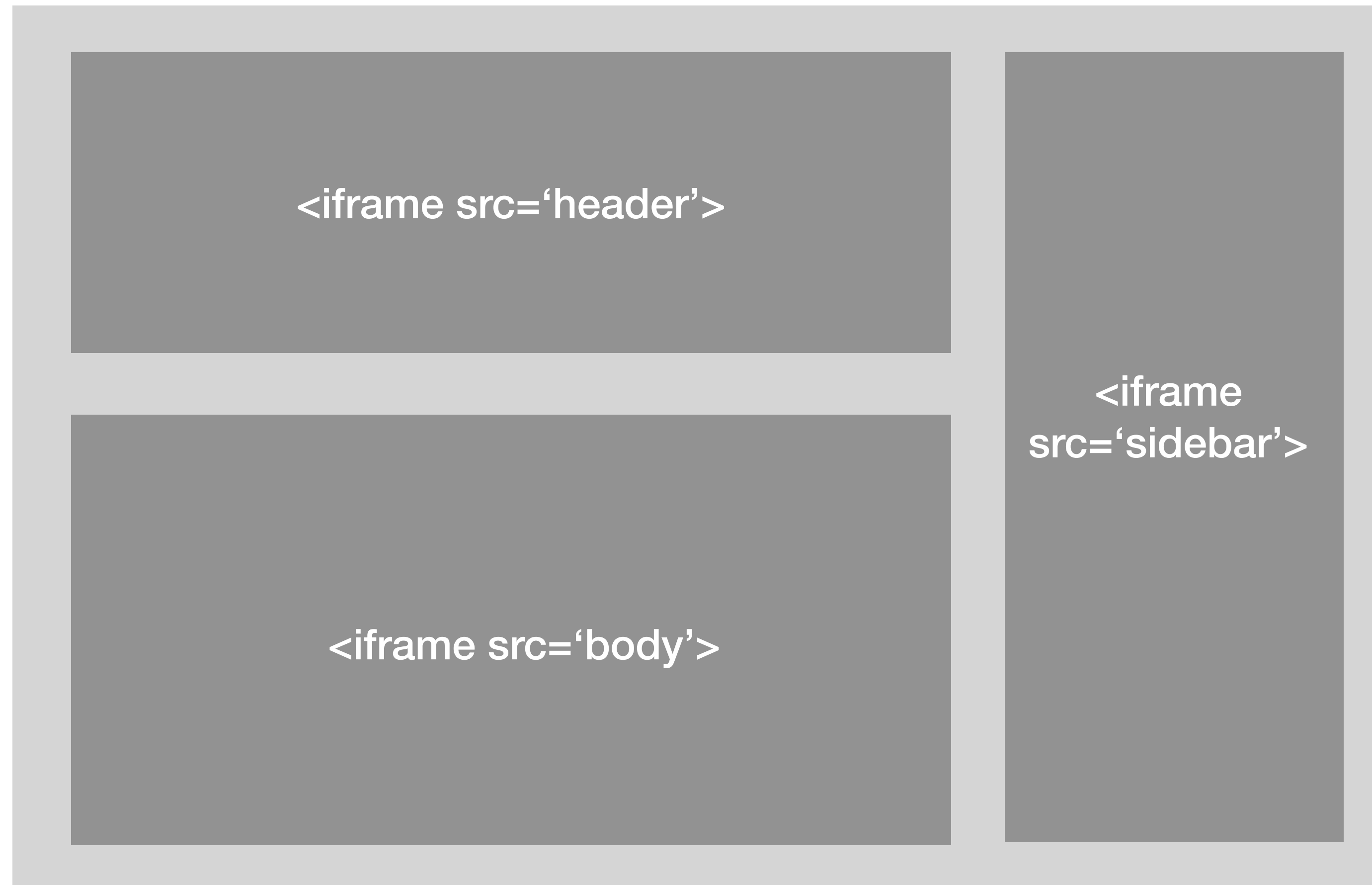
Микрофронт — микро-приложения

```
DI.inject({ cmp: "header@1.0.3" })
```

```
DI.inject({ cmp: "body@2.0" })
```

```
DI.inject({  
  cmp: "sidebar"  
})
```

Микрофронт — iframe



Микрофронт — плюсы

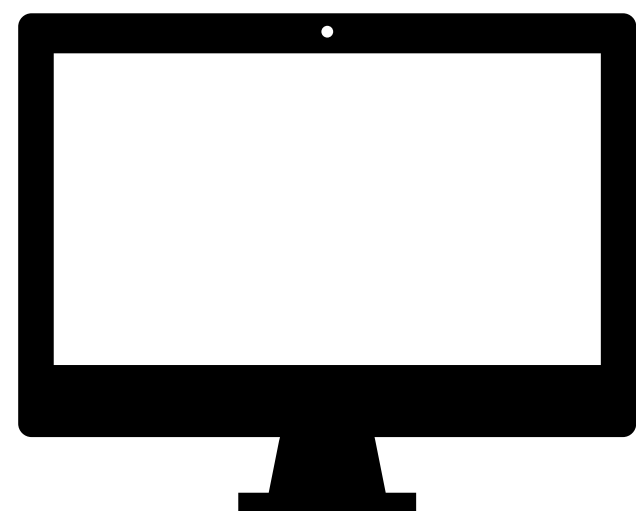
- В основном покомпонентно, но можно и постранично (iframe)
- Полная свобода, за исключением общей шины

Микрофронт — минусы

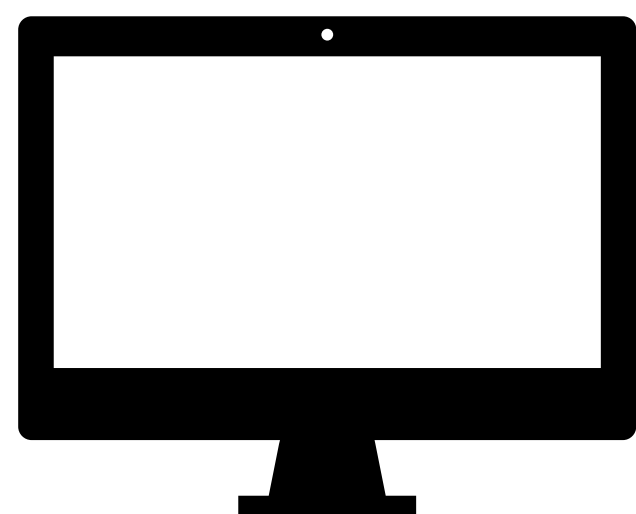
- “Осознанное усложнение архитектуры” (с)
- Существенные вложения в инфраструктуру
- Скорость приложения, дублирование зависимостей (Module Federation тут не спасет)

