

# **Кровь, пот, микрофронты и монолит**

Денис Колесников

Senior engineer @ Frontend Architecture



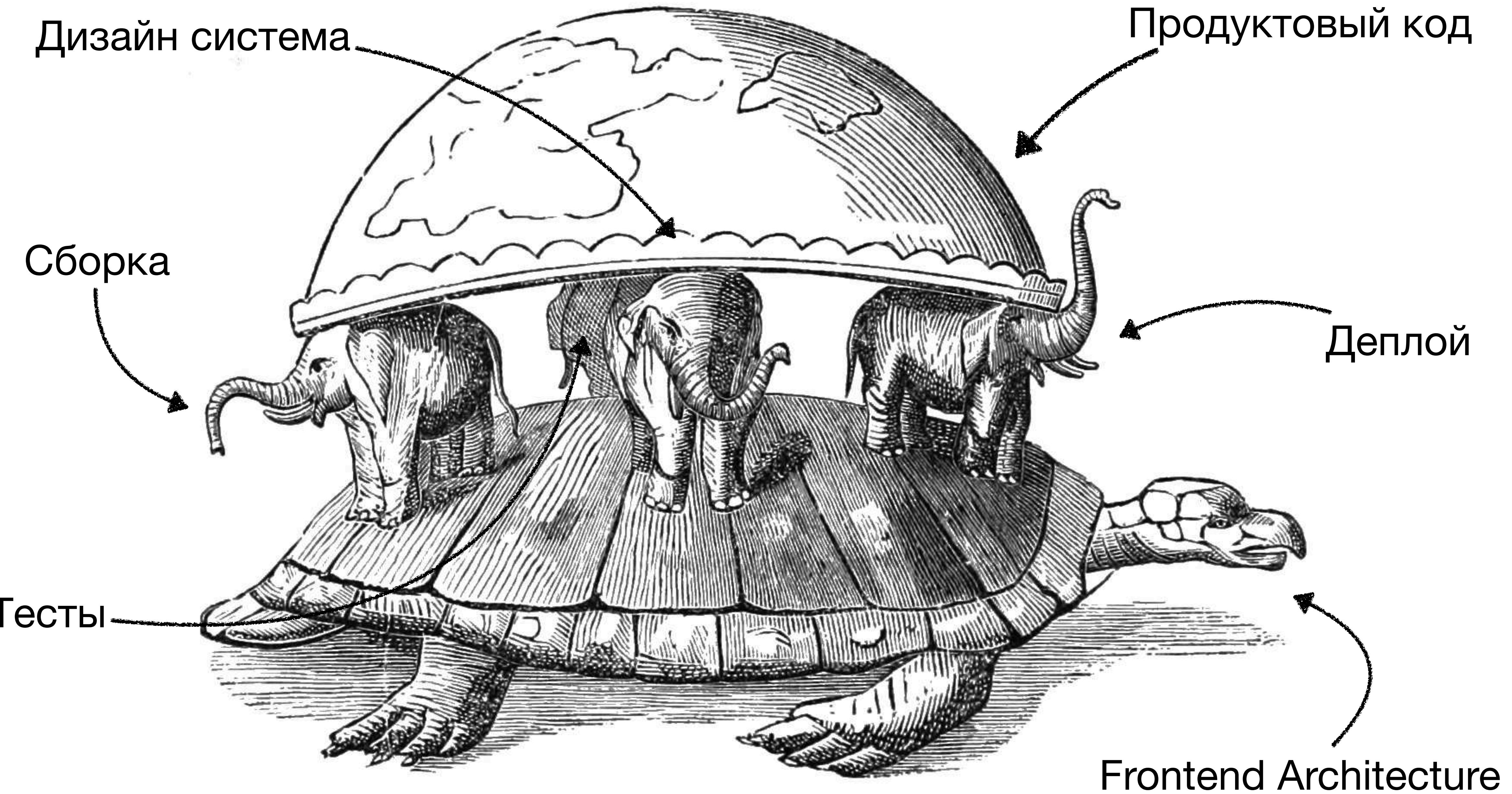
[dekolesnikov@avito.ru](mailto:dekolesnikov@avito.ru)

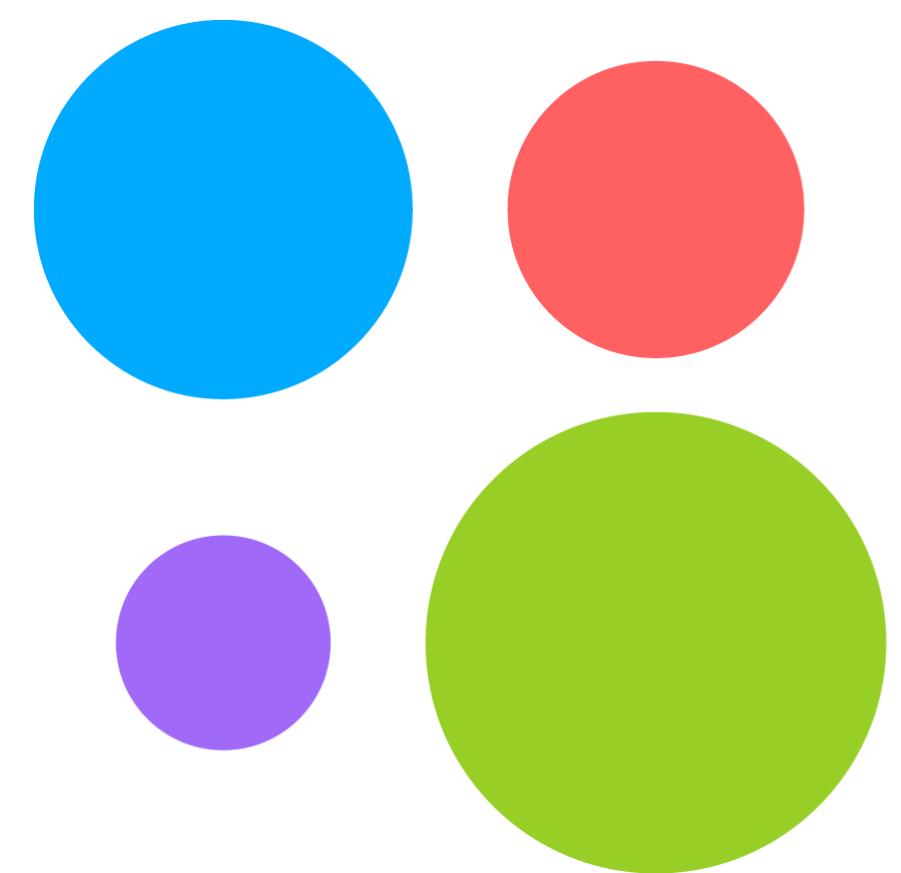
<https://facebook.com/DenKolesnikov>

<https://twitter.com/DenisKolesnikov>

# Frontend Architecture team

- Дизайн платформа
- Архитектура фронтенда
- Инфраструктура
- Инструменты
- Поддержка разработчиков





# Avito



- Самый большой сайт объявлений в России
- Более 20ти команд разработки
- > 500K LoC
- Монолит на PHP \\_(\_\)

