

Яндекс

Яндекс

JSX? Пфф!

Владимир Гриненко
WSD, Москва, 28 января 2017

| Я не умею в React



OMG

React



Чего не хватает

- | Модификаторов
- | Уровней переопределения
- | (для любителей) Предметной области БЭМ



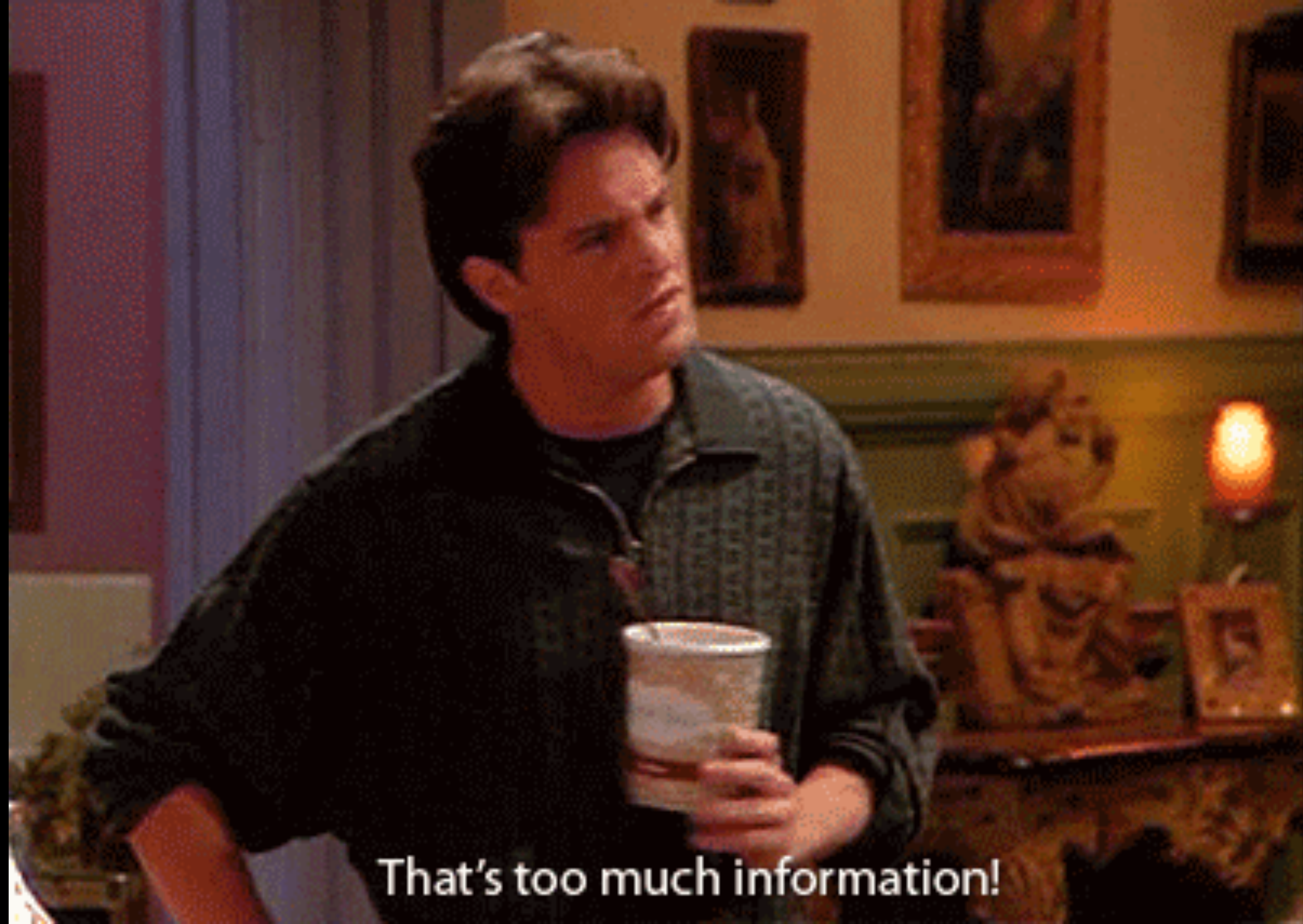
bem-react-core

Что это

- | Обертка над стандартными React-компонентами
- | Полностью совместима со стандартными компонентами
- | Не требует знания БЭМ
- | Не требует использования модификаторов
- | Не требует использования уровней переопределения

Что это

- | Автоматизирует генерацию классов
- | Позволяет использовать элементы, модификаторы и миксы
- | Позволяет использовать уровни переопределения
- | Оперирует компонентами, а не путями



That's too much information!