За одним Проху

За одним Proxy

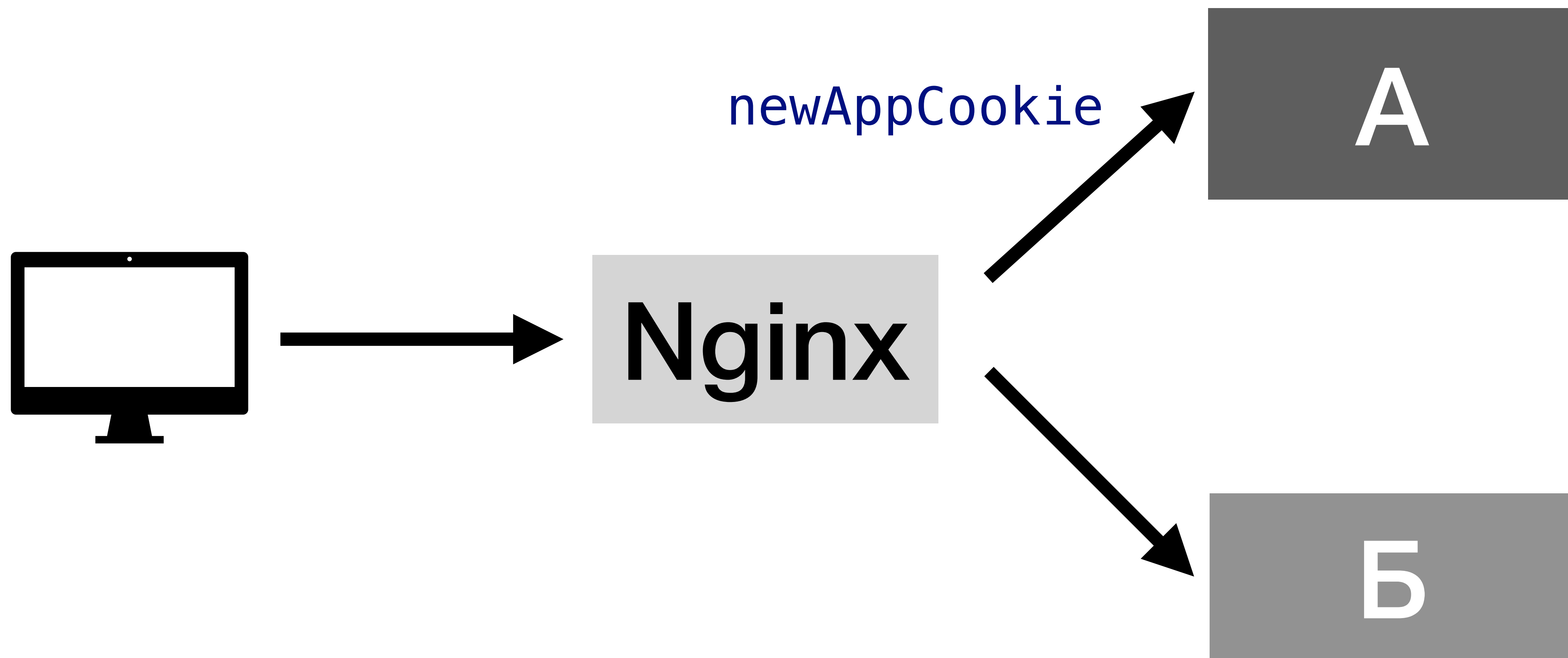


За одним Проху



Nginx

За одним Проху



За одним Proxu — плюсы

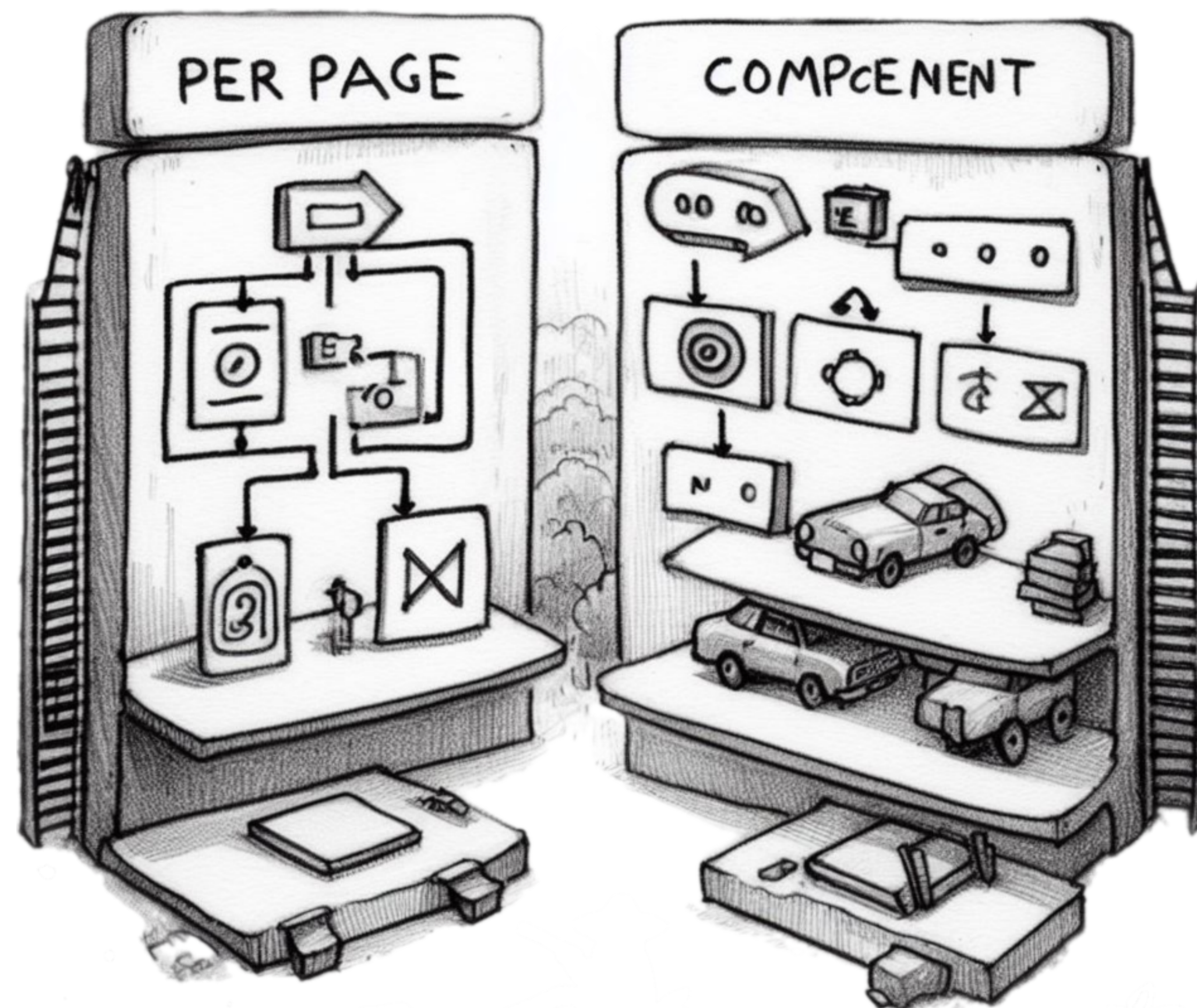
- Полная свобода
- Возможность создать Fallback на старый UI