# Avito Units



BackEnd



FrontEnd

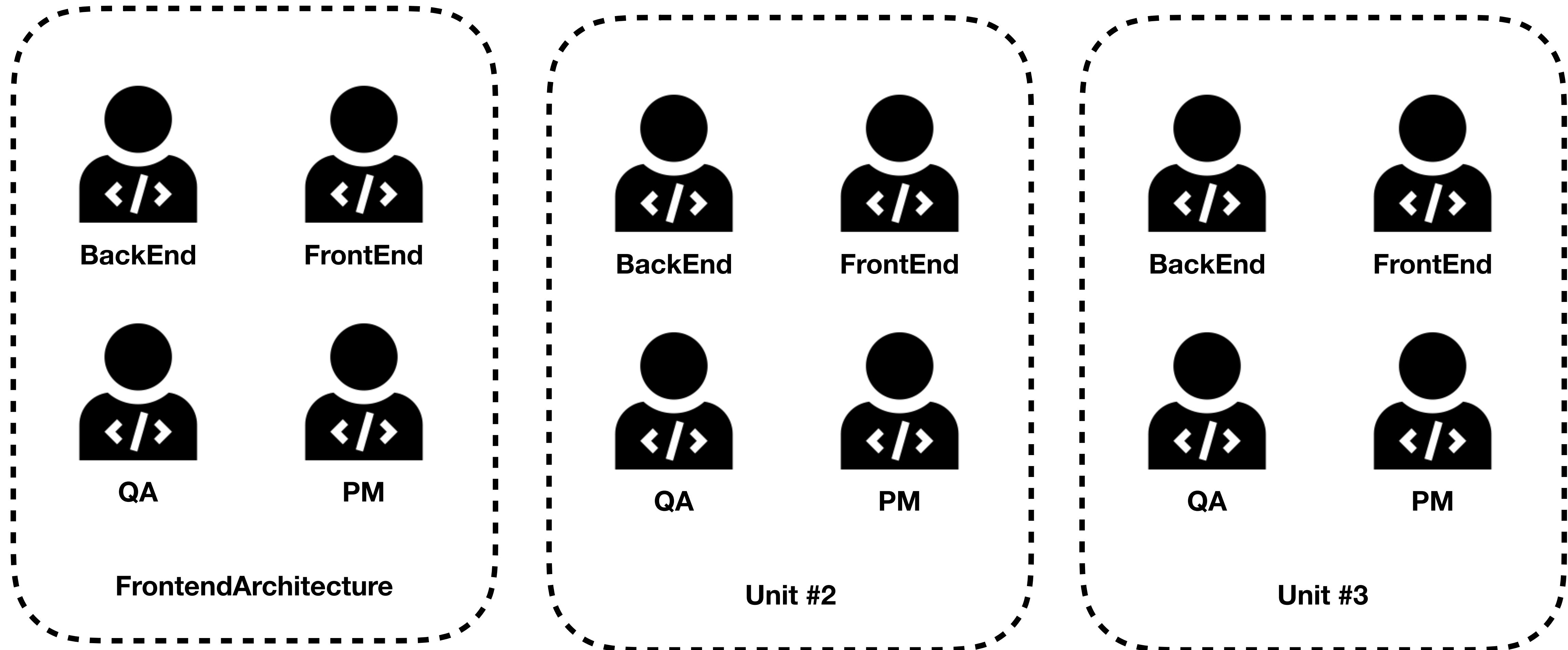


QA



PM

# Avito Units

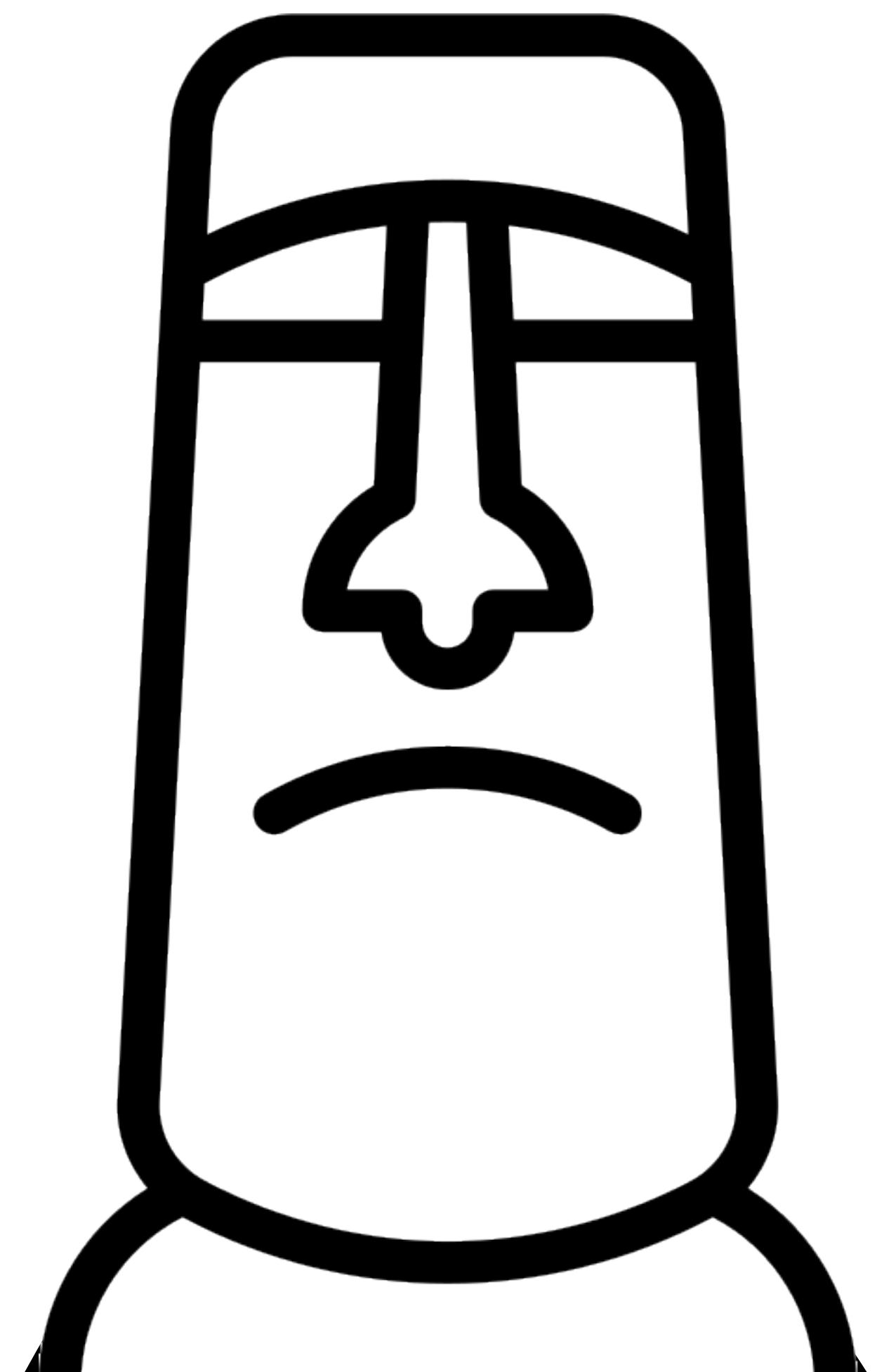


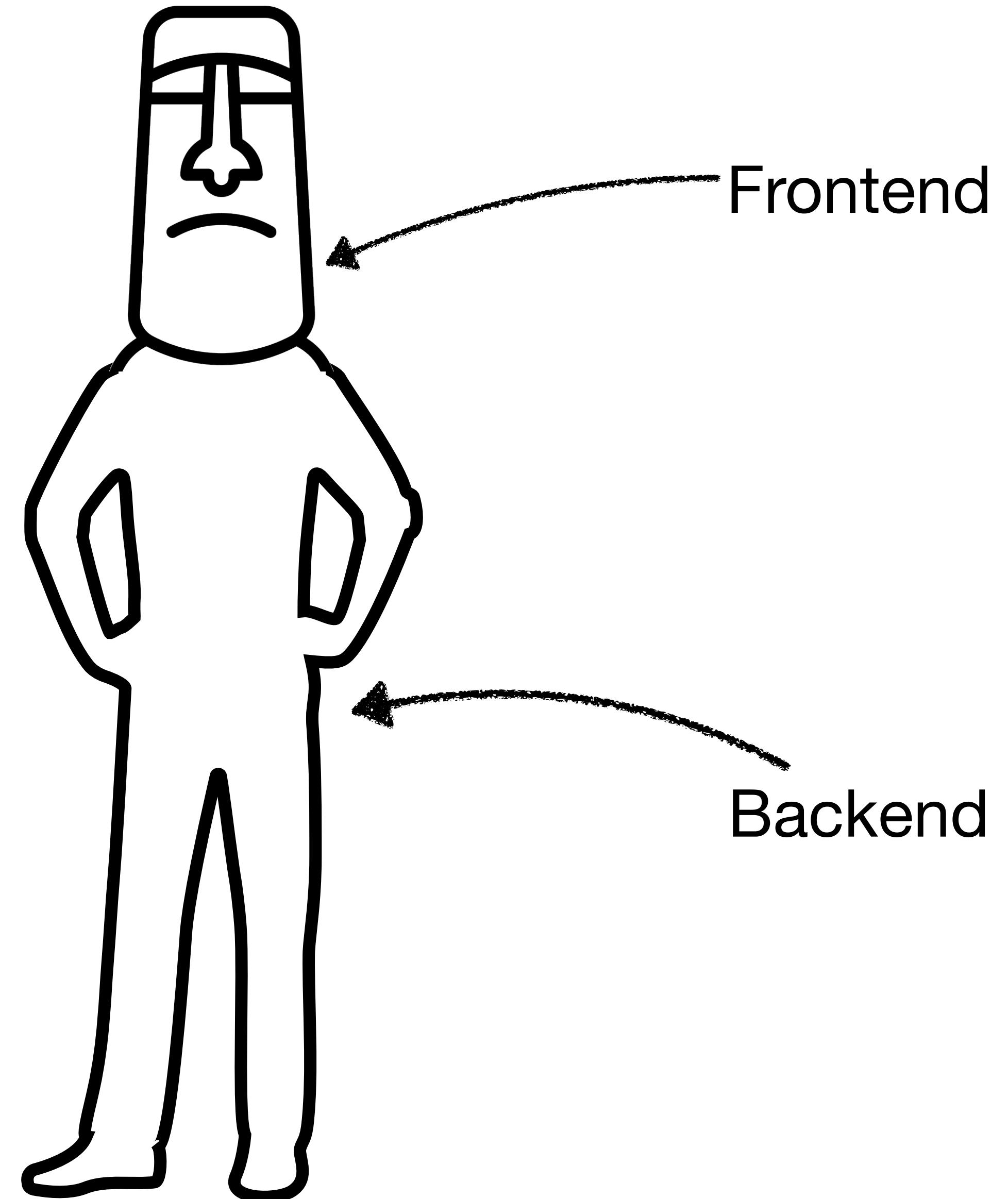
# **Монолит**

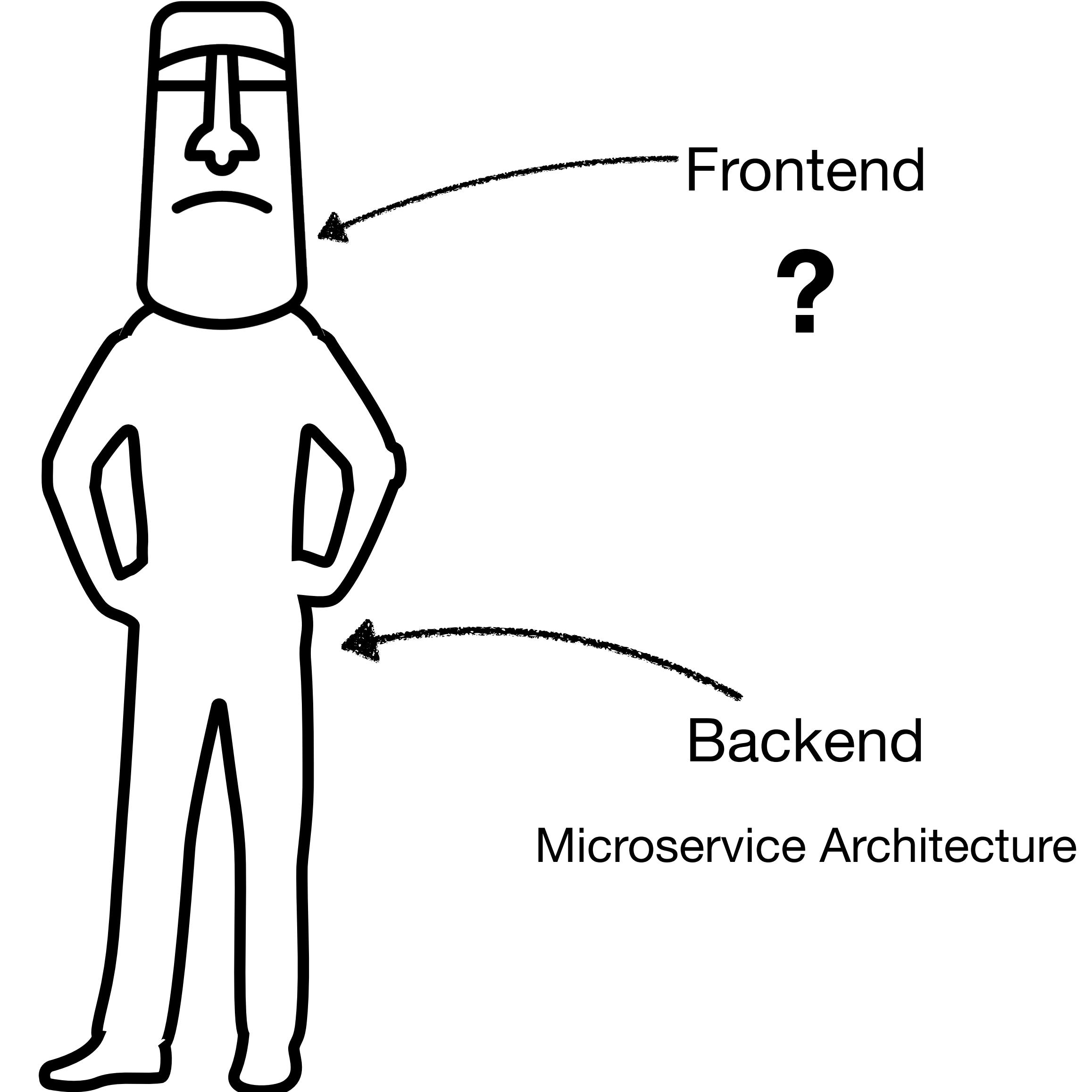
# Монолит

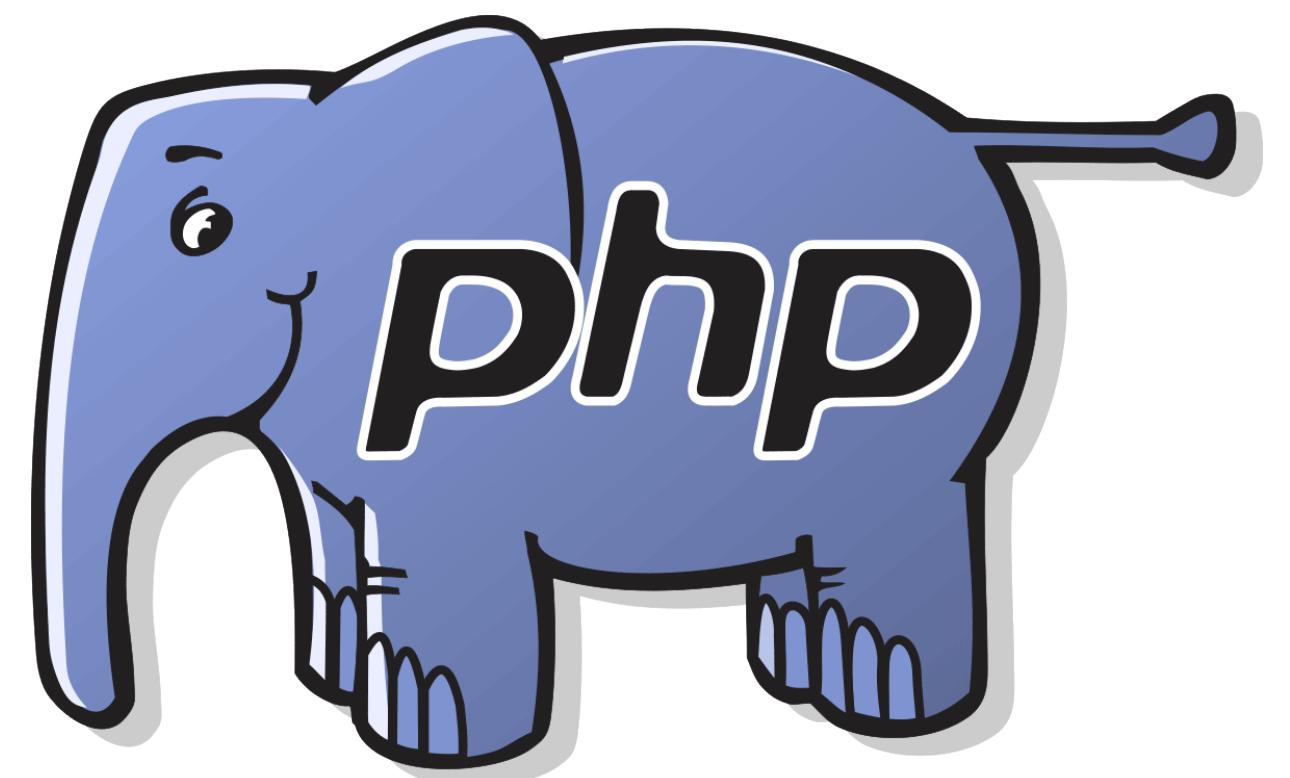
- Высокая связанность кода
- Общий деплой
- Тяжелые окружения для разработки

# Монолит









MVP





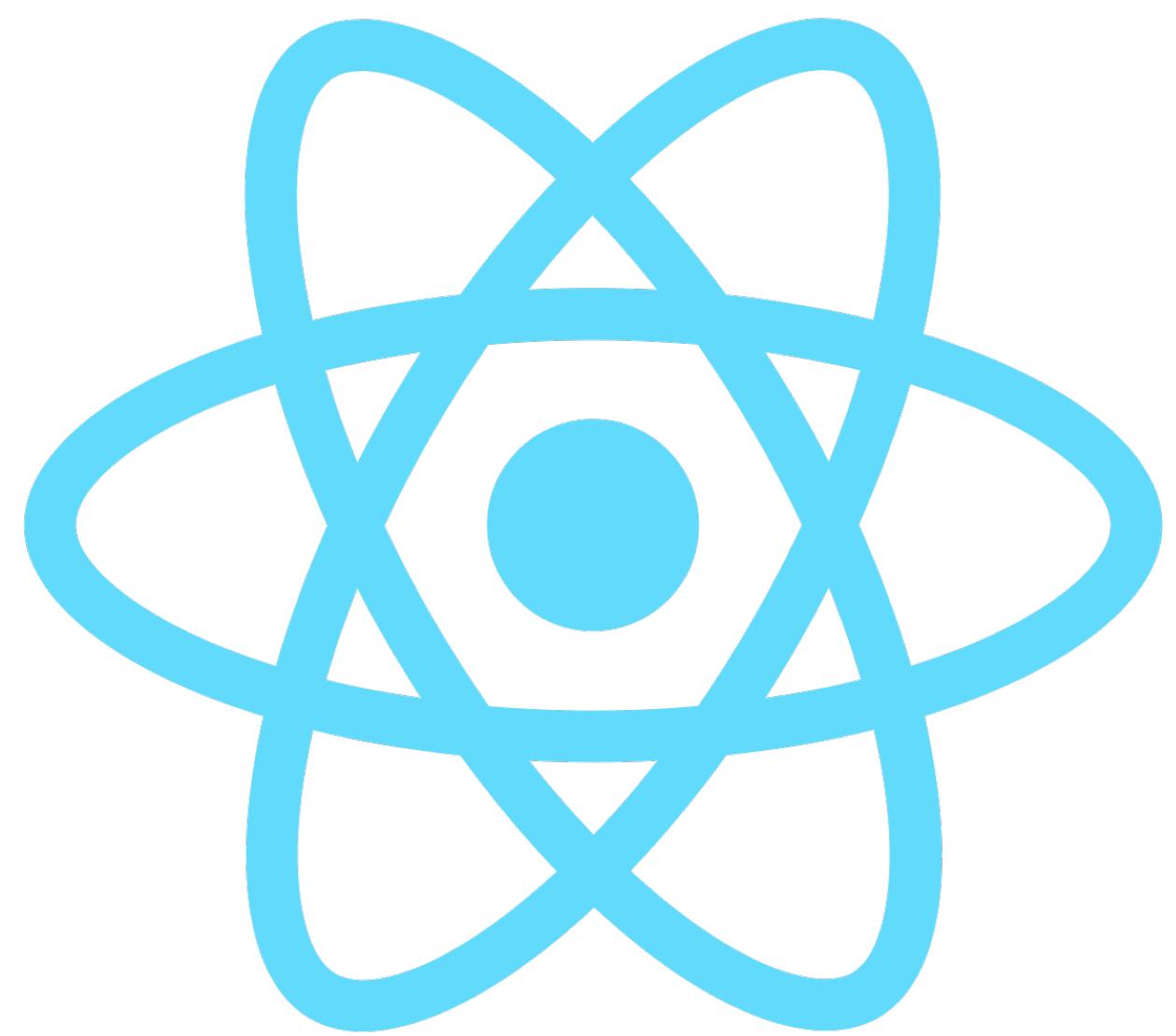
**jQuery**

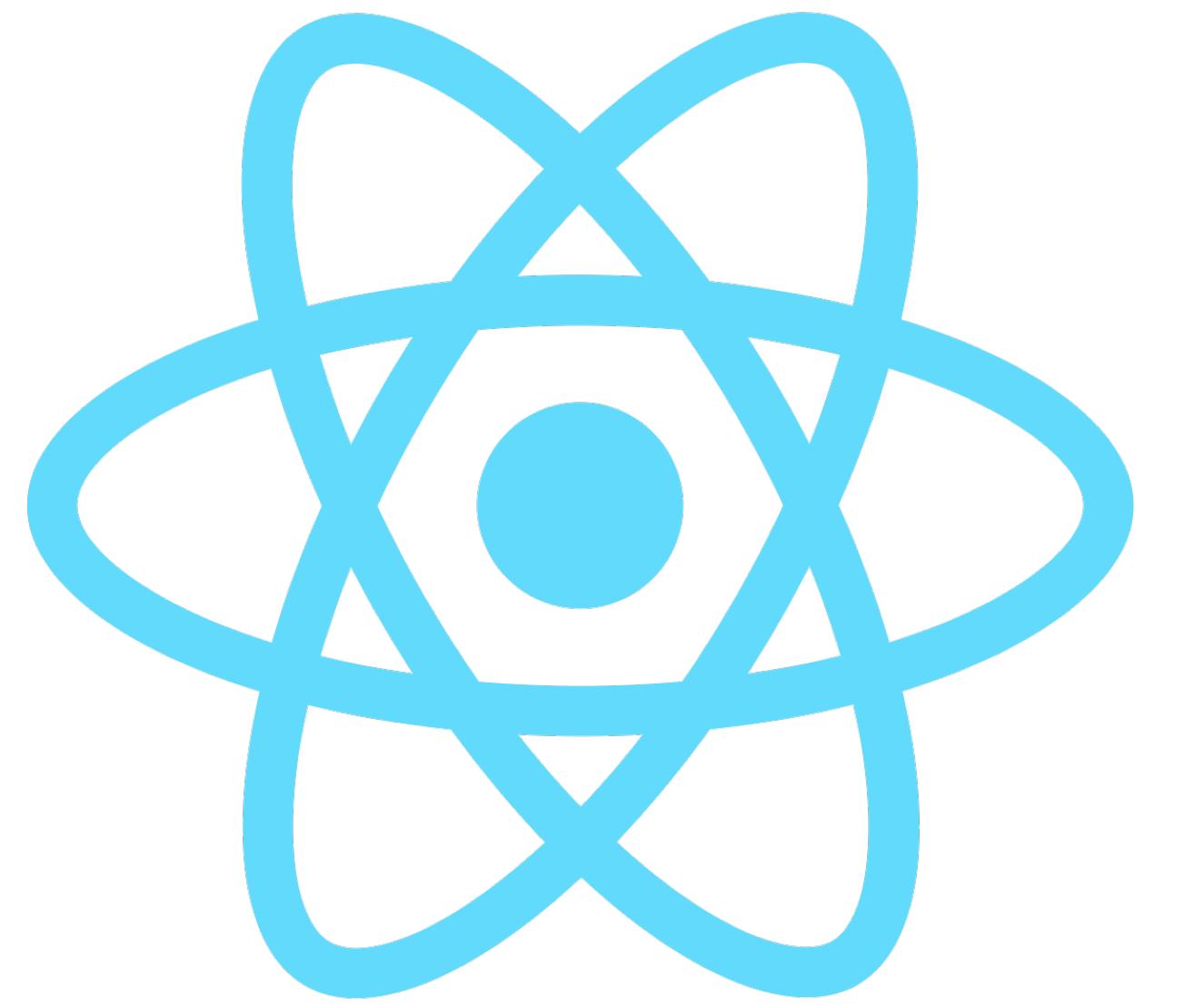


# Миграция технологического стека



PokéMon





Phakt

Phakt, Design System

← → C G | Incognito

Компоненты Примеры Использование Задачи Пакеты Логи изменений Войти

@avito/ui (46)

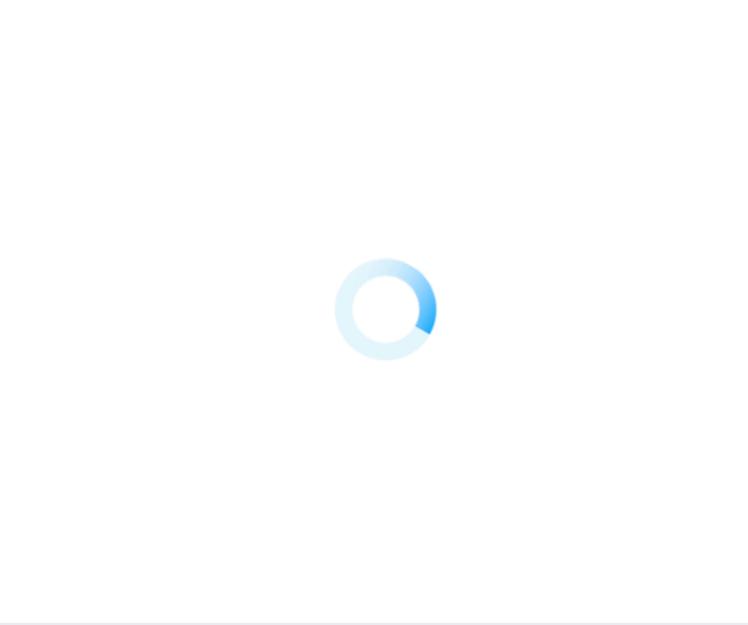
Все теги

Фильтр по имени

- Alert
- Breadcrumbs
- Button
- Captcha
- Carousel
- Checkbox
- CheckboxGroup
- Choice
- Content
- Datepicker
- Dropdown
- DropdownList
- Form
- Grid
- Group
- Heading
- Icon
- Input
- InputMask
- InputNumber
- InputPassword
- Link
- LinkAsButton
- List
- Loader
- Meter
- Notifications
- NumberFormat

## Компоненты

Content **USAGE: 4 / 7** @avito/ui #Лейаут



Datepicker **USAGE: 4 / 6** @avito/ui UI+Kit #Форма

ПН	ВТ	СР	ЧТ	ПТ	СБ	ВС
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

Dropdown **USAGE: 1 / 1** @avito/ui

Dropdown with check

Dropdown with check and multiple, value: null

Dropdown, value: null

DropdownList @avito/ui

Form **USAGE: 15 / 69** @avito/ui

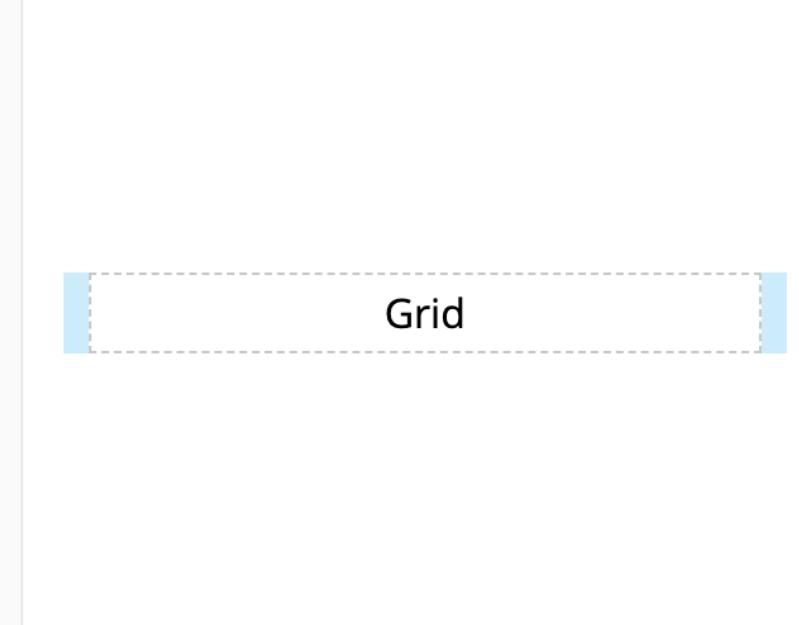
Header-footer Errors Comments Fieldset scheme #Форма #Лейаут

Заголовок формы

Label 1

Label 2

Grid **USAGE: 12 / 99** @avito/ui UI+Kit #Лейаут



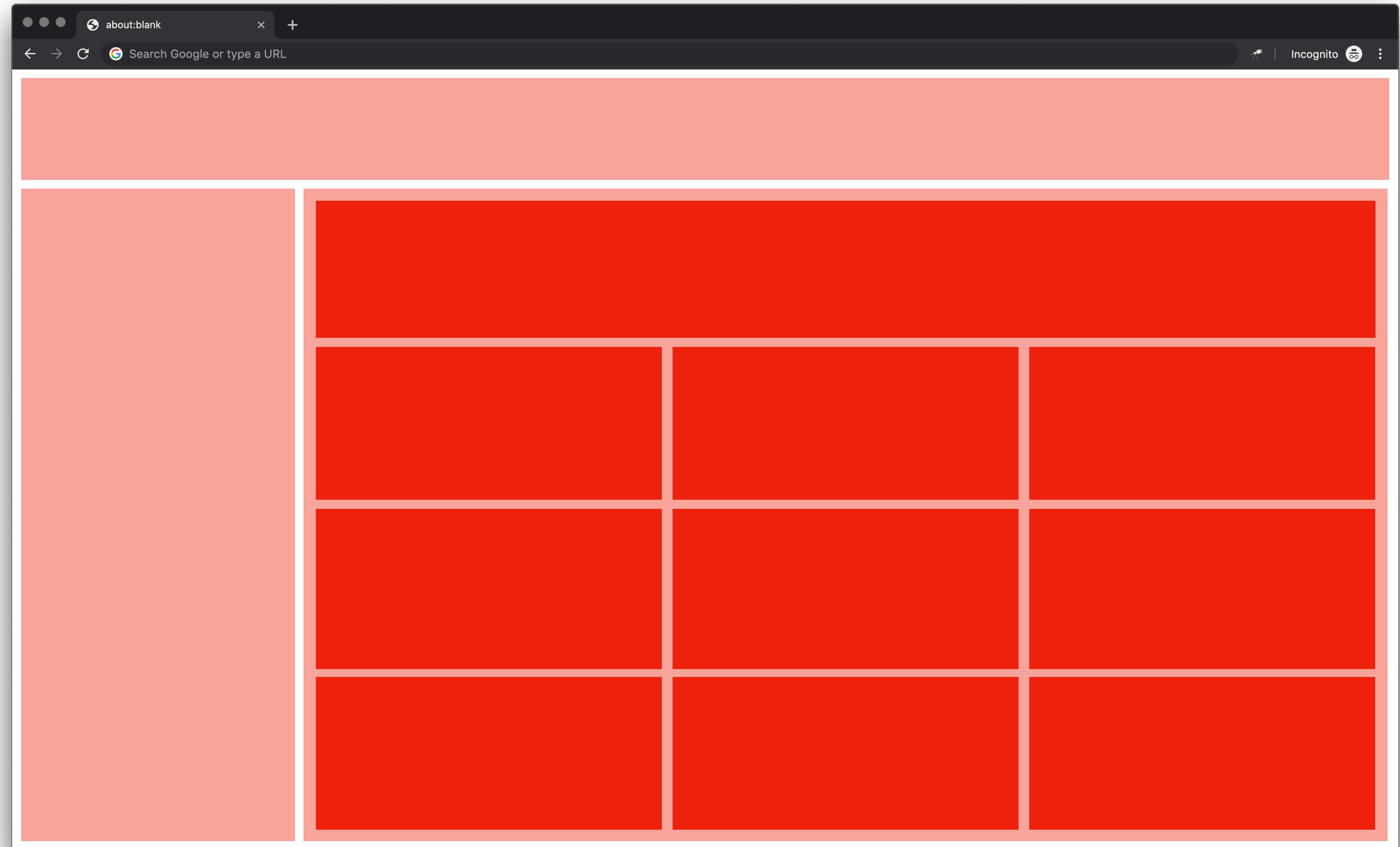
Group **USAGE: 13 / 72** @avito/ui Design #Форма #Лейаут