Пример

```
<div class='hello'>world</div>
```

Было

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

export default class Hello extends Component {
  render() {
    return (
      <div className='hello'>world</div>
    );
  }
}
```

Стало

```
import {decl} from 'bem-react-core';
```

```
export default decl({  
  block: 'hello',  
  content() {  
    return 'world';  
  }  
});
```

Lifecycle-методы

```
import decl from 'bem-react-core';
```

```
export default decl({  
  block: 'Hello',  
  willMount() {},  
  didMount() {},  
  willReceiveProps() {},  
  shouldUpdate() {},  
  willUpdate() {},  
  didUpdate() {},  
  willUnmount() {}  
});
```

Импорт

```
import Hello from 'b:hello';
```

```
ReactDOM.render(  
  <Hello />,  
  // ...  
);
```

Импорт

```
var Hello = [  
    require('path/to/hello/hello.js')  
][0].applyDecls();  
  
ReactDOM.render(<Hello />);
```

Траншпиллеры

- | webpack-bem-loader
- | babel-plugin-bem-import

Зачем все это?



Модификаторы

Модификаторы

```
.block1 {  
    color: red;  
}
```

Модификаторы

```
.block1 {  
    color: red;  
}  
  
.block1_modified {  
    color: green;  
    font-size: 24px;  
}
```

Модификаторы

```
.block1 {  
    color: red;  
}
```

```
.block1_modified {  
    color: green;  
    font-size: 24px;  
}
```

```
<div class="block1"></div>
```

```
<div class="block1 block1_modified"></div>
```

| Какие проблемы в
React?

Какие проблемы?

- | Модификации бывают опциональными
 - › отдельные файлы
 - › можно выразить через `class InputWithClear extends Input`

Какие проблемы?

- | Опциональные модификации
- | Композитные модификации
 - › пересечение всех вариантов не выражается через наследование
 - › Input, Input_with-clear, Input_with-icon

Модификатор: bem-react-core

```
// CheckBox_type_button.js
import {declMod} from 'bem-react-core';
import React from 'react';
import Button from 'b:Button';

export default declMod(({ type }) => type === 'button', {
  block: 'CheckBox',
  mods({ type }) {
    return { ...this.__base.apply(this, arguments), type };
  },
  content() { return <Button ... /> }
});
```

Модификатор: bem-react-core