За одним Proxu — минусы

- Только постраничный вариант переезда
- Нужен отдельный CI
- Более сложная клиент-серверная архитектура (L7, L3)
- Проблемы с A/V и 404



Миграция в рамках монолита



Проблемы покомпонентного подхода

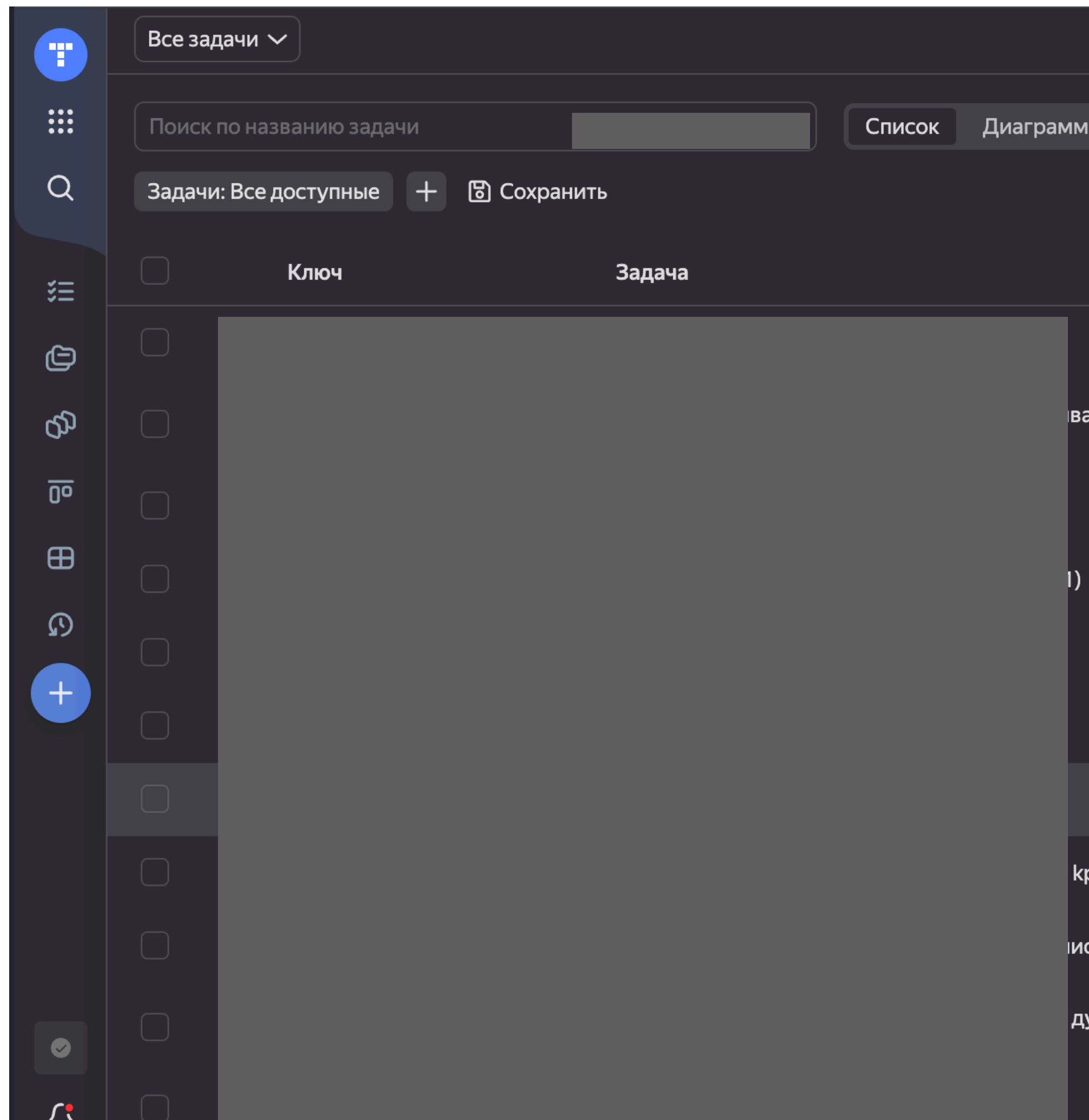
- Долго
- Сложно поменять архитектуру

Постраничный подход

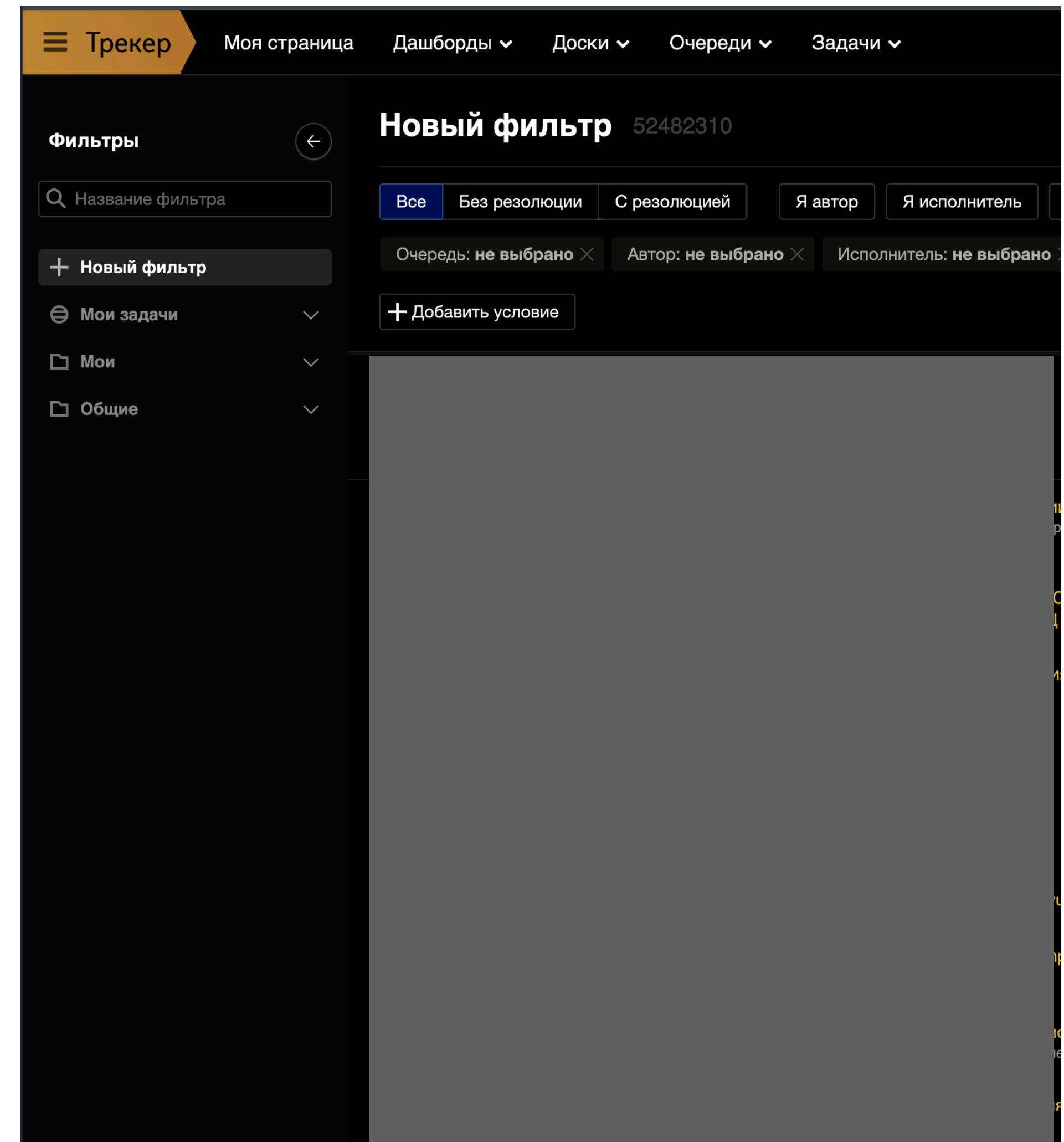
Проблемы



Новый UI



Старый UI



Сложно обновлять и то, и то

**Зависимости
(webpack 4 / webpack 5)**

Подпапка LEGACY

Очень много переписывать

Рефакторинг



Рефракторинг — это...



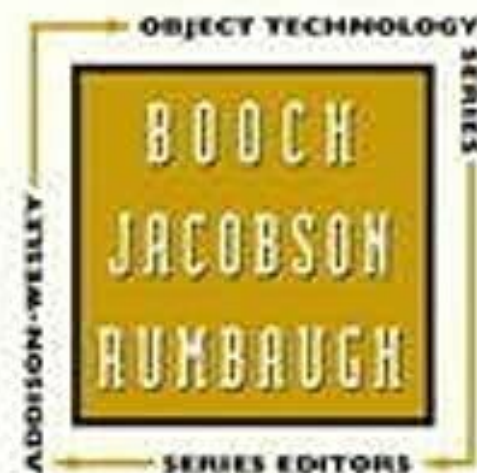
REFACTORING

IMPROVING THE DESIGN
OF EXISTING CODE

MARTIN FOWLER

With contributions by **Kent Beck, John Brant,
William Opdyke, and Don Roberts**

Foreword by **Erich Gamma**
Object Technology International, Inc.



The Addison-Wesley Signature Series



REFACTORING TO PATTERNS

JOSHUA KERIEVSKY

Forewords by *Ralph Johnson and Martin Fowler*
Afterword by *John Brant and Don Roberts*



- Находим CodeSmell 🍌
- Используем стандартные методы рефакторинга

Code Smell - Bloaters

Refactoring - Extract Component

```
EXPLORER  ...  <> app.component.html X  [Icons]

NG-TEST-FILE-UPLOAD
├── dist
├── e2e
├── node_modules
├── src
│   └── app
│       ├── # app.component.css
│       ├── <> app.component.html
│       ├── TS app.component.ts
│       ├── TS app.component.spec.ts
│       ├── TS app.module.ts
│       ├── TS file-upload.service.ts