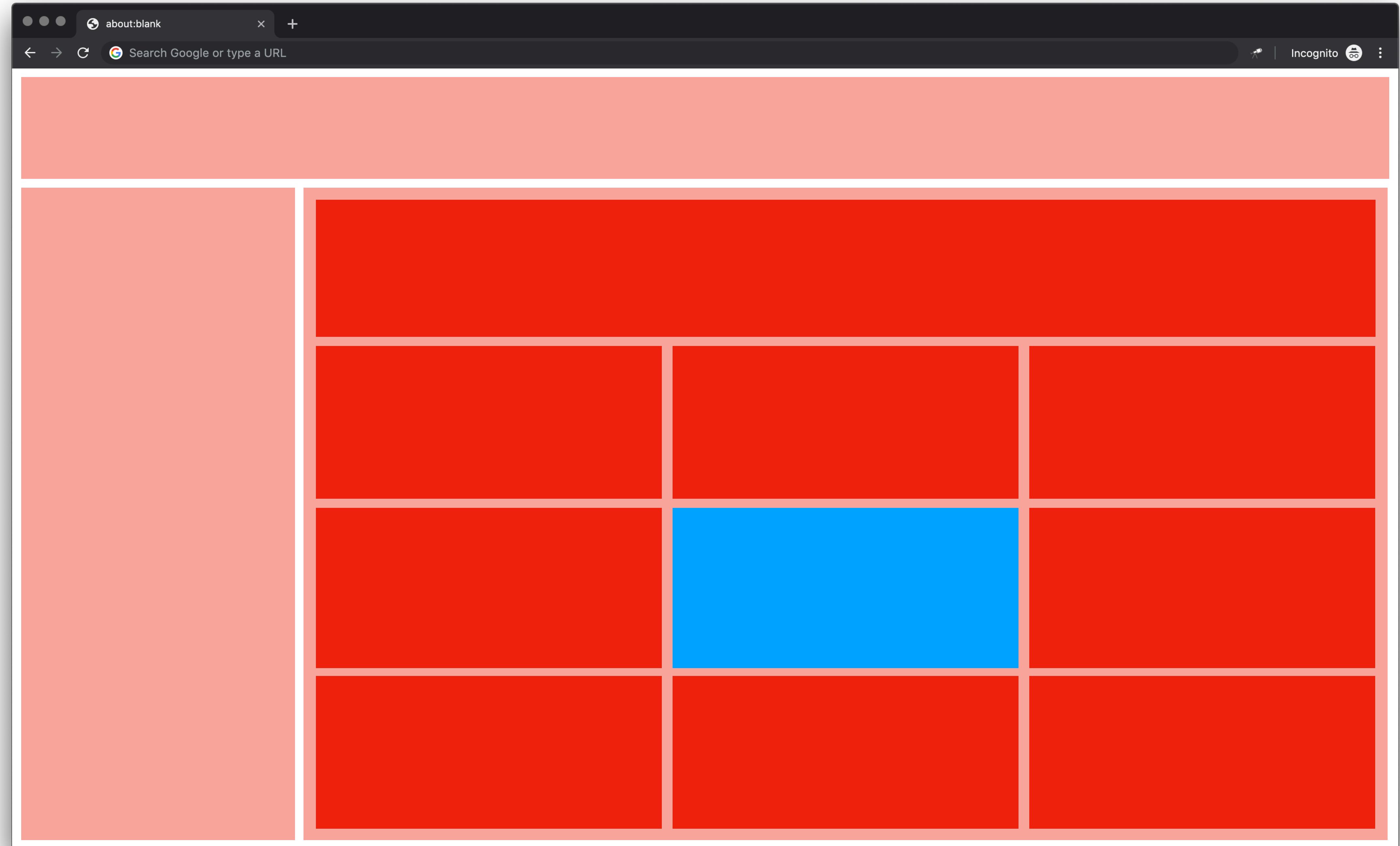
От  До

Heading **USAGE: 9 / 54** @avito/ui Design #Текст

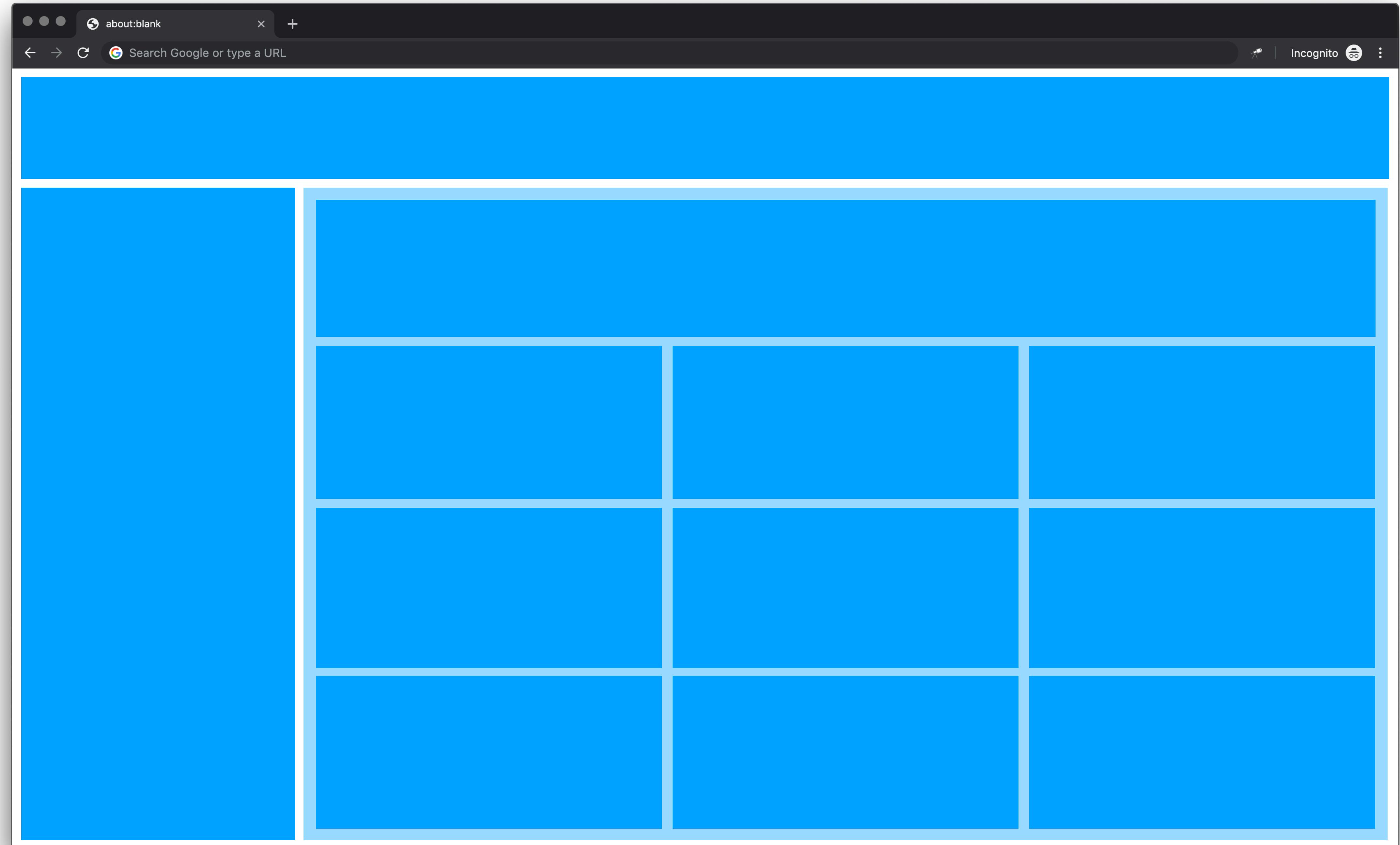
# Искусство медиапланирования

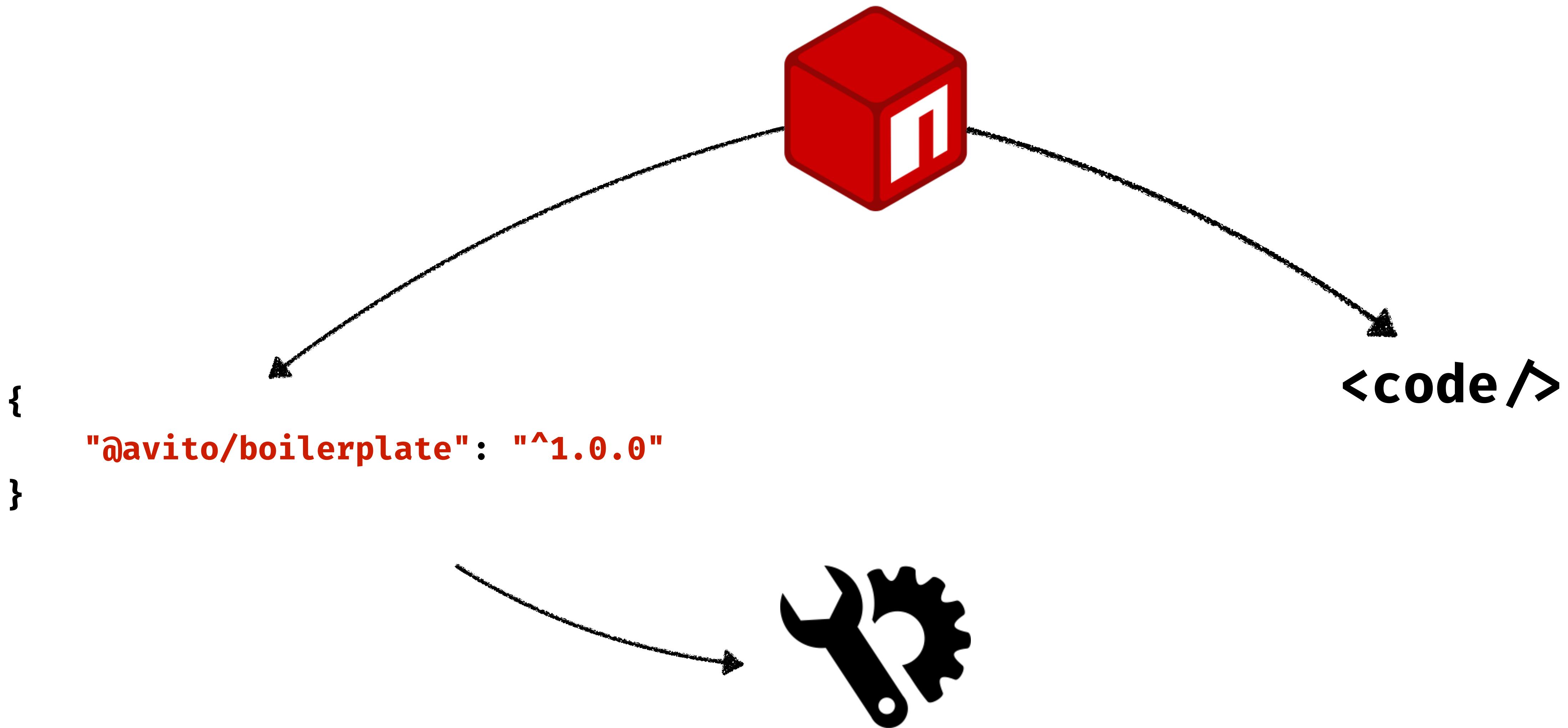
# Growing a seed









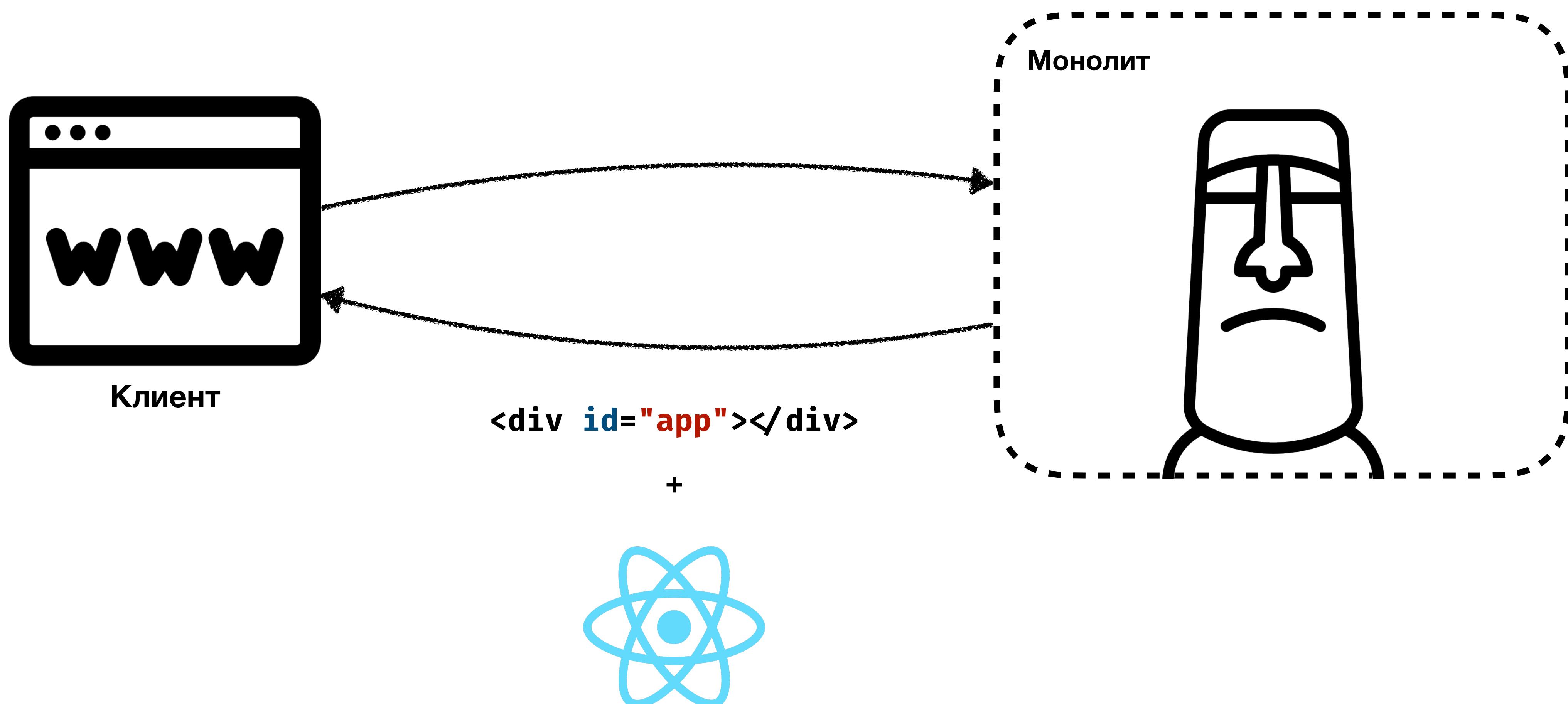


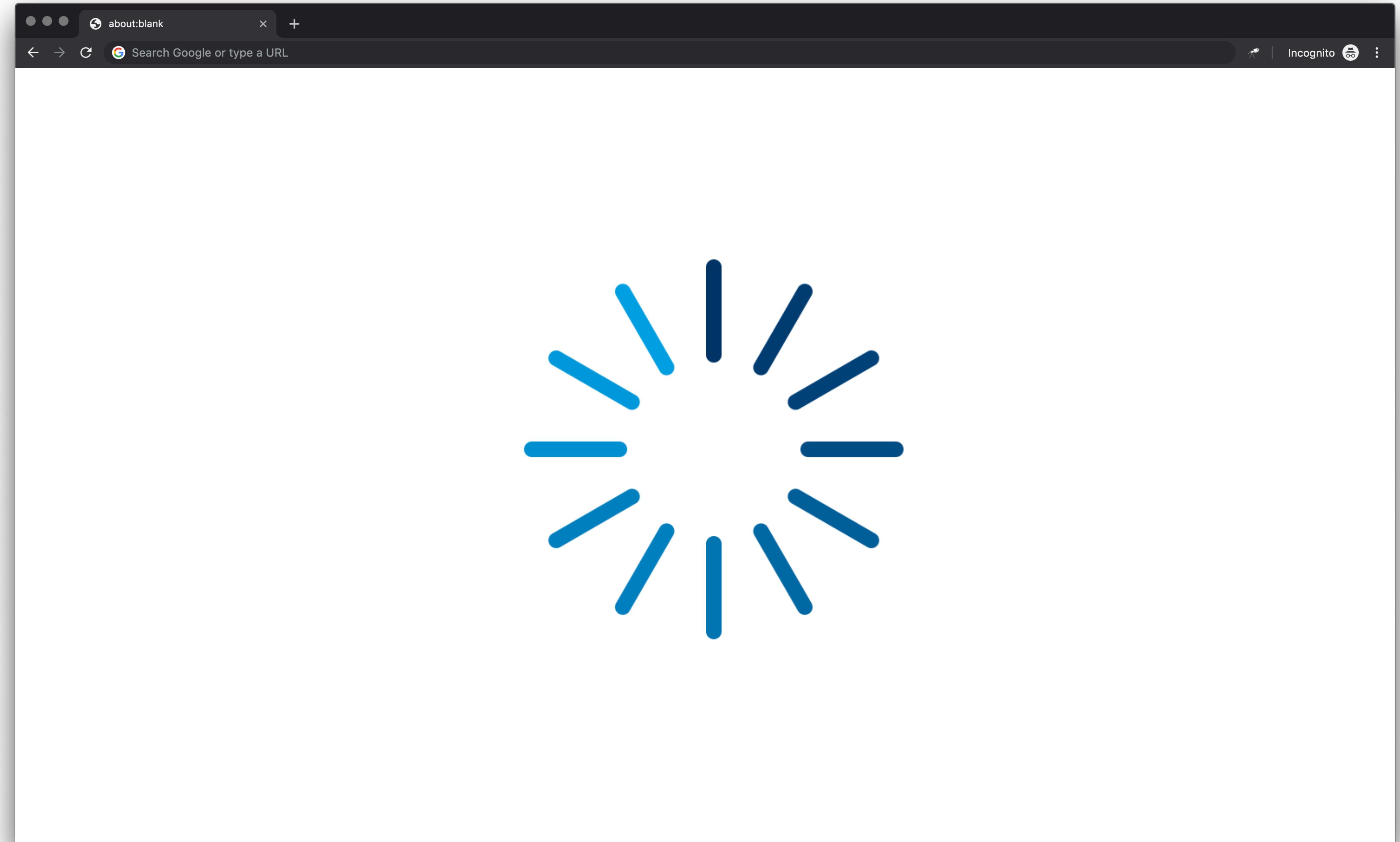
# Microapplication

- NPM пакет
- Содержит всю логику и поведение отдельного блока
- Разворачивается с помощью скаффолдера
- Тулинг для тестирования и разработки

# Growing a seed

- Можно последовательно заменять легаси
- **Возникают проблемы, если бекенд не NodeJS**