│       ├── > assets
│       ├── > environments
│       ├── ★ favicon.ico
│       ├── <> index.html
│       └── TS main.ts
└── ...

src > app > <> app.component.html > [Icon] h2
  You, 13 hours ago | 1 author (You)
  1  <!--The content below is only a placeholder and can be replaced.-->
  2  <div style="text-align:center">
  3      <h1>
  4          Welcome to {{ title }}!
  5      </h1>
  6      
 13     <li>
 14         <h2><a target="_blank" rel="noopener" href="https://angular.io/cli">CLI
 15     </li>
 16     <li>
 17         <h2><a target="_blank" rel="noopener" href="https://blog.angular.io/">An
 18     </li>
 19 </ul>
```

Refactoring - Extract Component

- VSCode - [obenjiro/vscode-arr](https://github.com/obenjiro/vscode-arr)
- WebStorm - из коробки

Паттерн — это...

```
class ВойтиВАйтиАдаптер {  
    пройтиКурсы() {  
        ...  
    }  
    устроитьсяНаРаботу() {  
        ...  
    }  
    ...  
}
```



```
class ВойтиВАйТиАдаптер {
    пройтиКурсы() {
        this.читатьКниги();
        this.пробоватьКодить();
        this.бытьМенти();
        this.ходитьНаМитапыИКонференции();
        this.читатьДокументацию();
    }
    устроитьсяНаРаботу() {
        this.подготовитьсяКСобеседованию();
        this.раздобытьТриГодаОпытаРаботы();
        this.знатьБазыДанных();
        this.знатьБек();
        this.уметьБутстрапитьПроект();
    }
    ...
}
```




react-beautiful-dnd (rbd)