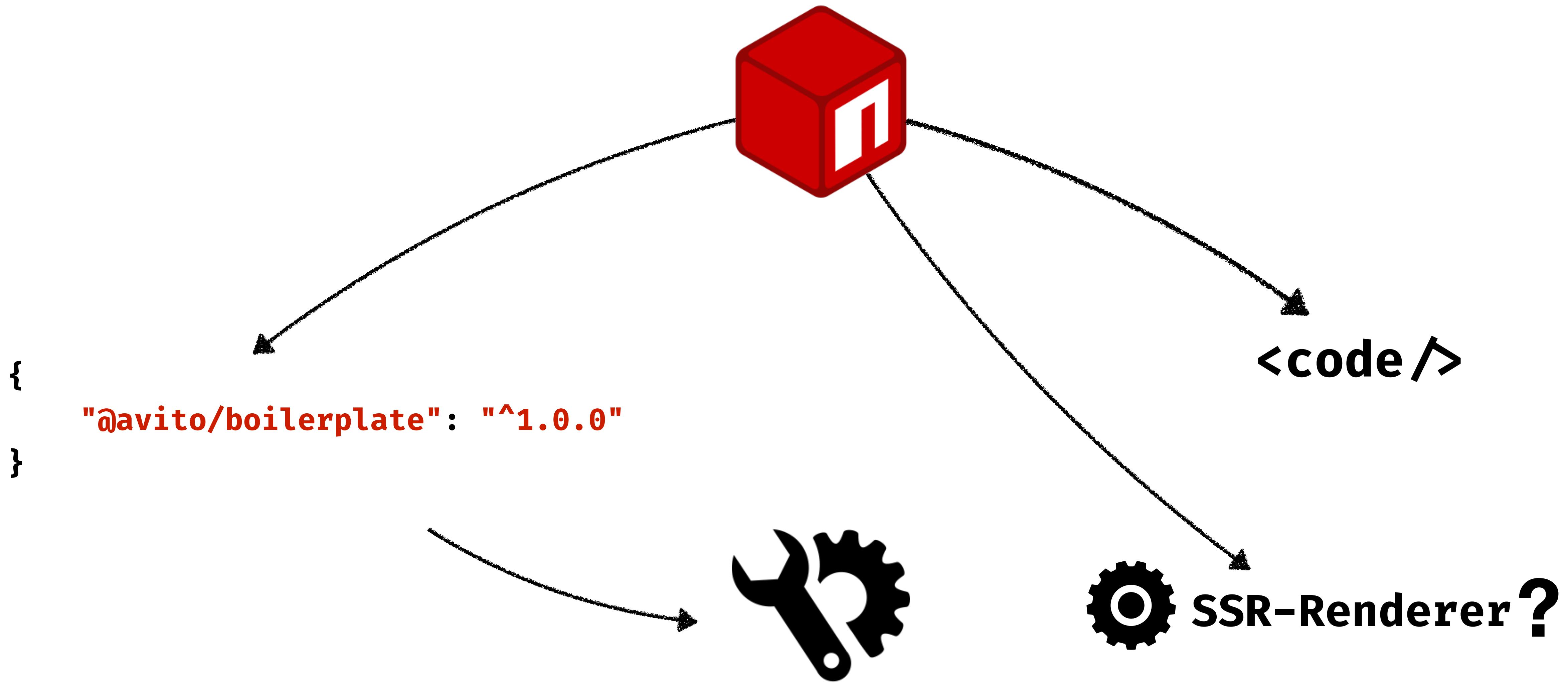




**SEO ❤ SSR**

# SSR





# Microapp SSR renderer

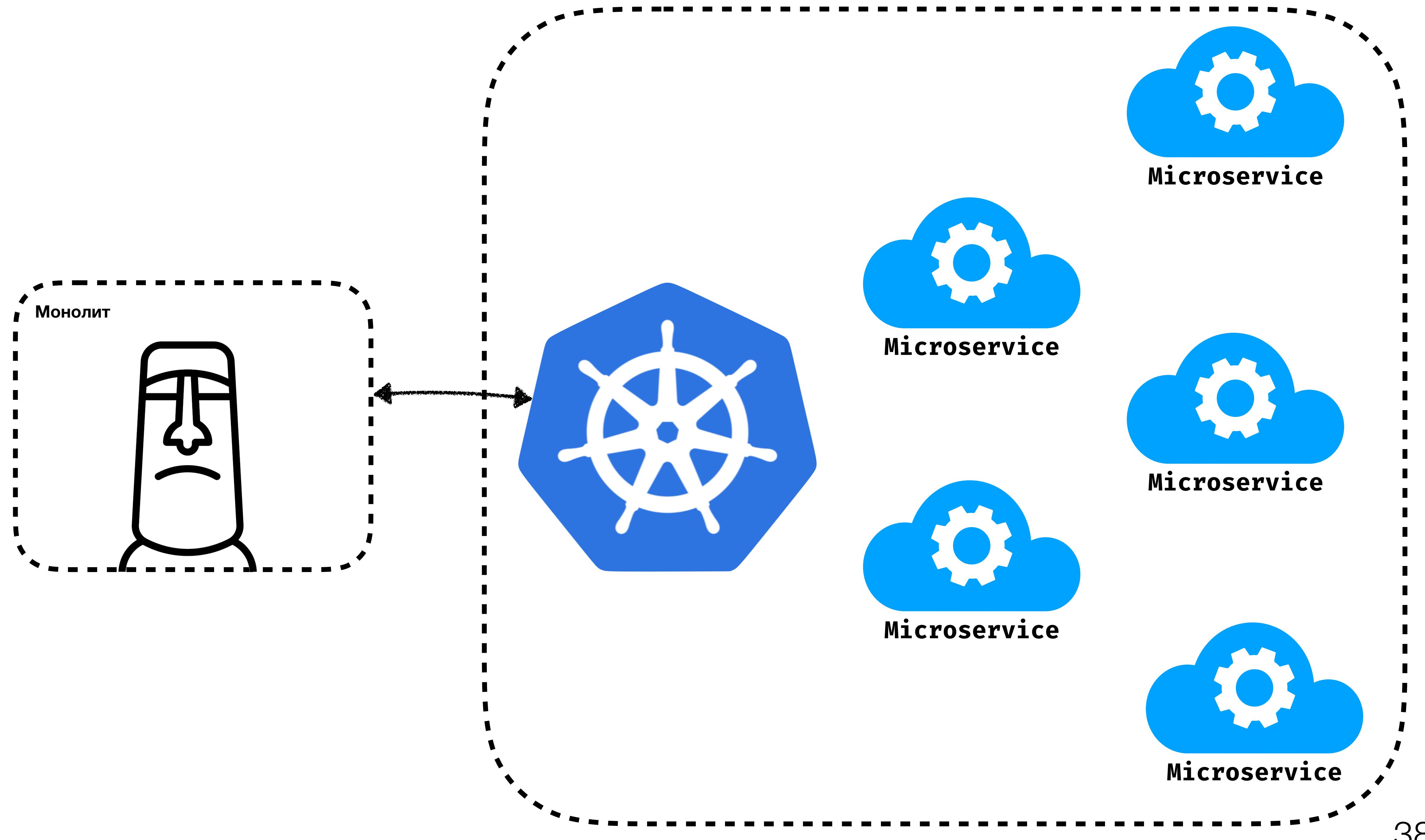
- Код рендерера в каждом пакете микроприложения
- Код рендерера нужно как-то обновлять
- Разработчики смогут вносить изменения в рендеринг, тем самым будет расти энтропия
- Поддержка огромного количества версий рендерера
- Пакеты с рендерингом нужно как-то деплоить

# Common Renderer

# **PHP + JavaScript?**

# PHP + JavaScript

- V8js
- ...?

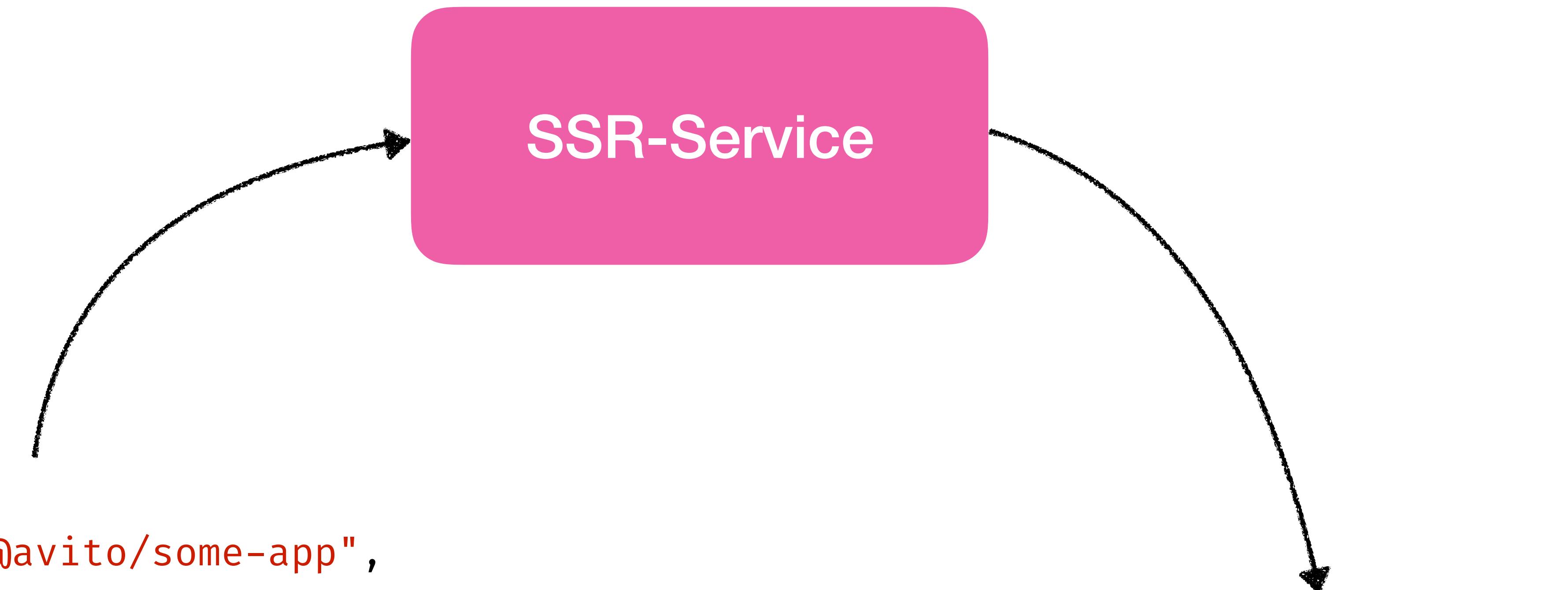


# **SSR as Service**

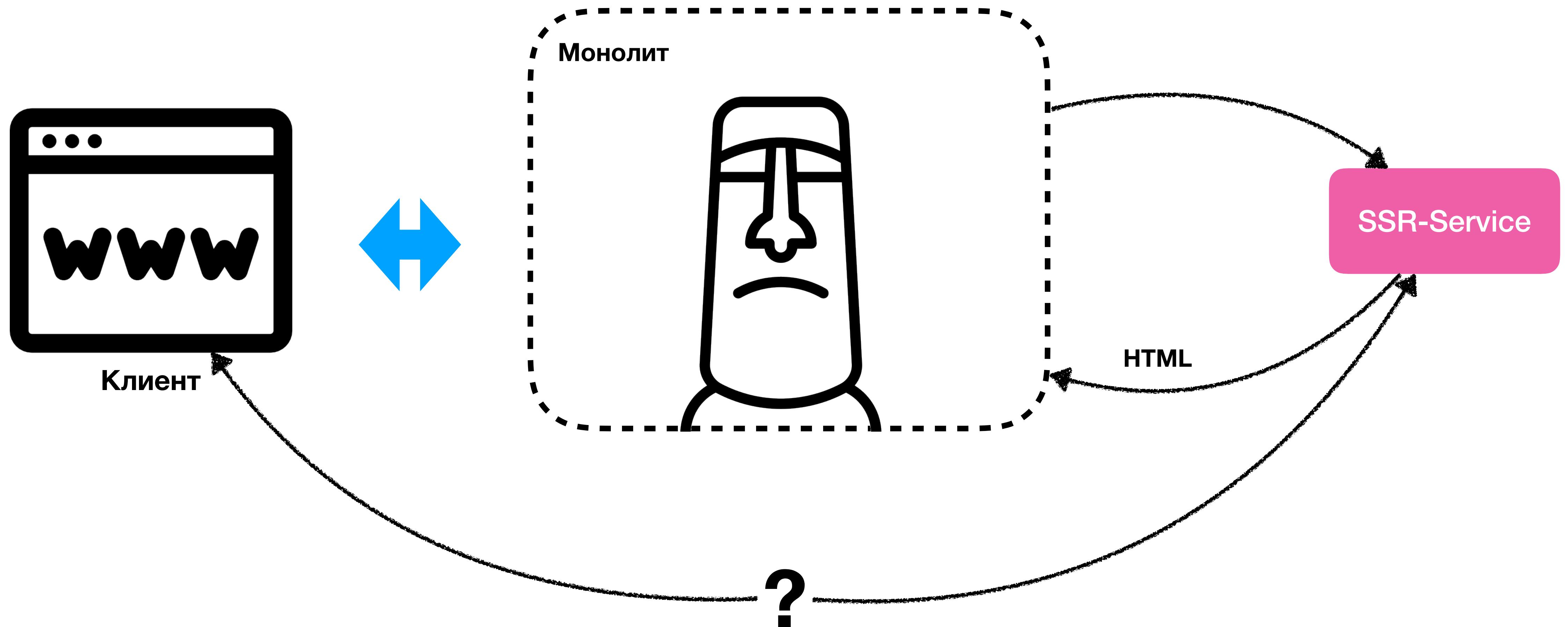
# SSR Service

$f(\text{appName}, \text{version}, \text{props}) \rightarrow \text{HTML}$

```
{  
  "app": "@avito/some-app",  
  "version": "1.2.3",  
  "params": {  
    "prop": "value"  
  }  
}
```



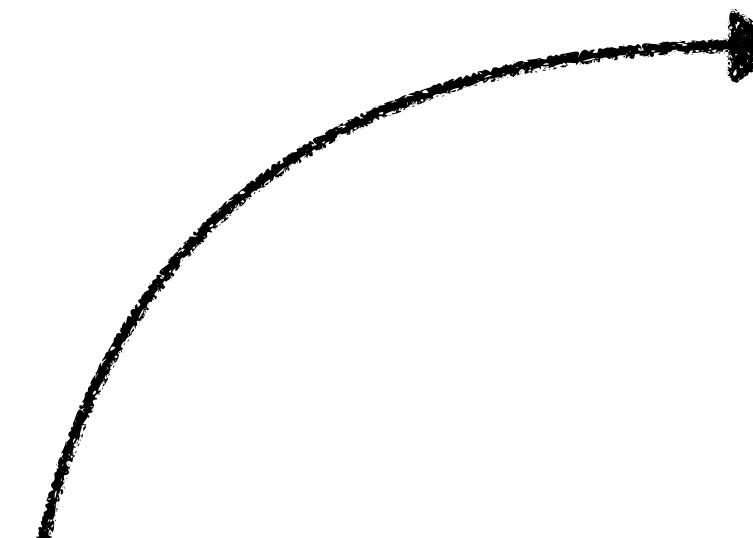
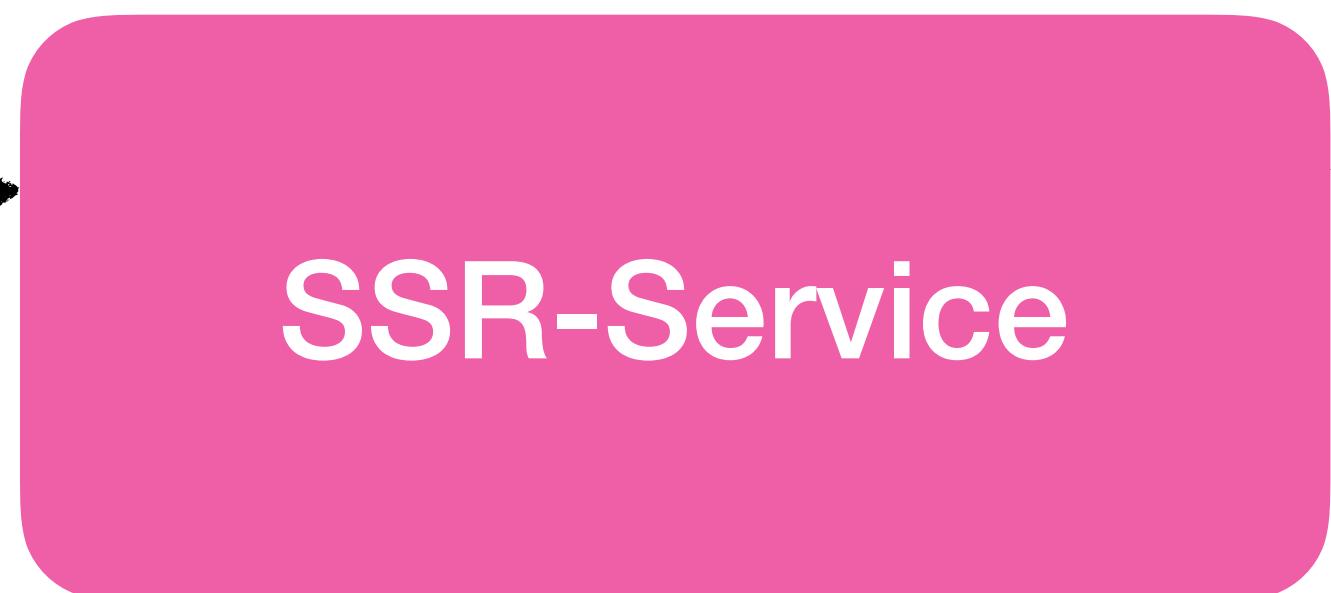
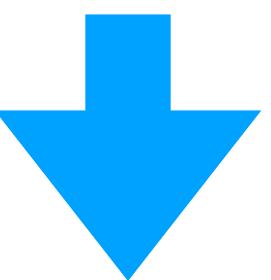
<html>



# SSR Service

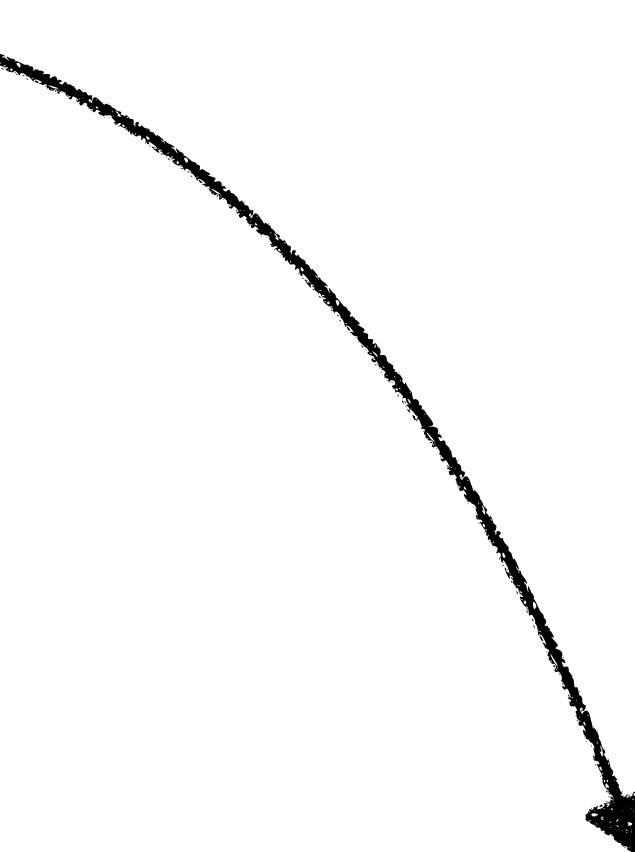
- Как сервис узнает что именно ему рендерить?
- Как на клиенте реидентифицировать состояние отрендеренного приложения?

# { code }



<html>

```
{  
  "app": "avito/some-app",  
  "version": "1.2.3",  
  "params": {  
    "prop": "value"  
  }  
}
```



# Build as a Service

# BaaS

# BaaS

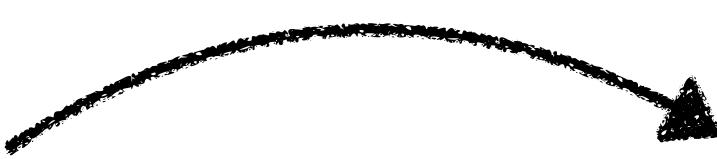
- Сервис для сборки пакетов
- Собирает артефакты (бандлы, стили, статика)
- Собирает манифесты приложений
- Выкладывает артефакты на сервера статики



Developer

**npm publish**

BaaS



# SSR microapp

- NPM-пакет
- Точка входа для клиента
- Точка входа для сервера
- Флаг **ssr: true** в package.json

# client-entry.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './app';

export const render = (mountNode, props) => {
  const Application = props => (
    <App {...props} />
  );

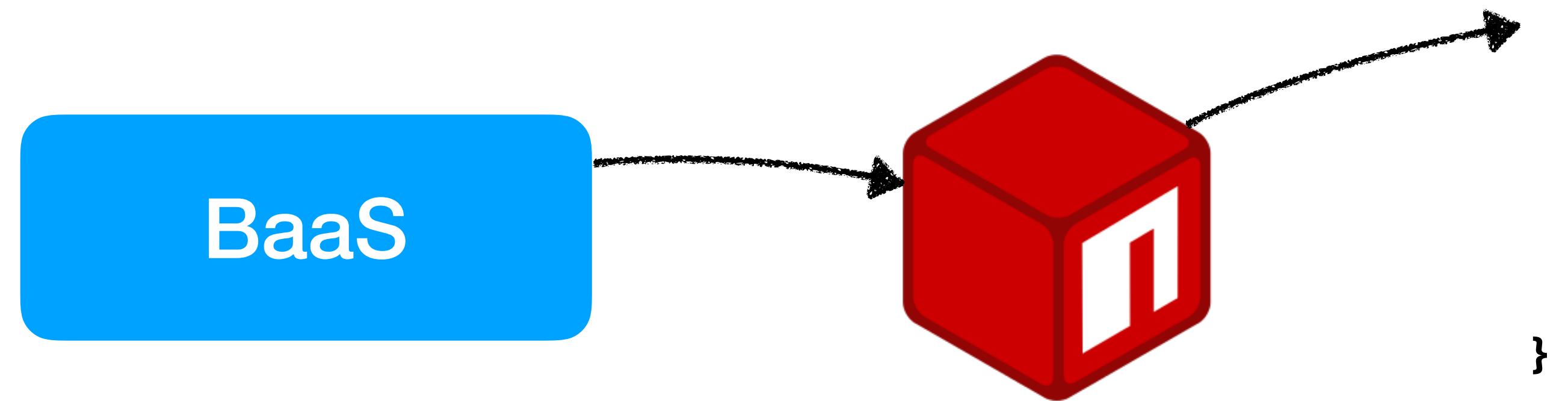
  if (mountNode.querySelector('[data-reactroot]')) {
    ReactDOM.hydrate(Application(props), mountNode);
  } else {
    ReactDOM.render(Application(props), mountNode);
  }
};
```

# server-entry.js

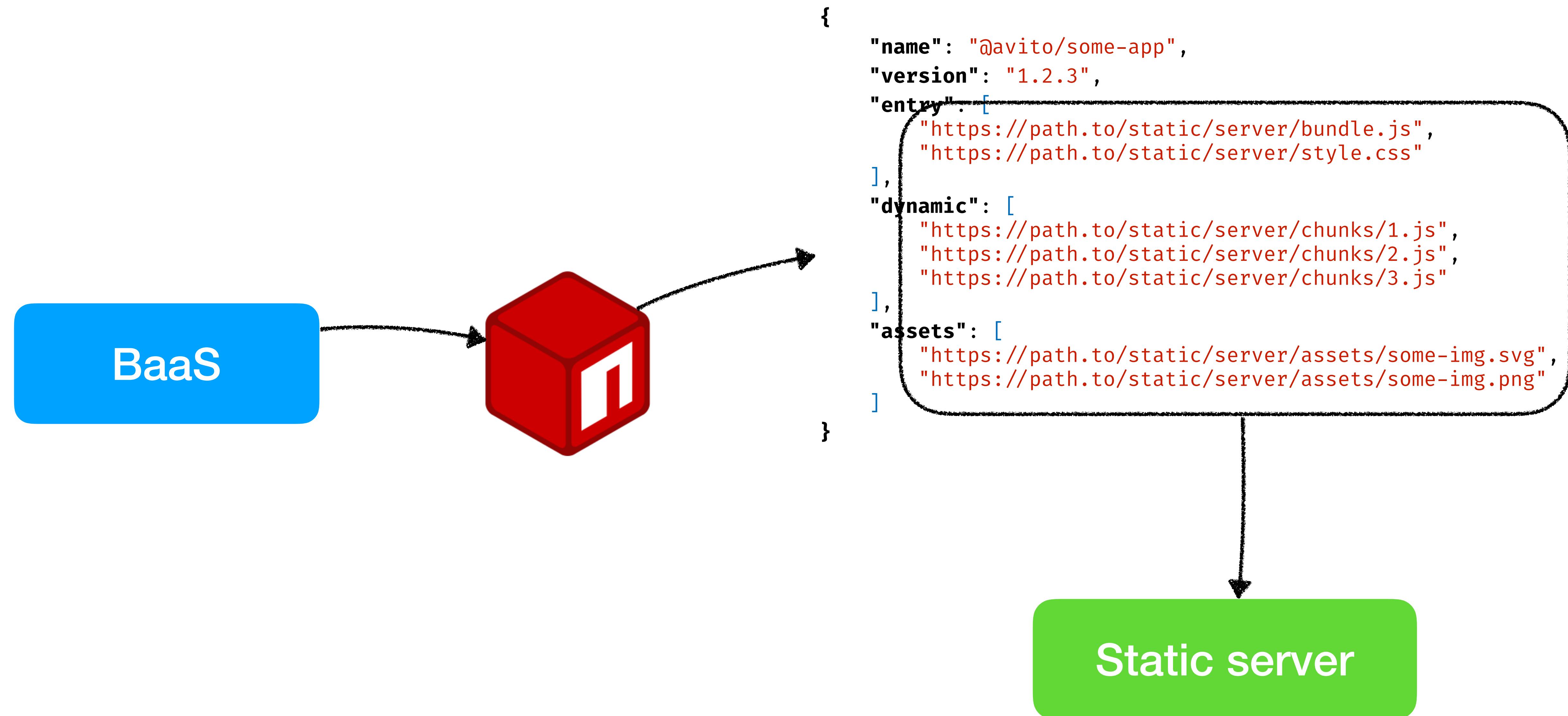
```
import React from 'react';
import { renderToString } from 'react-dom/server';
import App from './app';

export const render = (props) => {
  const html = renderToString(
    <App {...props} />
  );

  return { html };
};
```

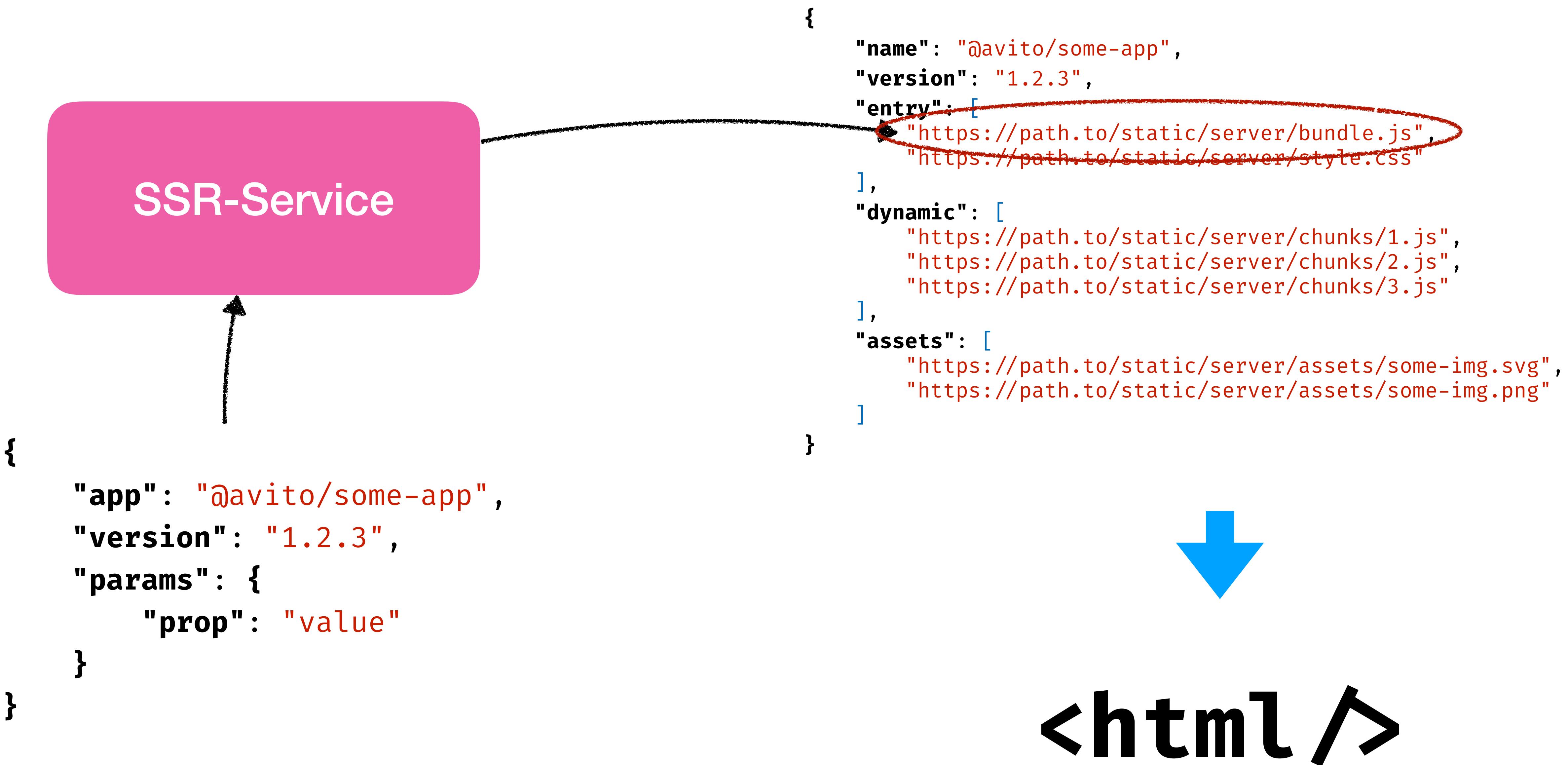


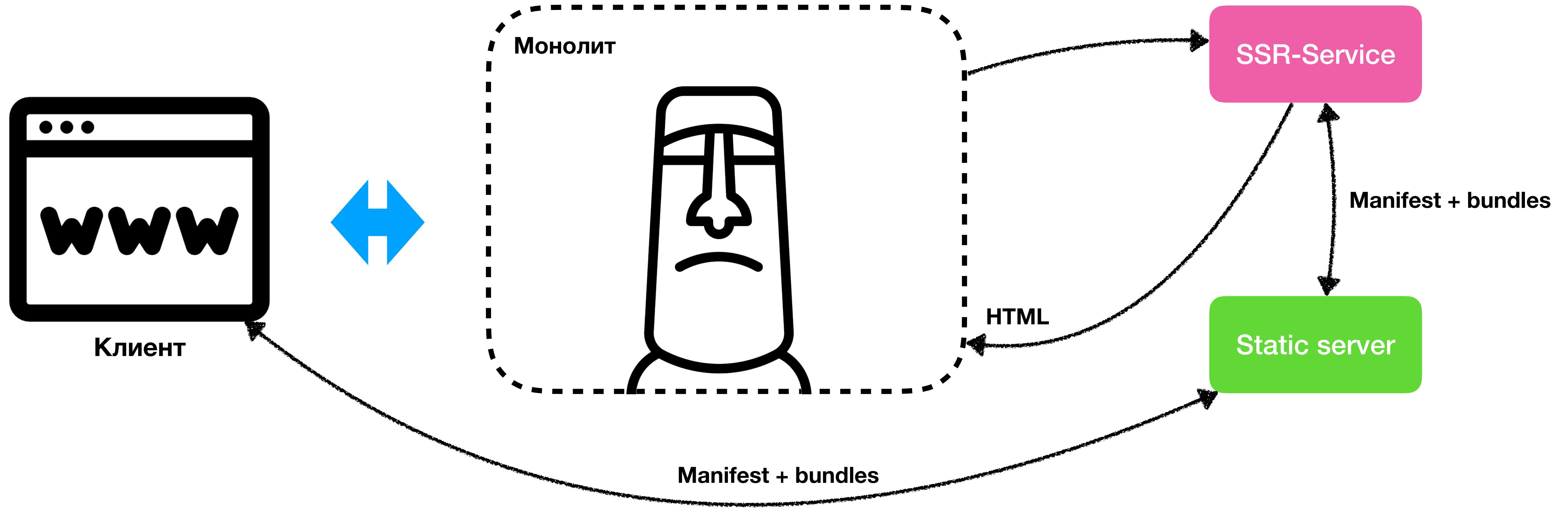
```
{  
  "name": "@avito/some-app",  
  "version": "1.2.3",  
  "entry": [  
    "https://path.to/static/server/bundle.js",  
    "https://path.to/static/server/style.css"  
  ],  
  "dynamic": [  
    "https://path.to/static/server/chunks/1.js",  
    "https://path.to/static/server/chunks/2.js",  
    "https://path.to/static/server/chunks/3.js"  
  ],  
  "assets": [  
    "https://path.to/static/server/assets/some-img.svg",  
    "https://path.to/static/server/assets/some-img.png"  
  ]  
}
```



# App Manifest

- Имя приложения
- Версию
- Содержит точки входа в приложение
- Динамические чанки
- Статические ассеты

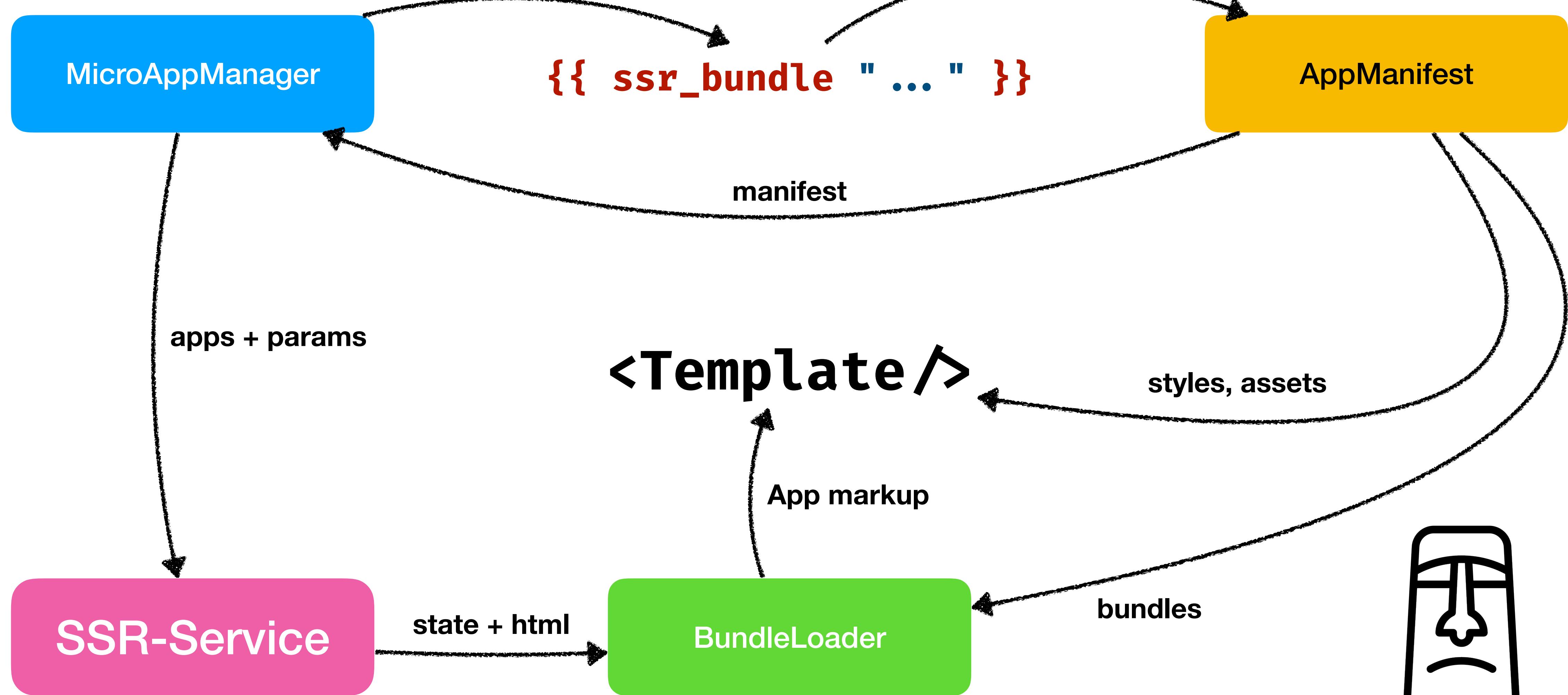




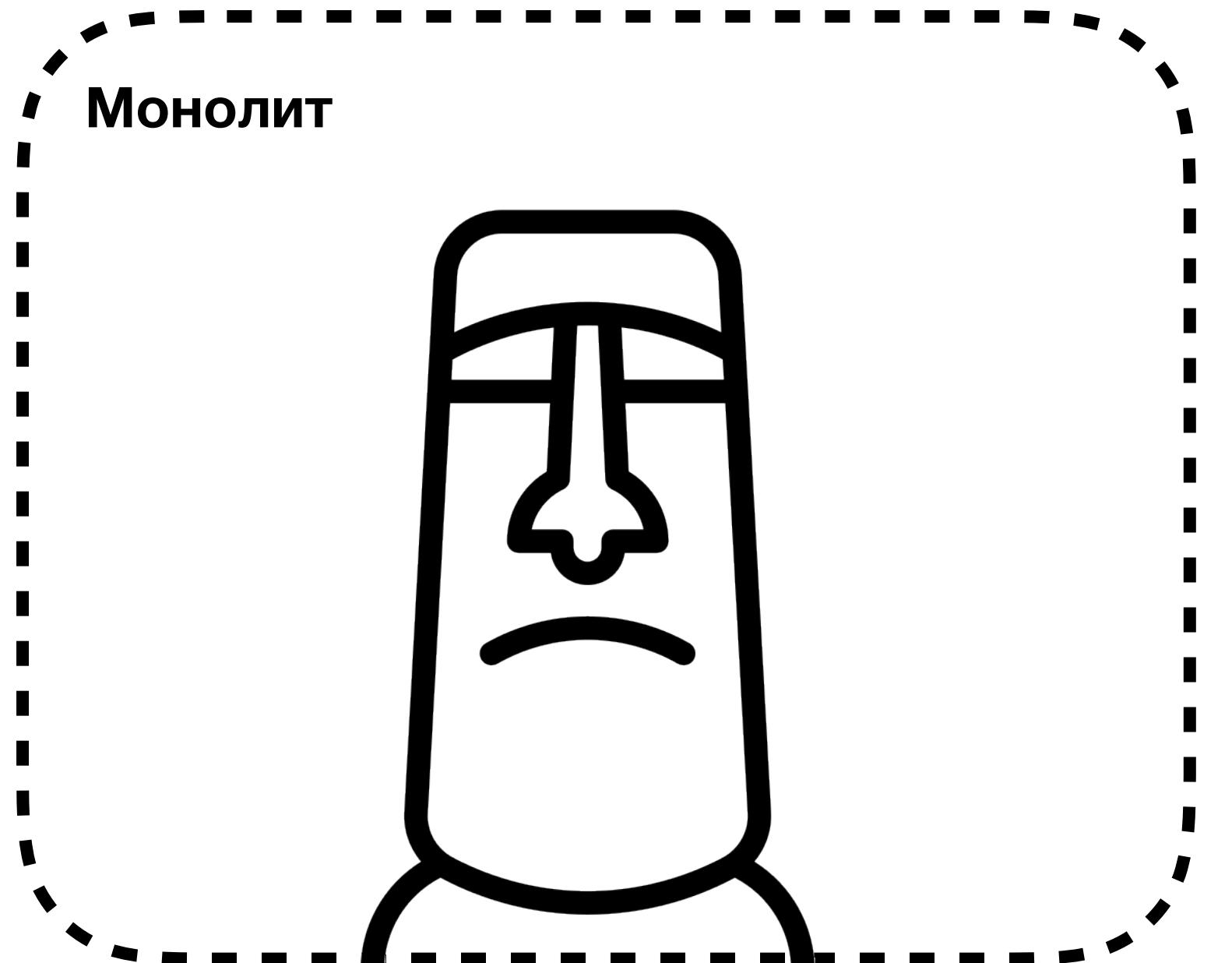
# Интеграция в монолит