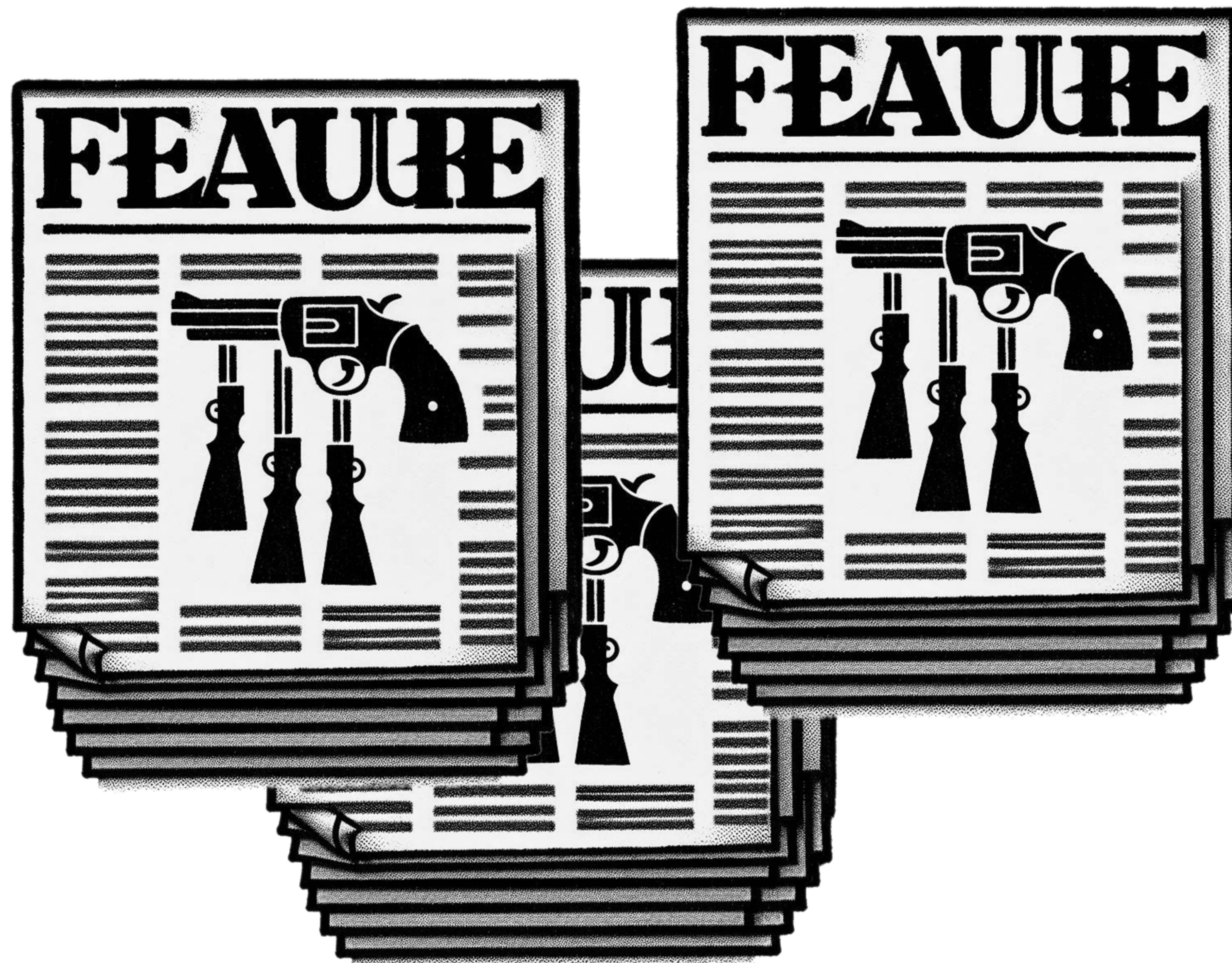
! Maintenance & support

This library continues to be relied upon heavily by Atlassian products, but we are focused on other priorities right now and have no current plans for further feature development or improvements.

It will continue to be here on GitHub and we will still make critical updates (e.g. security fixes, if any) as required, but will not be actively monitoring or replying to issues and pull requests.

We recommend that you don't raise issues or pull requests, as they will not be reviewed or actioned until further notice.

Большое кол-во файлов



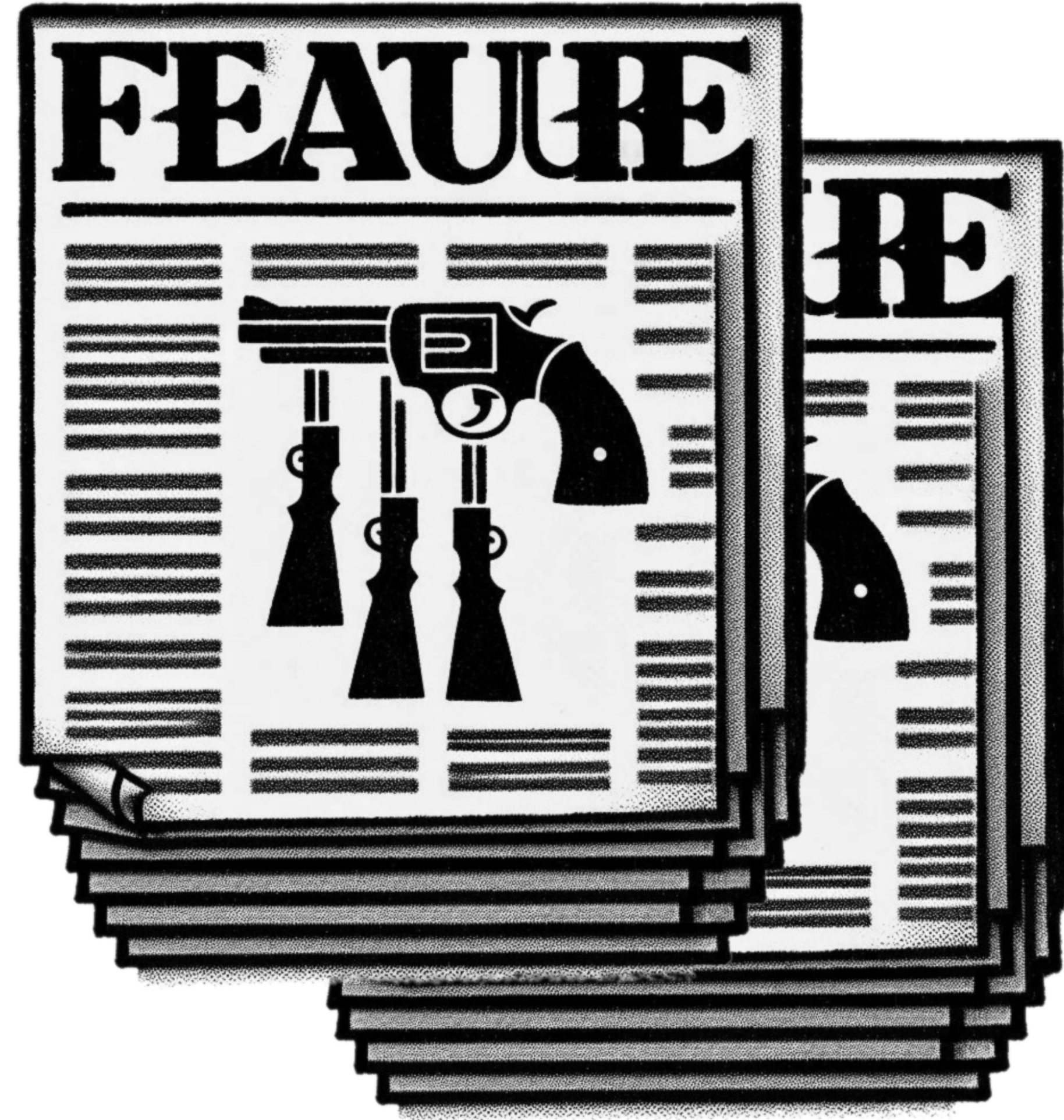
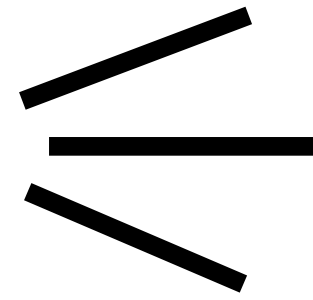