```
{{ ssr_bundle "some_app" }}  
<html>  
  <head>  
    {{ ssr_css "some_app" }}  
  </head>  
  <body>  
    <content/>  
    {{ ssr_content "some_app" someProps }}  
    {{ ssr_js "some_app" }}  
  </body>  
</html>
```

# Монолит



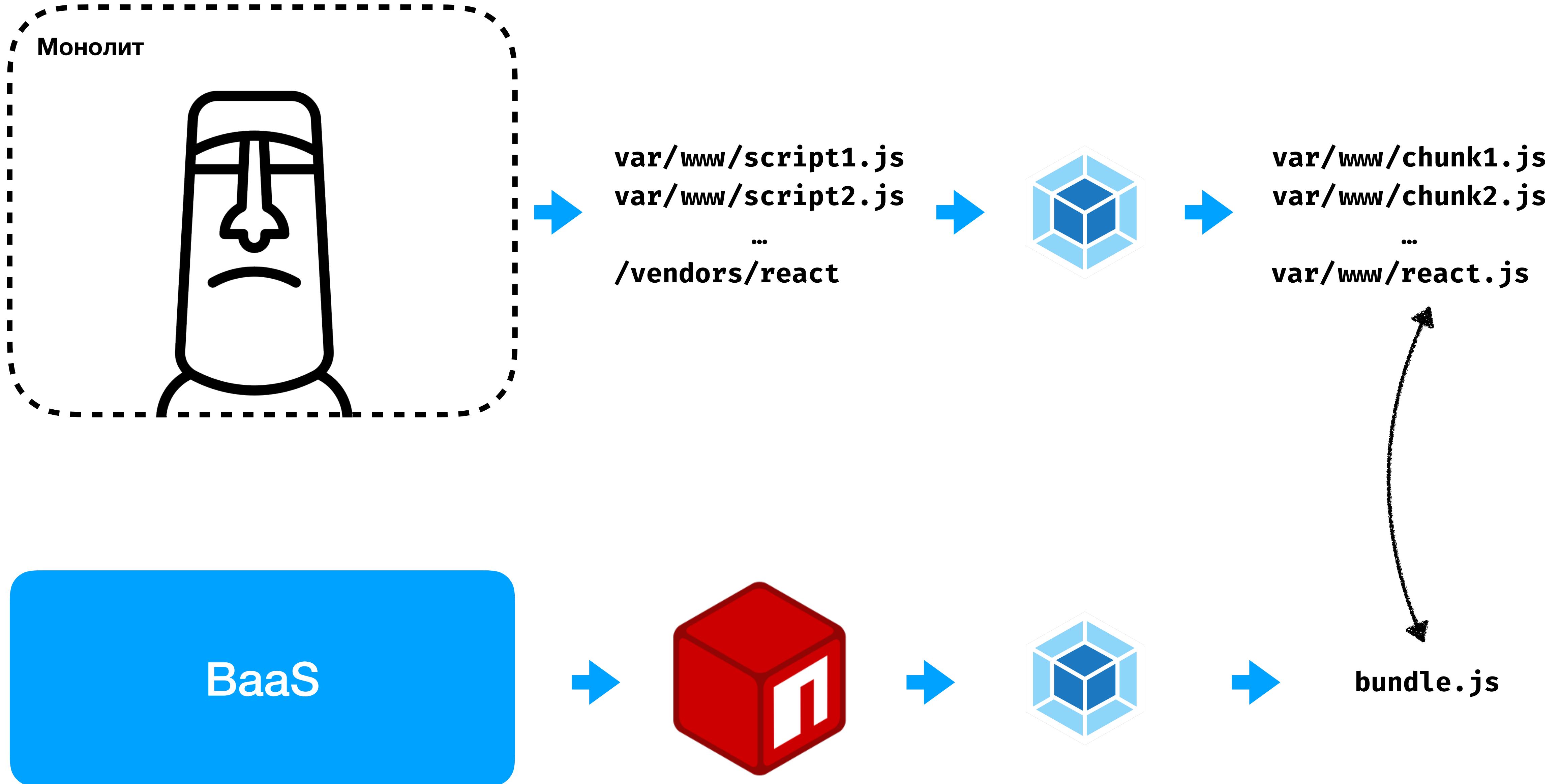
# Проблемы внешней сборки

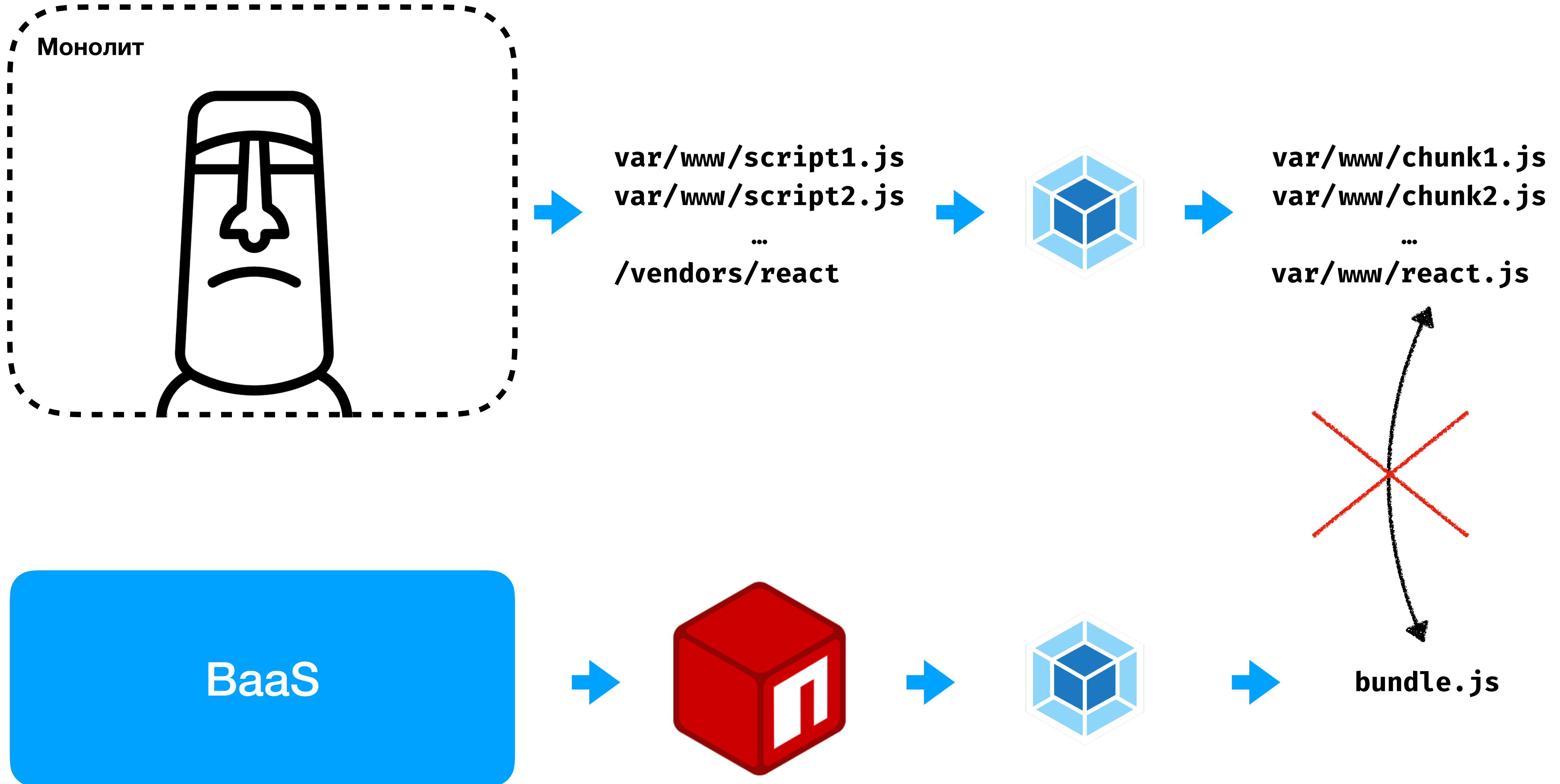


`var/www/script1.js`  
`var/www/script2.js`  
...  
`/vendors/react`



`var/www/chunk1.js`  
`var/www/chunk2.js`  
...  
`var/www/react.js`

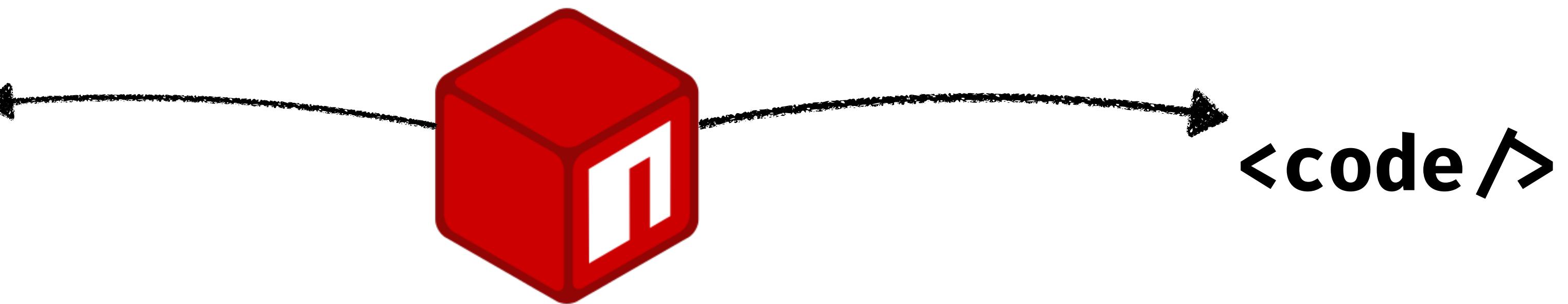




# Проблемы внешней сборки

- Обособленные сборки собирают и оптимизируют код под конкретные точки входа
- Сторонние микроприложения собираются со своими зависимостями
- Невозможно переиспользовать зависимости между хостовым окружением и микроприложениями

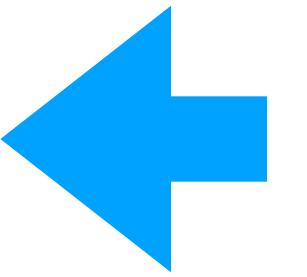
```
{  
  "@avito/eslint-config-es5": "^2.5.0",  
  "@avito/eslint-config-es6": "^1.7.0",  
  "@avito/eslint-config-react": "^2.7.0",  
  "@avito/jest-snapshot-image": "^2.0.2",  
  "@avito/phakt": "^9.6.4",  
  "@avito/react-snapshot": "^3.7.0",  
  "@avito/react-utils": "^2.1.2",  
  "@avito/stylelint-config": "^2.0.0",  
  "@avito/test-utils": "^1.3.2",  
  "@avito/theme-avito-site": "0.0.1",  
  "@avito/webpack-plugin-theming": "^2.0.1",  
  "ajv": "^6.5.5",  
  "autoprefixer": "^9.3.1",  
  "babel-core": "^6.26.3",  
  "babel-jest": "^23.6.0",  
  "babel-loader": "^7.1.5",  
  "babel-plugin-css-modules-transform": "^1.6.2",  
  "babel-plugin-dynamic-import-node": "^2.2.0",  
  "babel-plugin-external-helpers": "^6.22.0",  
  "babel-plugin-syntax-dynamic-import": "^6.18.0",  
  "babel-plugin-transform-async-to-generator": "^6.24.1",  
  "babel-plugin-transform-class-properties": "^6.24.1",  
  "babel-plugin-transform-export-extensions": "^6.22.0",  
  "babel-plugin-transform-object-rest-spread": "^6.26.0",  
  "babel-plugin-transform-react-remove-prop-types": "^0.4.20",  
  "babel-polyfill": "^6.26.0",  
  "babel-preset-env": "^1.7.0",  
  "babel-preset-react": "^6.24.1",  
  "chalk": "^2.4.1",  
  "css-loader": "^1.0.1",  
  "edit-json-file": "^1.2.0",  
  "eslint": "4.4.1",  
  "eslint-plugin-import": "2.11.0",  
  "eslint-teamcity": "2.0.1",  
  "express": "^4.16.4",  
  "express-http-proxy": "^1.5.0",  
  "html-webpack-plugin": "3.2.0",  
  "intl": "^1.2.5",  
  "jest": "^23.6.0",  
  "jest-teamcity-reporter": "^0.9.0",  
  "mime": "^2.4.0",  
  "postcss-loader": "3.0.0",  
  "react": "^16.6.3",  
  "react-dom": "16.6.3",  
  "react-hot-loader": "4.3.12",  
  "react-test-renderer": "16.6.3",  
  "style-loader": "0.23.1",  
  "stylelint": "8.4.0",  
  "stylelint-csstree-validator": "1.3.0",  
  "stylelint-teamcity-reporter": "1.0.0",  
  "url-loader": "1.1.2",  
  "webpack": "4.26.1",  
  "webpack-dev-middleware": "3.4.0",  
  "webpack-hot-middleware": "2.24.3"  
}
```



<code />

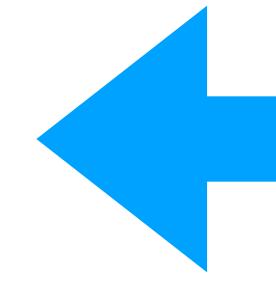
```
{  
  "@avito/eslint-config-es5": "^2.5.0",  
  "@avito/eslint-config-es6": "^1.7.0",  
  "@avito/eslint-config-react": "^2.7.0",  
  "@avito/jest-snapshot-image": "^2.0.2",  
  "@avito/phakt": "^9.6.4",  
  "@avito/react-snapshot": "^3.7.0",  
  "@avito/react-utils": "^2.1.2",  
  "@avito/stylelint-config": "^2.0.0",  
  "@avito/test-utils": "^1.3.2",  
  "@avito/theme-avito-site": "0.0.1",  
  "@avito/webpack-plugin-theming": "^2.0.1",  
  "ajv": "^6.5.5",  
  "autoprefixer": "^9.3.1",  
  "babel-core": "^6.26.3",  
  "babel-jest": "^23.6.0",  
  "babel-loader": "^7.1.5",  
  "babel-plugin-css-modules-transform": "^1.6.2",  
  "babel-plugin-dynamic-import-node": "^2.2.0",  
  "babel-plugin-external-helpers": "^6.22.0",  
  "babel-plugin-syntax-dynamic-import": "^6.18.0",  
  "babel-plugin-transform-async-to-generator": "^6.24.1",  
  "babel-plugin-transform-class-properties": "^6.24.1",  
  "babel-plugin-transform-export-extensions": "^6.22.0",  
  "babel-plugin-transform-object-rest-spread": "^6.26.0",  
  "babel-plugin-transform-react-remove-prop-types": "^0.4.20",  
  "babel-polyfill": "^6.26.0",  
  "babel-preset-env": "^1.7.0",  
  "babel-preset-react": "^6.24.1",  
  "chalk": "^2.4.1",  
  "css-loader": "^1.0.1",  
  "edit-json-file": "^1.2.0",  
  "eslint": "4.4.1",  
  "eslint-plugin-import": "2.11.0",  
  "eslint-teamcity": "2.0.1",  
  "express": "^4.16.4",  
  "express-http-proxy": "^1.5.0",  
  "html-webpack-plugin": "3.2.0",  
  "intl": "^1.2.5",  
  "jest": "^23.6.0",  
  "jest-teamcity-reporter": "^0.9.0",  
  "mime": "^2.4.0",  
  "postcss-loader": "3.0.0",  
  "react": "^16.6.3",  
  "react-dom": "^16.6.3",  
  "react-hot-loader": "^4.3.12",  
  "react-test-renderer": "16.6.3",  
  "style-loader": "0.23.1",  
  "stylelint": "8.4.0",  
  "stylelint-csstree-validator": "1.3.0",  
  "stylelint-teamcity-reporter": "1.0.0",  
  "url-loader": "1.1.2",  
  "webpack": "4.26.1",  
  "webpack-dev-middleware": "3.4.0",  
  "webpack-hot-middleware": "2.24.3"  
}
```

<code>



Baas

<code/>



BaaS

# Переиспользование зависимостей

- Зависимости пакета должны быть собраны отдельно
- Зависимости пакета должны быть доступны вместе с бандлом пакета
- Зависимости пакета должны переиспользоваться другими пакетами

# Dependency Injection

DI

**In software engineering, dependency injection is a technique whereby one object supplies the dependencies of another object.**

*–Wikipedia*

```
import React from 'react';
import { Button } from '@avito/ui';
import Component from './component';

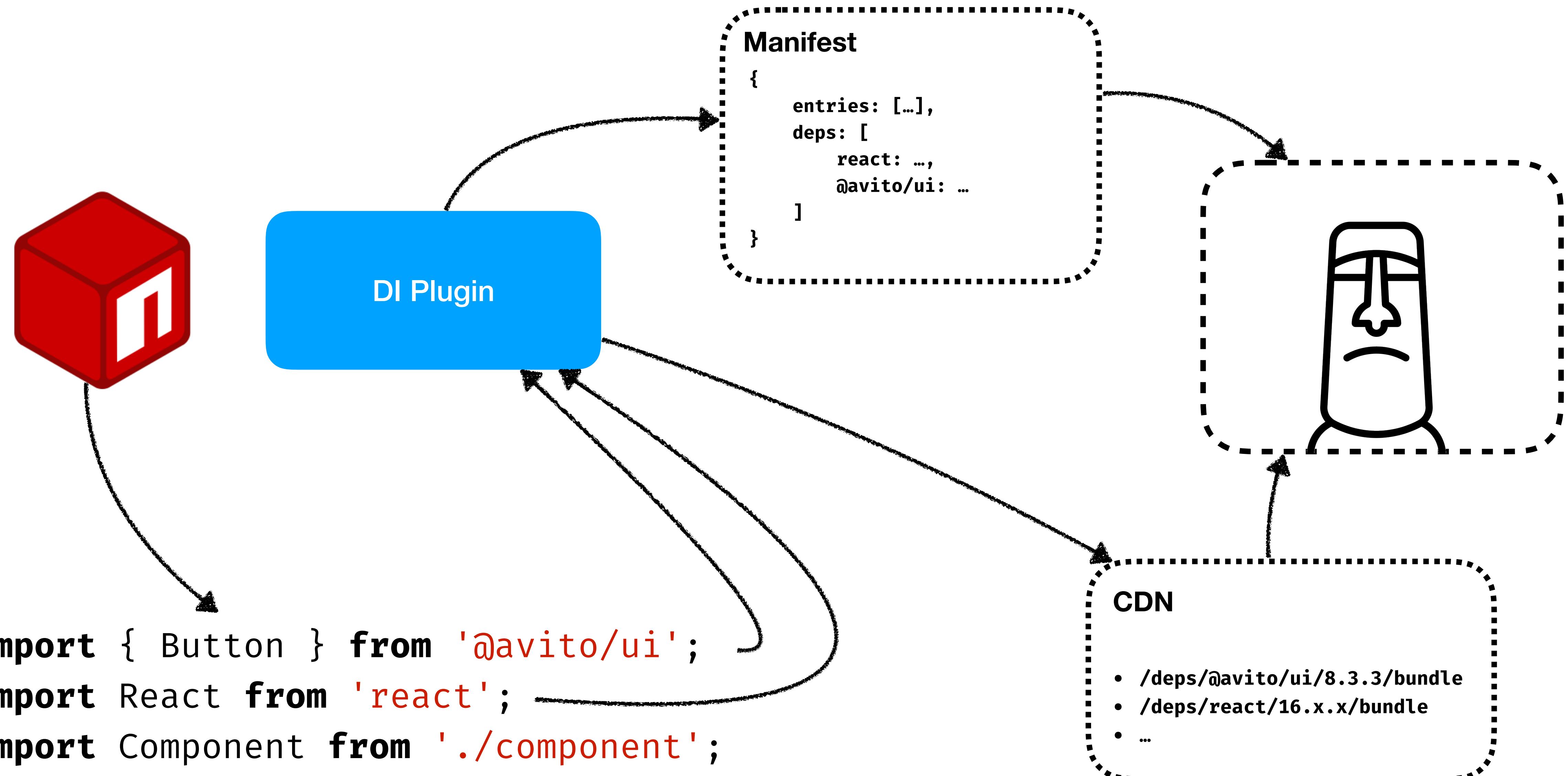
export default class ...
```

```
import React from 'react';  
import { Button } from '@avito/ui';  
import Component from './component';
```

```
export default class ...
```

```
const React = DI.get('react');
const { Button } = DI.get('@avito/ui');
import Component from './component';

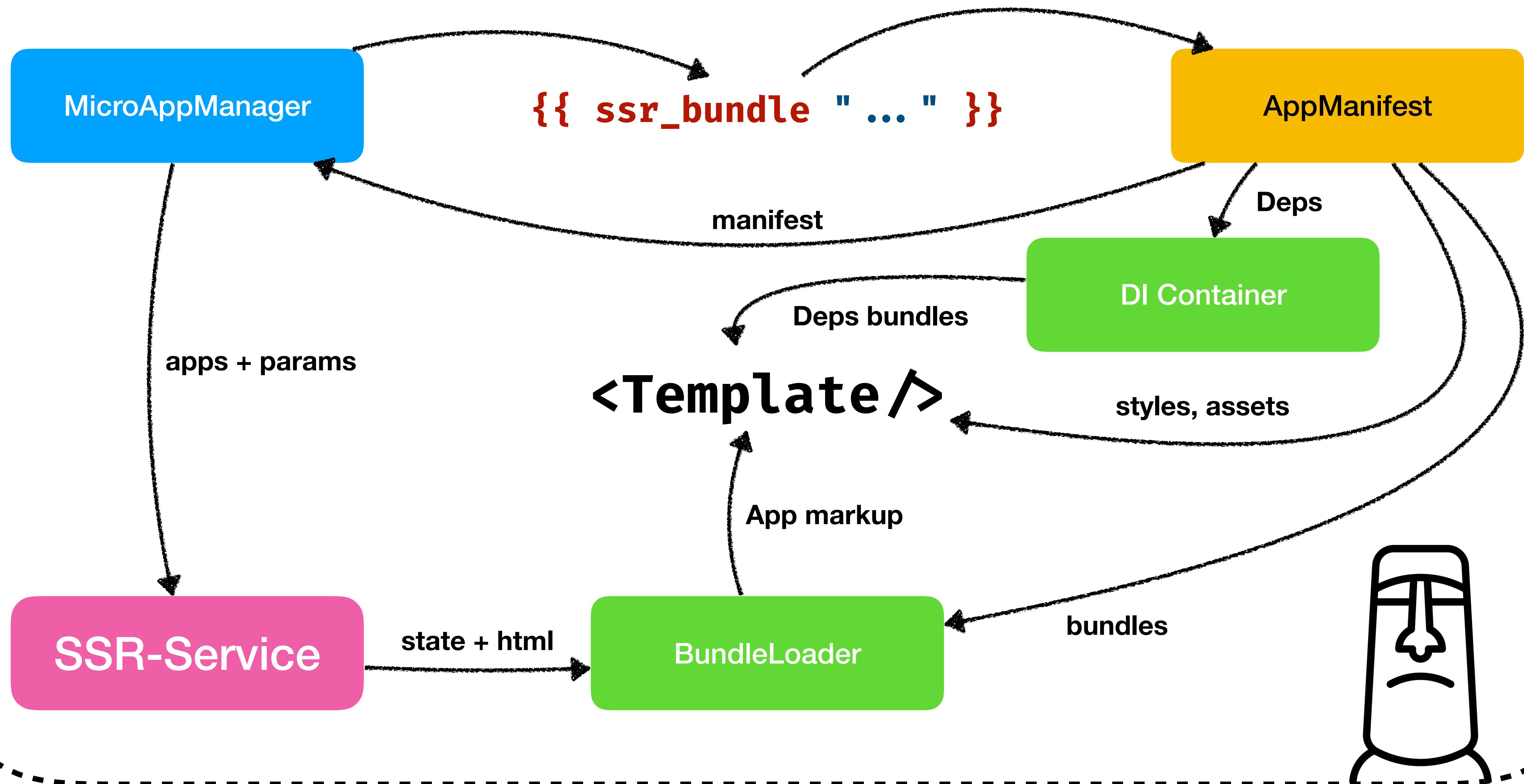
export default class ...
```



# DI

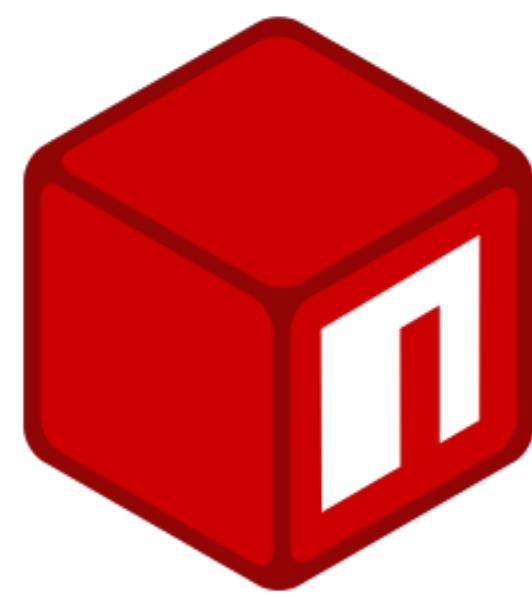
- При сборке пакета, вычисляются зависимости пакета
- Импорты зависимостей заменяются на обращения к DI-контейнеру
- Параллельно запускаются сборки зависимостей (и зависимостей зависимостей рекурсивно)

# Монолит

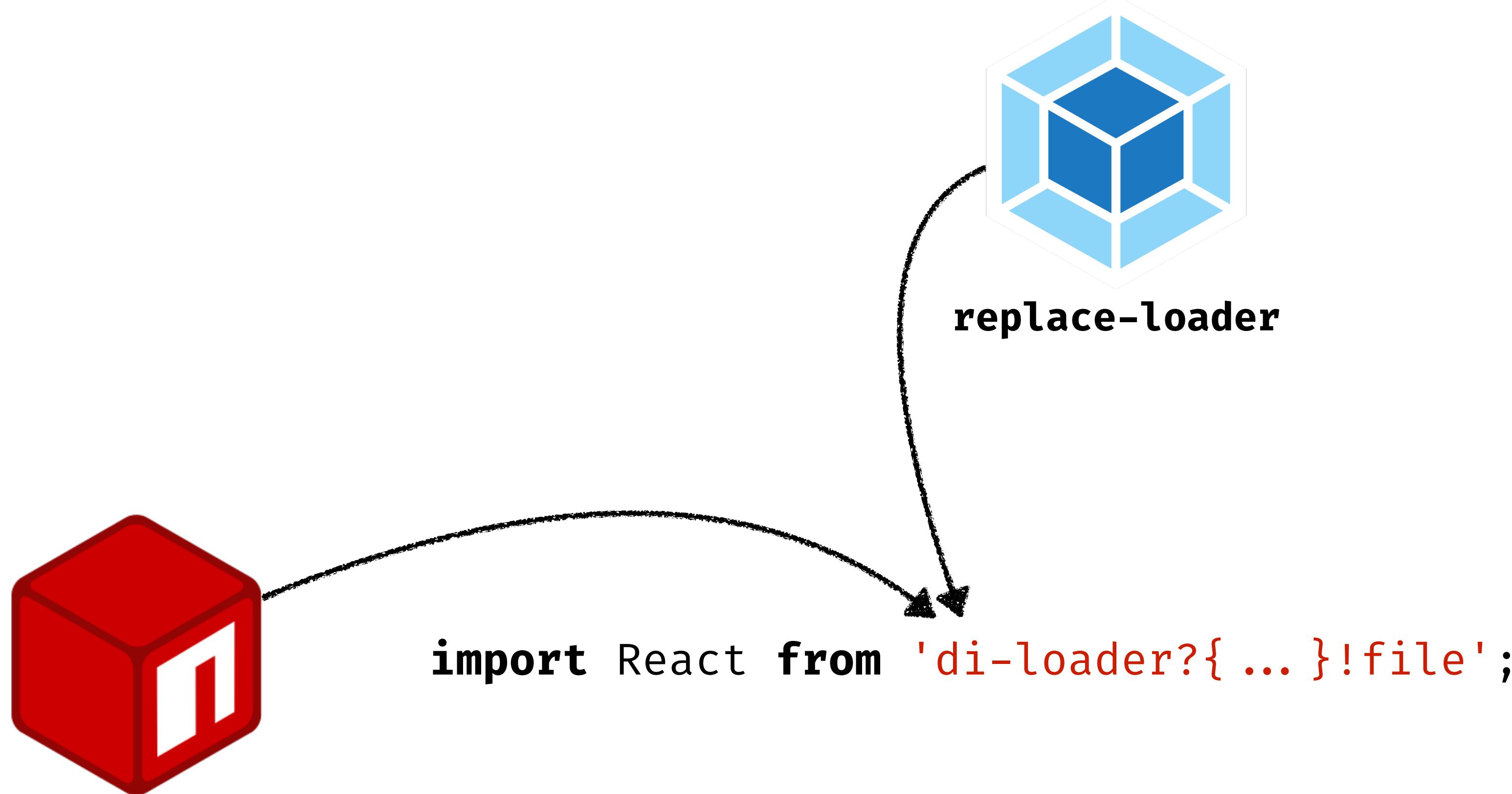


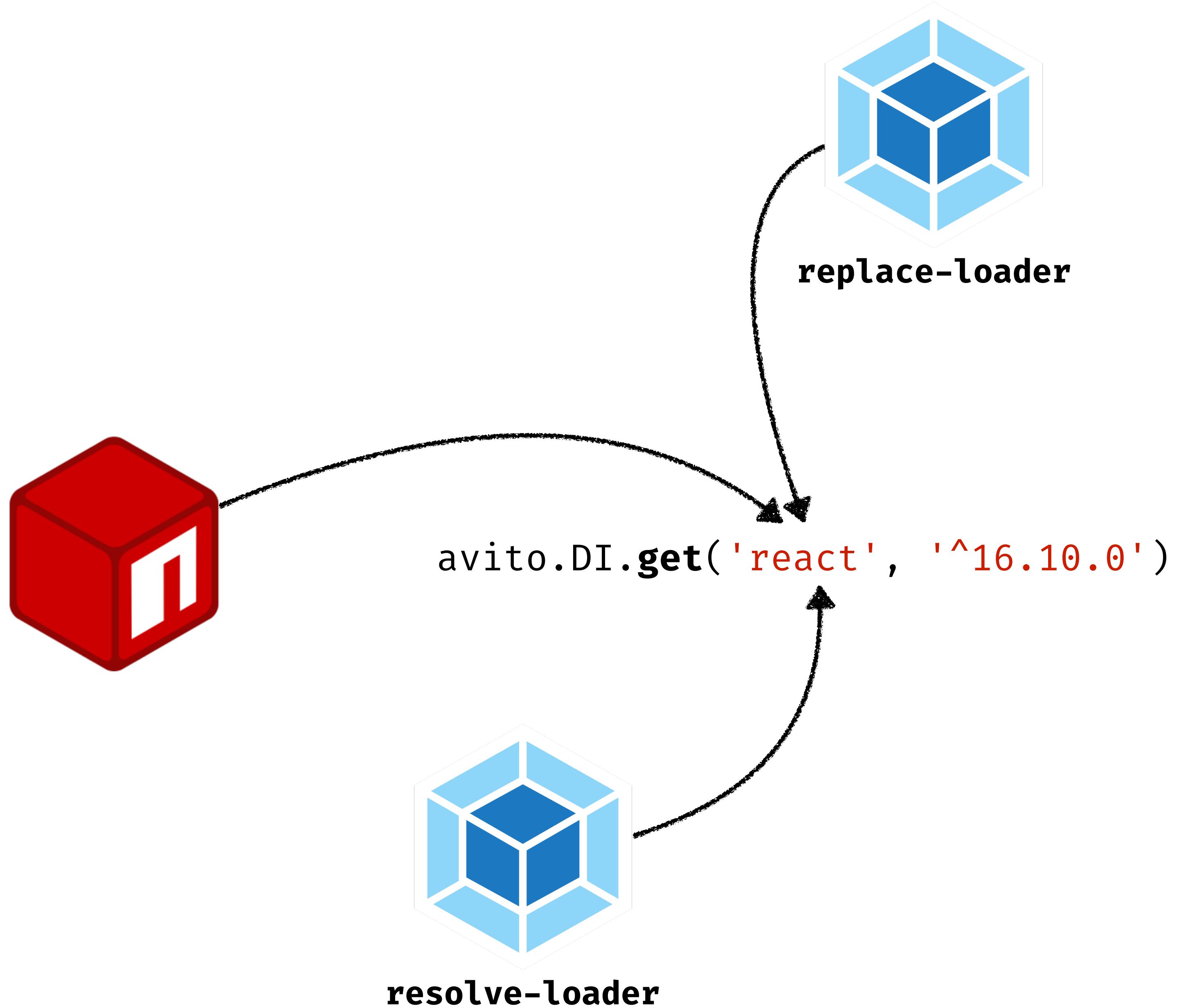
# DI Implementation

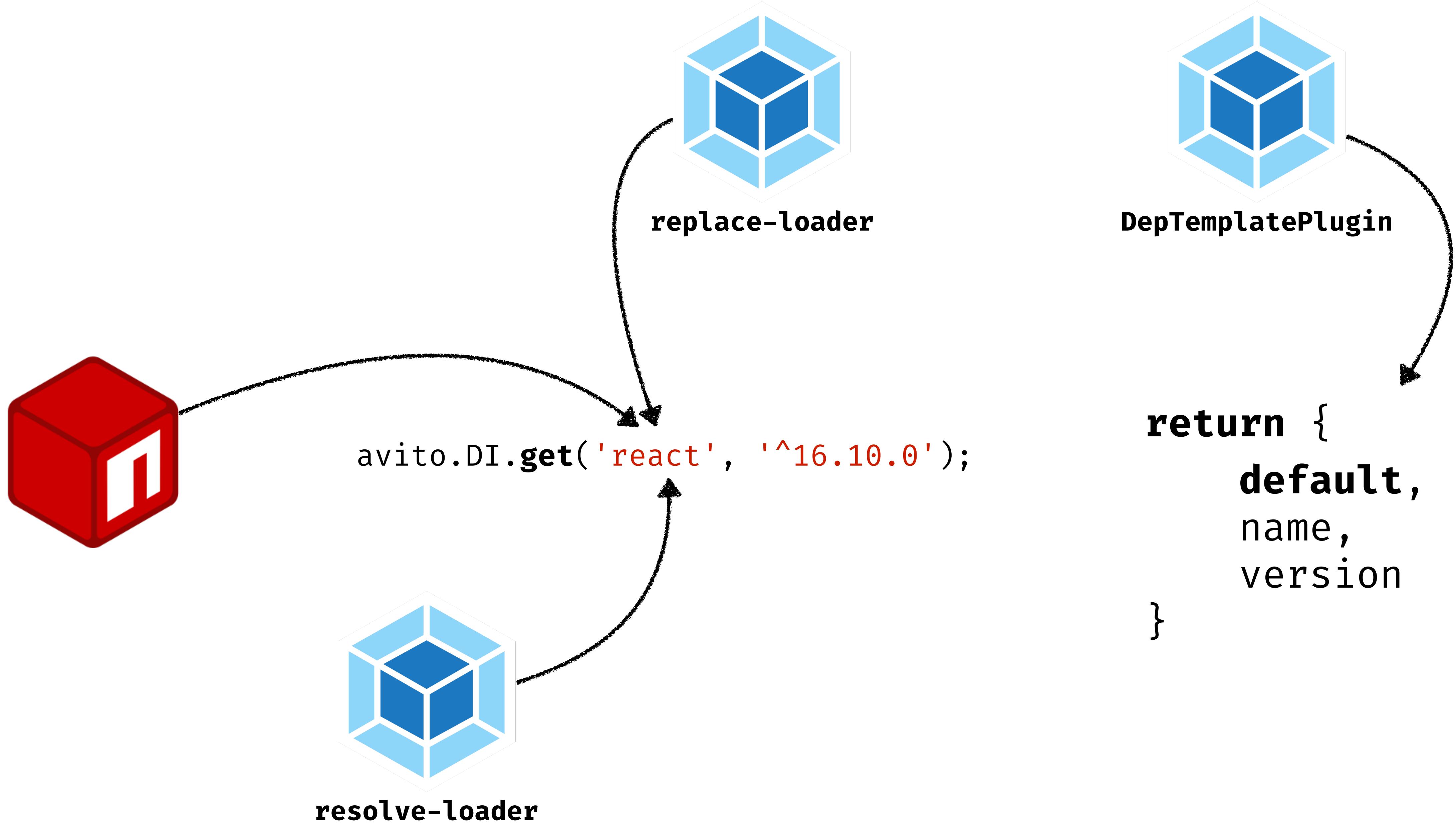
- Webpack loaders (replace-loader, resolve-loader)
- Webpack Tempalte plugin
- DI container



**import React from 'react';**







```
const module = require('module');
```

```
import module from 'module';
```

```
import { foo as bar } from 'module';
```

```
import * as foo from 'module';
```

```
export { foo } from 'module';
```

```
export { foo as bar } from 'module';
```

```
const module = avito.DI.get('module');
```

```
const module = avito.DI.get('module');
```

```
const bar = avito.DI.get('module').foo;
```

```
const foo = avito.DI.get('module');
```

```
const foo = avito.DI.get('module').foo;
```

```
const bar = avito.DI.get('module').foo;
```

```
const module = avito.DI.get('module');  
const module = avito.DI.get('module');  
const bar = avito.DI.get('module').foo;  
const foo = avito.DI.get('module');  
const foo = avito.DI.get('module').foo;  
const bar = avito.DI.get('module').foo;
```

```
const module = require('module');
```

```
import module from 'module';
```

```
import { foo as bar } from 'module';
```

```
import * as foo from 'module';
```

```
export { foo } from 'module';
```

```
export { foo as bar } from 'module';
```

```
const module = require('!loader?');

import module from '!loader?';

import { foo as bar } from '!loader?';

import * as foo from '!loader?';