Рефакторинг -> паттерн



- Находим CodeSmell 🍌
- Используем стандартные методы рефакторинга
- Рефакторим в паттерн
- Используем стандартные методы, чтобы вернуть все обратно

CodeSmell - Shotgun Surgery



Стандартные методы рефакторинга

- **Extract Class**
- **Extract Method**
- **Extract Component**

DndAdapter.tsx

```
import type { DragDropContextProps, DraggableProps, DroppableProps } from 'react-beautiful-dnd';  
import { Droppable as Droppable2 } from 'react-beautiful-dnd';  
import { Draggable as Draggable2 } from 'react-beautiful-dnd';  
import { DragDropContext as DragDropContext2 } from 'react-beautiful-dnd';
```

DndAdapter.tsx

```
import type { DragDropContextProps, DraggableProps, DroppableProps } from 'react-beautiful-dnd';
import { Droppable as Droppable2 } from 'react-beautiful-dnd';
import { Draggable as Draggable2 } from 'react-beautiful-dnd';
import { DragDropContext as DragDropContext2 } from 'react-beautiful-dnd';

export const DragDropContext = (props: DragDropContextProps) => {
  return <DragDropContext2 {...props} />;
};

export const Droppable = (props: DroppableProps) => {
  return <Droppable2 {...props} />;
};

export const Draggable = (props: DraggableProps) => {
  return <Draggable2 {...props} />;
};
```

DndAdapter.tsx

```
export const DragDropContext = (props: DragDropContextProps) => {  
  return <DragDropContext2 {...props} />;  
};
```

DndAdapter.tsx

```
export const DragDropContext = (props: {}) => {  
  return <DragDropContext2 {...props} />;  
};
```

DndAdapter.tsx

```
export const DragDropContext = (props: {
  children: ReactNode;
  onDragEnd: ((result: DropResult) => void) | ((result: DropResult) => Promise<void>);
  onBeforeDragStart?: (dragStart: DragStart) => void;
  onDragStart?: (dragStart: DragStart) => void;
}) => {
  return <DragDropContext2 {...props} />;
};
```

DndAdapter.tsx

```
export const DragDropContext = (props: {
  children: ReactNode;
  onDragEnd: ((result: DropResult) => void) | ((result: DropResult) => Promise<void>);
  onBeforeDragStart?: (dragStart: DragStart) => void;
  onDragStart?: (dragStart: DragStart) => void;
}) => {
  return <DragDropContext2 {...props} />;
};
```


DndAdapter.tsx

```
export const DragDropContext = (props: {
  children: ReactNode;
  onDragEnd: ((result: DropResult) => void) | ((result: DropResult) => Promise<void>);
  onBeforeDragStart?: (dragStart: DragStart) => void;
  onDragStart?: (dragStart: DragStart) => void;
}) => {
  return <DragDropContext2 {...props} />;
};
```

Стандартные методы рефакторинга

- **Inline Class**
- **Inline Method**
- **Inline Component**

Все файлы -> Adapter -> Все файлы

Каждый шаг можно проверить

STCAIMONPY / COMDRBCY PEELCP

SEMGREP/
COMBY

ELCITAM

1100

```
import React, { Component } from 'react';

class Welcome extends Component {
  state = {
    name: 'Visitor'
  };

  render() {
    return <h1>Welcome, {this.state.name}!</h1>;
  }
}

export default Welcome;
```



```
import React, { useState } from 'react';

const Welcome = () => {
  const [name, setName] = useState('Visitor');

  return <h1>Welcome, {name}!</h1>;
}

export default Welcome;
```

x 200

```
import React, { useState } from 'react';

const Welcome = () => {
  const [name, setName] = useState('Visitor');

  return <h1>Welcome, {name}!</h1>;
}

export default Welcome;
```

Comby

```
comby 'failUnlessEqual(:[a],:[b])' 'assertEqual(:[a],:[b])' example.py
```

```
--- example.py  
+++ example.py  
@@ -1,6 +1,6 @@  
     def test(self):  
         r = self.parse("if 1 fooze", 'r3')  
-         self.failUnlessEqual(  
+         self.assertEqual(  
             r.tree.toStringTree(),  
             '(if 1 fooze)'  
         )
```

```
[rule]
description = "Transform React class component to functional component using hooks"
match = '''
class :[classname] extends Component {
  state = {
    :[statekey]: :[statevalue]
  };

  render() {
    return :[body];
  }
}
'''

rewrite = '''
import React, { useState } from 'react';

function :[classname]() {
  const [:[statekey], set:[statekey]] = useState(:[statevalue]);
  return :[body];
}
'''
```

```
comby -config rewrite.toml  
-in-place -matcher .tsx -d path
```



```
import React, { Component } from 'react';

class Welcome extends Component {
  state = {
    name: 'Visitor'
  };

  render() {
    return <h1>Welcome, {this.state.name}!</h1>;
  }
}

export default Welcome;
```

```
import React, { useState } from 'react';

const Welcome = () => {
  const [name, setName] = useState('Visitor');

  return <h1>Welcome, {name}!</h1>;
}

export default Welcome;
```