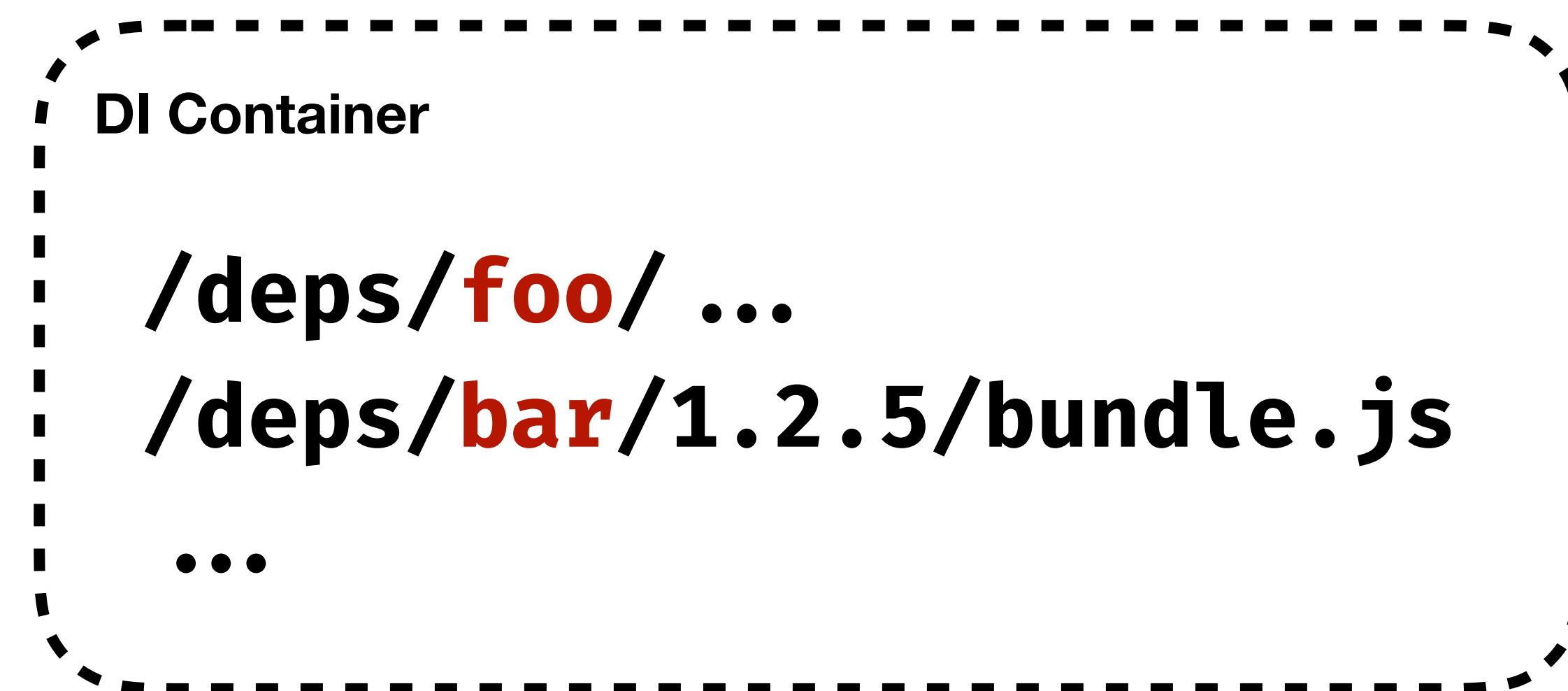
export { foo } from '!loader?';

export { foo as bar } from '!loader?';
```

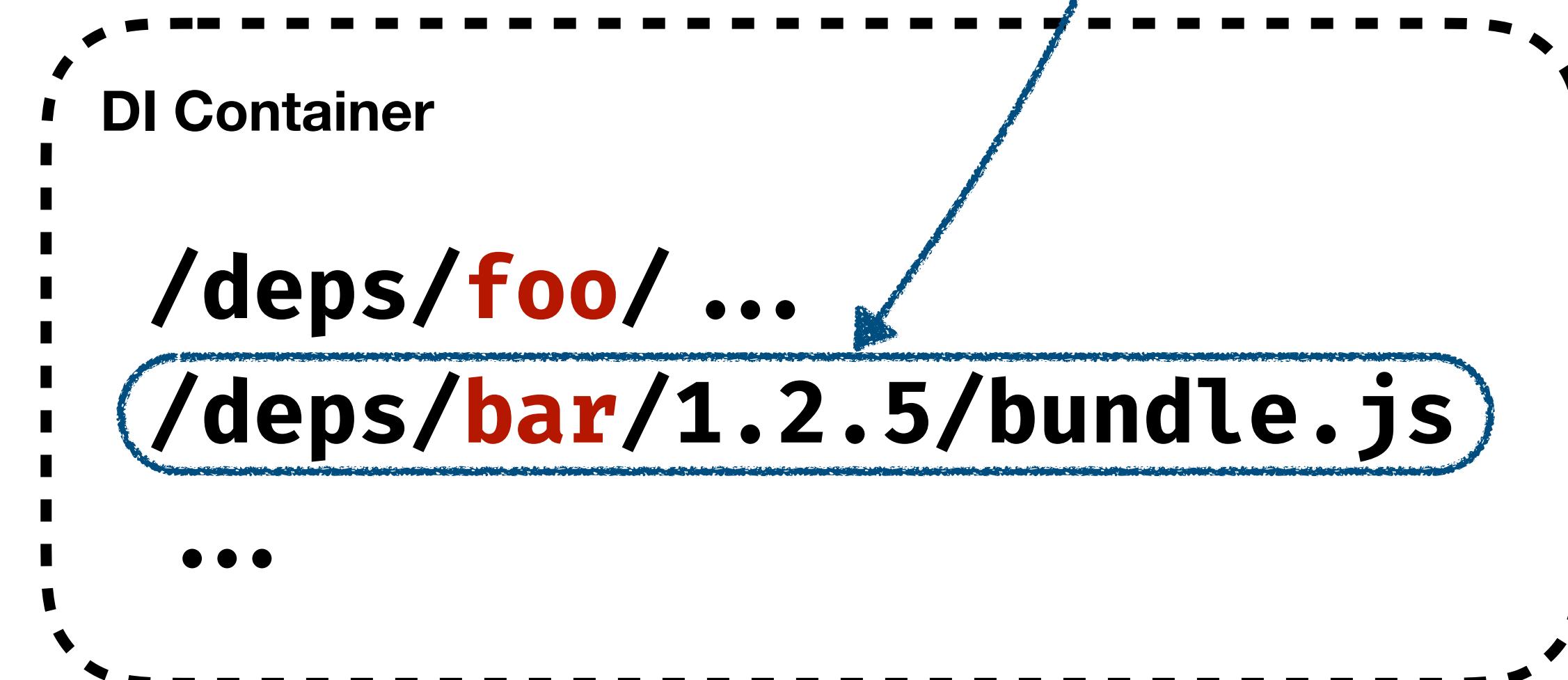
```
module.exports = DI.get('module', '^1.2.3');
```

```
import Foo from 'bar';  
import { Foo } from 'bar';  
import Baz from 'bar';  
  
DI.get('bar', '^1.2.3');
```

```
DI.get('bar', '^1.2.3');  
DI.get('bar', '^1.1.0');  
DI.get('bar', '~1.2.0');
```



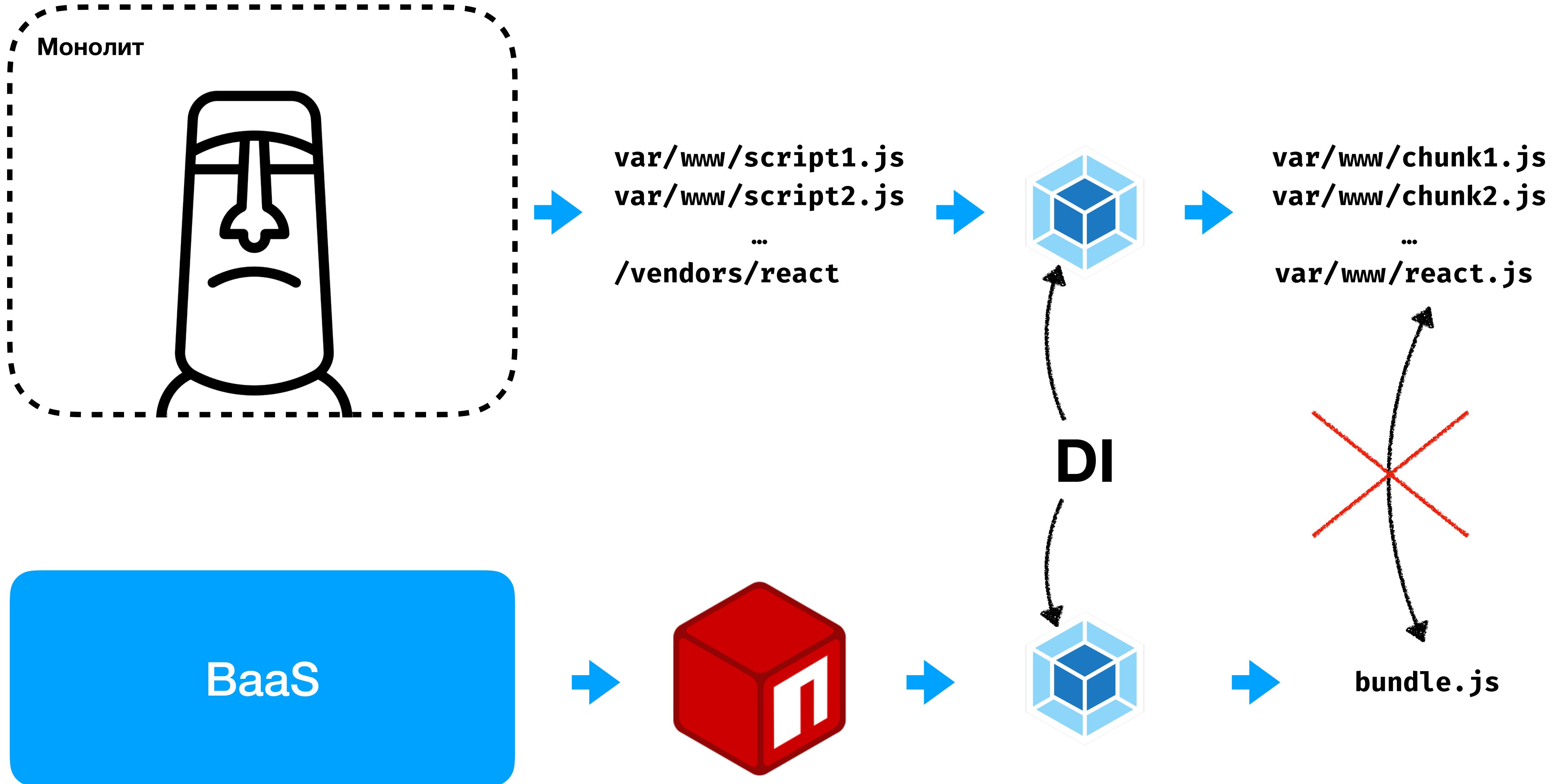
```
DI.get('bar', '^1.2.3');  
DI.get('bar', '^1.1.0');  
DI.get('bar', '~1.2.0');
```

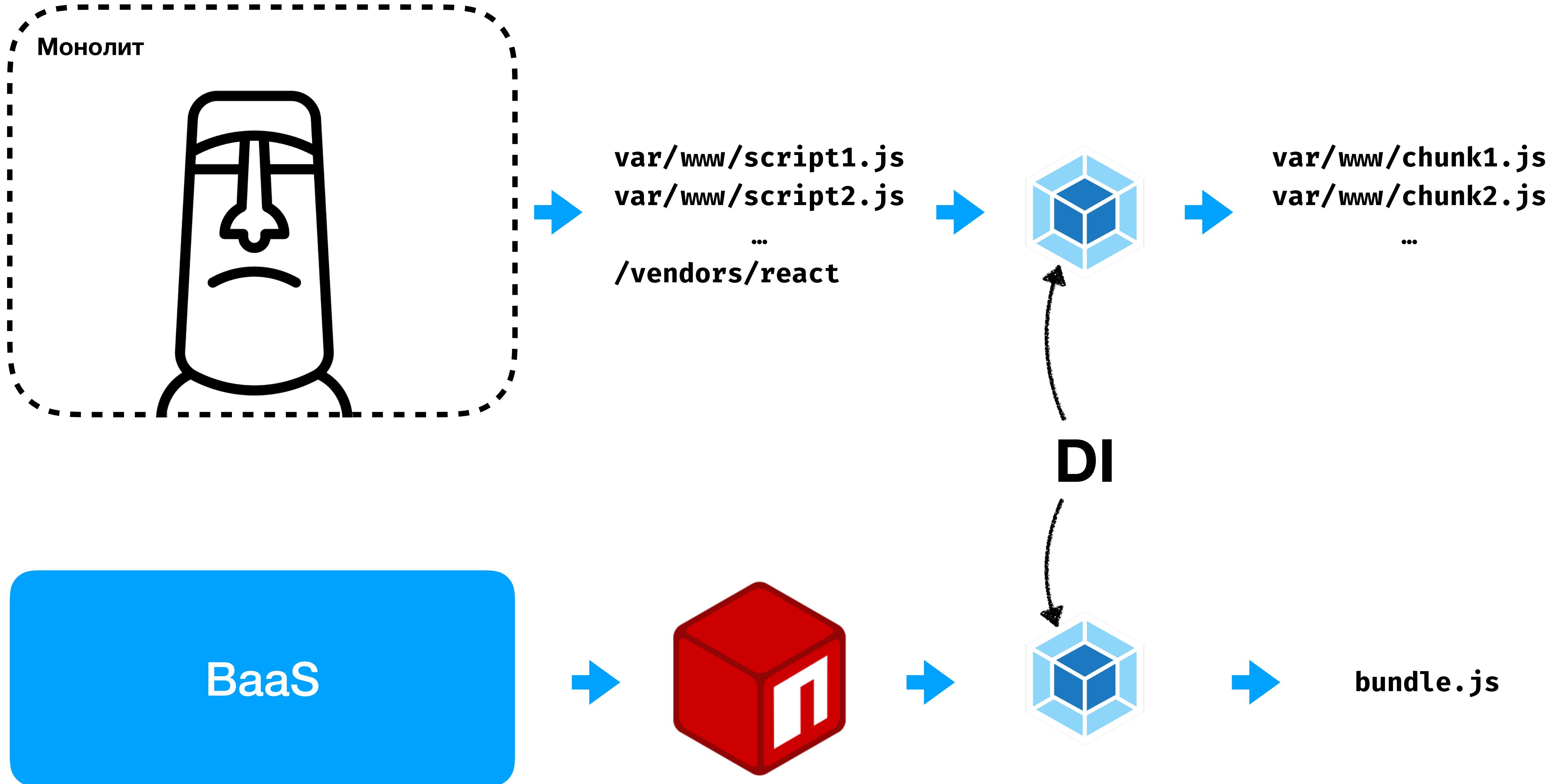


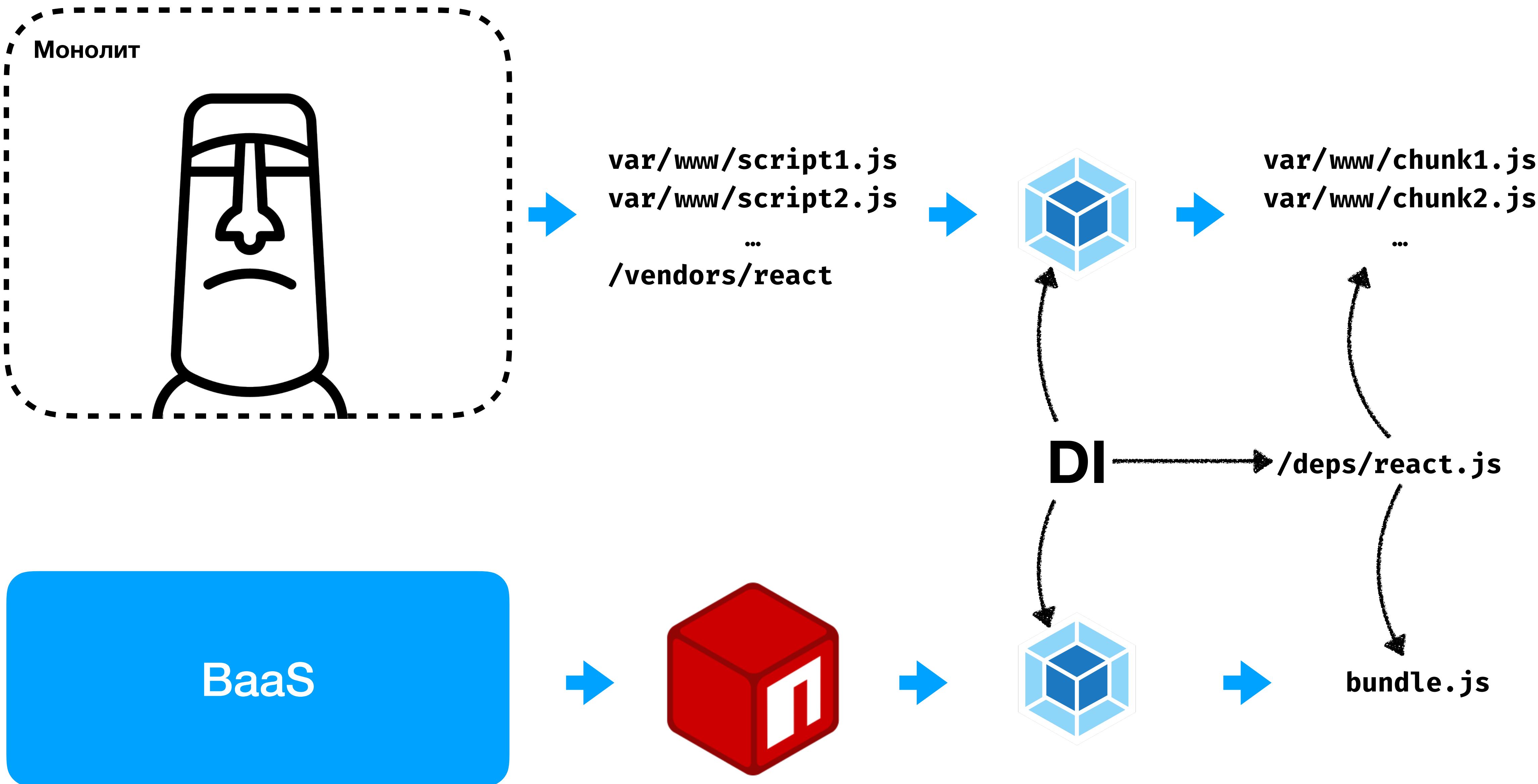
# DI

- Разработчику не нужно ничего менять в коде
- Зависимости автоматически переиспользуются между приложениями
- Количество зависимостей минимизируется через SemVer

**DI/DX**







# DI

- Разработчику не нужно ничего менять в коде
- Зависимости автоматически переиспользуются между приложениями
- Количество зависимостей минимизируется через SemVer

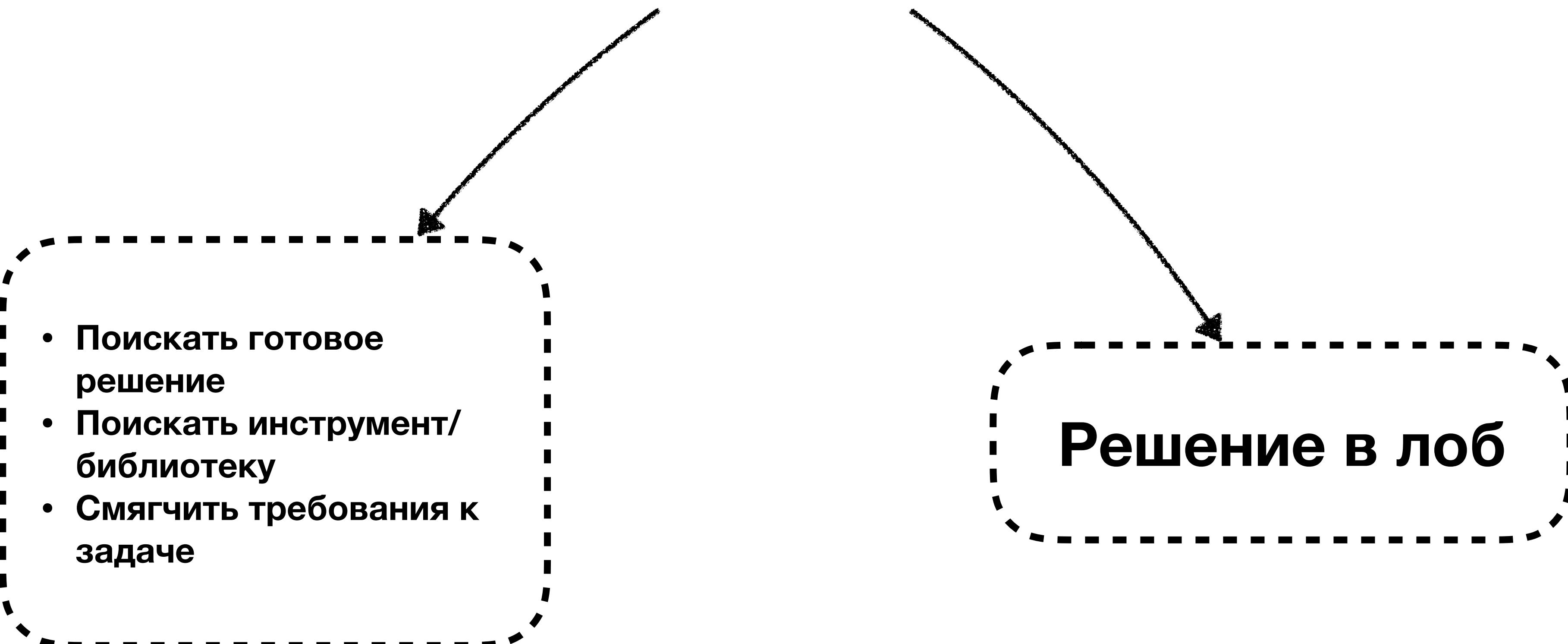
# DI

- Разработчику не нужно ничего менять в коде
- Зависимости автоматически переиспользуются между приложениями и хостовым окружением (**монолитом**)
- Количество зависимостей минимизируется через SemVer

# Подводим итоги

# Задача

# Задача



# Решение в лоб

Очевидное

Очень сложное



**Не стоит бояться дерзких идей. Пробуйте,  
экспериментируйте, ведь без экспериментов  
не бывает открытий!**

# Профиты для нас

- Построили базовую платформу для микрофронтендов
- Получили систему менеджмента зависимостей кода
- Улучшили перформанс
- Разблокировали возможности для мониторинга зависимостей в реальном времени
- Не создали никаких ограничений для разработчиков

**Получили бесценный опыт**

**И еще несколько безумных идей ;)**

Денис Колесников

Спасибо!

[dekolesnikov@avito.ru](mailto:dekolesnikov@avito.ru)

<https://facebook.com/DenKolesnikov>

<https://twitter.com/DenisKolesnikov>