Подводные камни

comby-wrapper / comby-server

```
const CW = require('comby-wrapper');
const R = require('rambdax');

async function main() {
  const files = await CW.GetFiles(["./react_app/**/*.js"], {
    ignore: ['**/node_modules/**']
  });

  await R.pipedAsync(
    files,
    // query
    CW.match(`const :[[cname]] = props => {:[hole]}`),
    // here we preparing replacments
    CW.replaceWithSmartIndent(`class :[[cname]] extends React.Component {
      render() {
        const props = this.props;
        :[hole]
      }
    }`),
    // this command is applying all the replacments in original files
    CW.flushWith(CW.flushInPlace)
  )
}
main();
```

**Совет: Сортируйте методы,
импорты и свойства
(eslint)**

Semngrep

```
import React, { Component } from 'react';

class Welcome extends Component {
  state = {
    name: 'Visitor'
  };

  render() {
    return <h1>Welcome, {this.state.name}!</h1>;
  }
}

export default Welcome;
```

rules:

- id: react-class-to-function-hooks

pattern: |

```
class $CLASS extends React.Component {  
  state = { $INIT_STATE };  
  render() {  
    return $BODY;  
  }  
}
```

rewrite: |

```
import React, { useState } from 'react';
```

```
const $CLASS = () => {
```

```
  const [$STATE_KEY, set$STATE_KEY] = useState($INIT_VALUE);
```

```
  return $BODY;
```

```
}
```

```
import React, { Component } from 'react';

class Welcome extends Component {
  state = {
    name: 'Visitor'
  };

  render() {
    return <h1>Welcome, {this.state.name}!</h1>;
  }
}

export default Welcome;
```

```
import React, { useState } from 'react';

const Welcome = () => {
  const [name, setName] = useState('Visitor');

  return <h1>Welcome, {name}!</h1>;
}

export default Welcome;
```




AI to rescue

Супер простой пример
var* -> *let


```
var x = 10;  
function increaseNumber() {  
    var y = x + 1;  
    return y;  
}
```



You

```
var x = 10;
function increaseNumber() {
  var y = x + 1;
  return y;
}
```

Can you refactor this code by replacing all var with let?



ChatGPT

Certainly! Here's the refactored JavaScript code where all instances of `var` have been replaced with `let`:

javascript

Copy code

```
let x = 10;
function increaseNumber() {
  let y = x + 1;
  return y;
}
```



You

```
var x = 10;
function increaseNumber() {
  var y = x + 1;
  return y;
}
```

Can you refactor this code by replacing all var with let?



ChatGPT

Certainly! Here's the refactored JavaScript code where all `var` is replaced with `let`:

javascript

```
let x = 10;
function increaseNumber() {
  let y = x + 1;
  return y;
}
```



**Напиши js скрипт jscodeshift для
замены всех объявлений var на
let в кодовой базе JavaScript**

replaceVarWithLet.js

```
module.exports = function(fileInfo, api, options) {  
  const j = api.jscodeshift;  
  
  return j(fileInfo.source)  
    .find(j.VariableDeclaration)  
    .forEach(path => {  
      if (path.node.kind === 'var') {  
        path.node.kind = 'let';  
      }  
    })  
    .toSource();  
};
```

```
npx jscodeshift -t replaceVarWithLet.js path/to/code/
```



```
var x = 10;  
function increaseNumber() {  
    var y = x + 1;  
    return y;  
}
```

```
let x = 10;  
function increaseNumber() {  
    let y = x + 1;  
    return y;  
}
```

```
let x = 10;  
function increaseNumber(x) {  
  let y = x + 1;  
  return y;  
}
```



Не технические проблемы миграции



ADR

Architecture Desigion Records

Context

Detailed description of context

Options

Option 1. Title of Option 1

Option 2. Title of Option 2

Option 1. Title of Option

Detailed Description of Option 1

Advantages: bla

Disadvantages: bla

Option 2. Title of Option 2

Detailed Description of Option 2

Option 2. Title of Option 2

Detailed Description of Option 2

Advantages: bla

Disadvantages: bla

Decision

Our desigion

Reasoning

Why we made that decision

Consequences

bla

ПРИВЫЧКИ



Eslint

/ eslint-plugin-unicorn / eslint-plugin-sonarjs /

Папка SHAME / LEGACY :)

Борьба с велосипедными

Button

 GitHub



view
action

size
xs s m l xl

pin
round-round

selected

disabled

loading

startIcon

endIcon

```
import {Button} from '@gravity-ui/uikit';
```


*.CSS

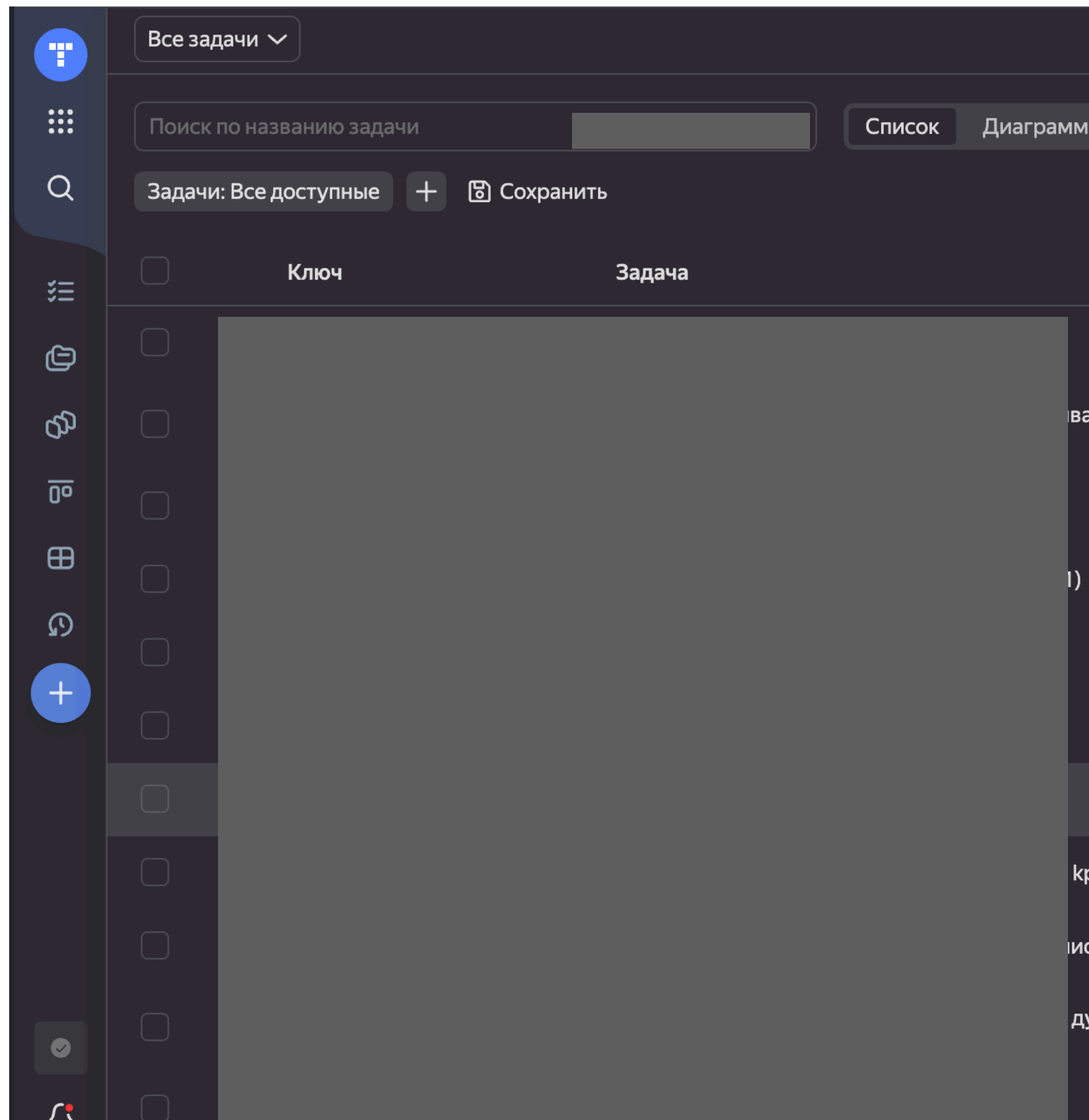
```
var(--g-color-text-primary)  
var(--g-color-base-simple-hover)  
var(--g-color-line-focus)
```

Результат

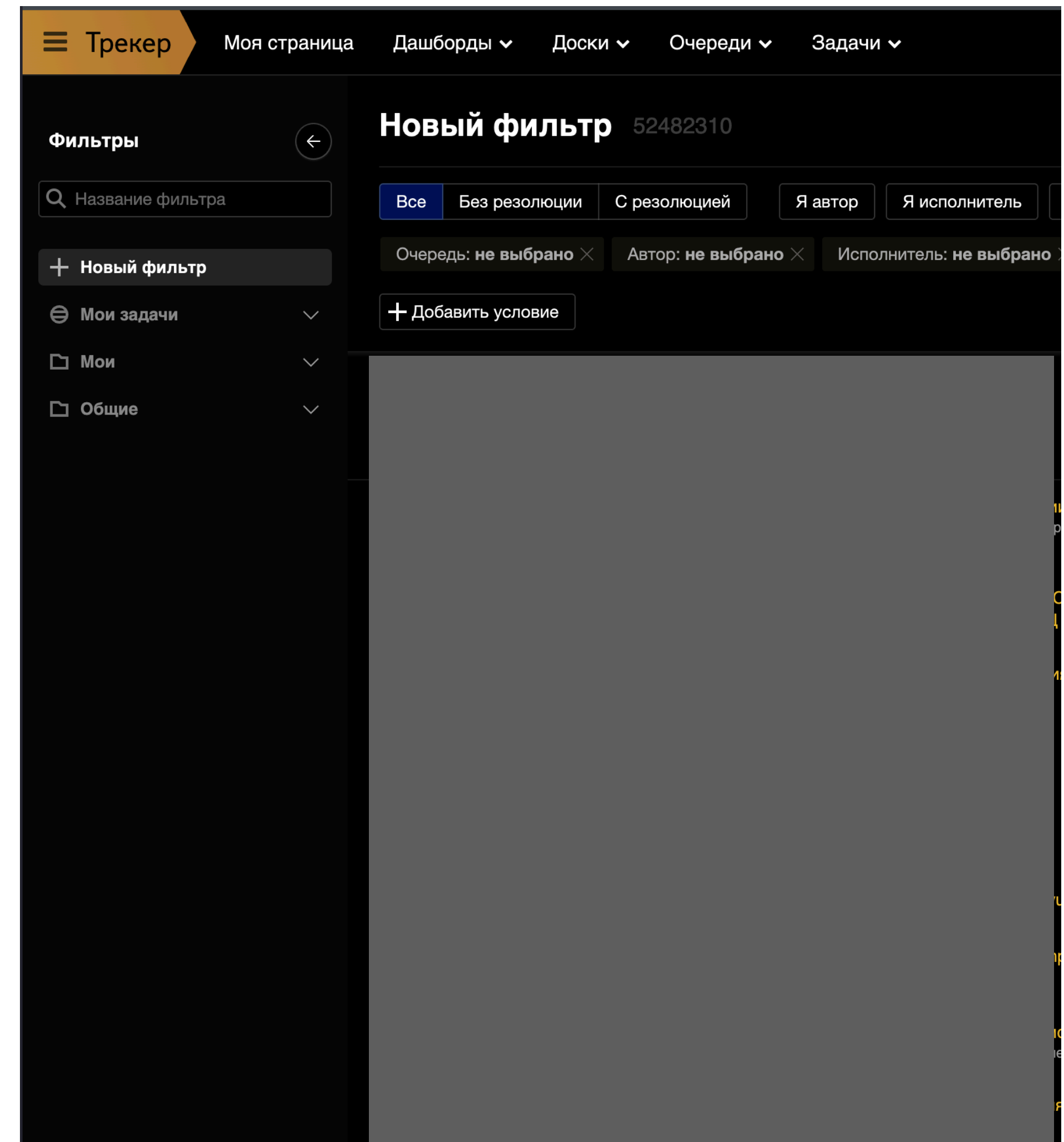
+6101362 -4875502

+165987 -114396

Новый UI



Старый UI



Команда

Выводы

**Все получилось в рамках
монокита**

Стоило идти путем proxy

Сроки

**Миграции, миграции никогда
не меняются**

Спасибо за внимание

Алексей Охрименко

Techlead → Yandex Cloud → Tracker



bit.ly/3wbj2ww