```
import CheckBox from 'b:MyBlock m:type=button';
```

```
<CheckBox type="button">my check box</CheckBox>
```



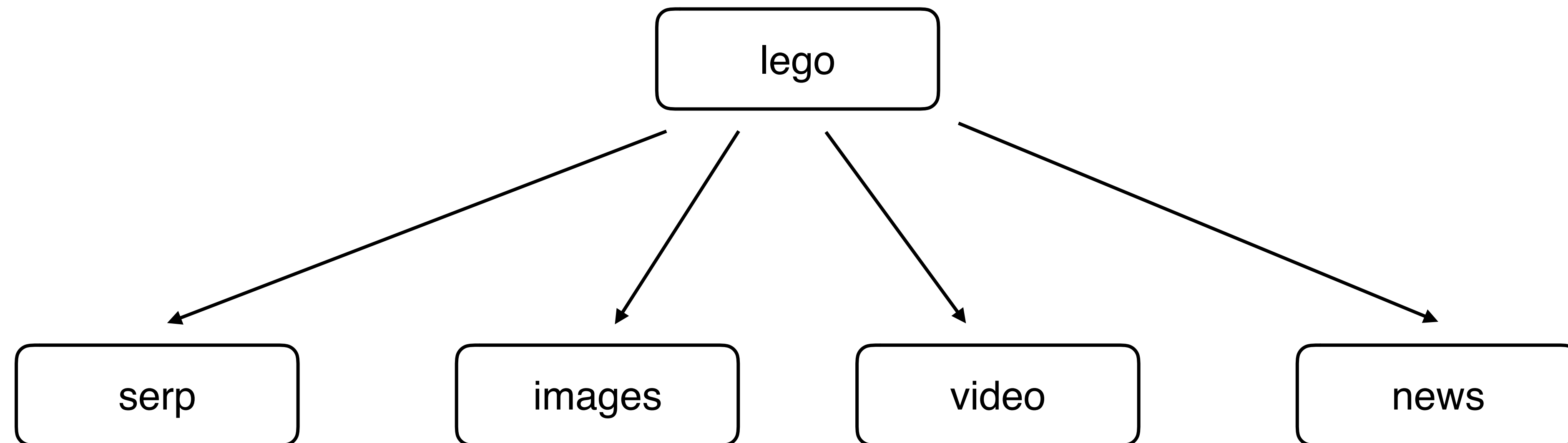
Уровни переопределения

Для чего?

- | Общая библиотека и проекты

Для чего?

| Общая библиотека и проекты

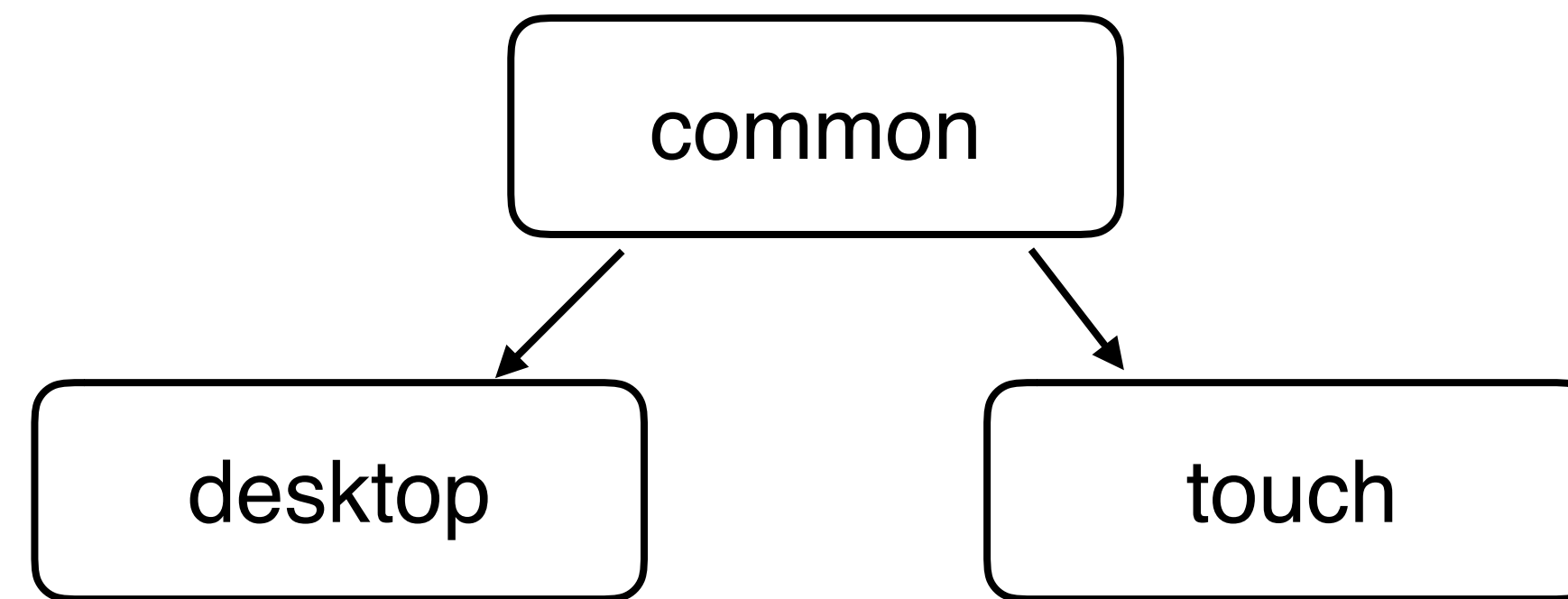


Для чего?

- | Общая библиотека и проекты
- | Платформы

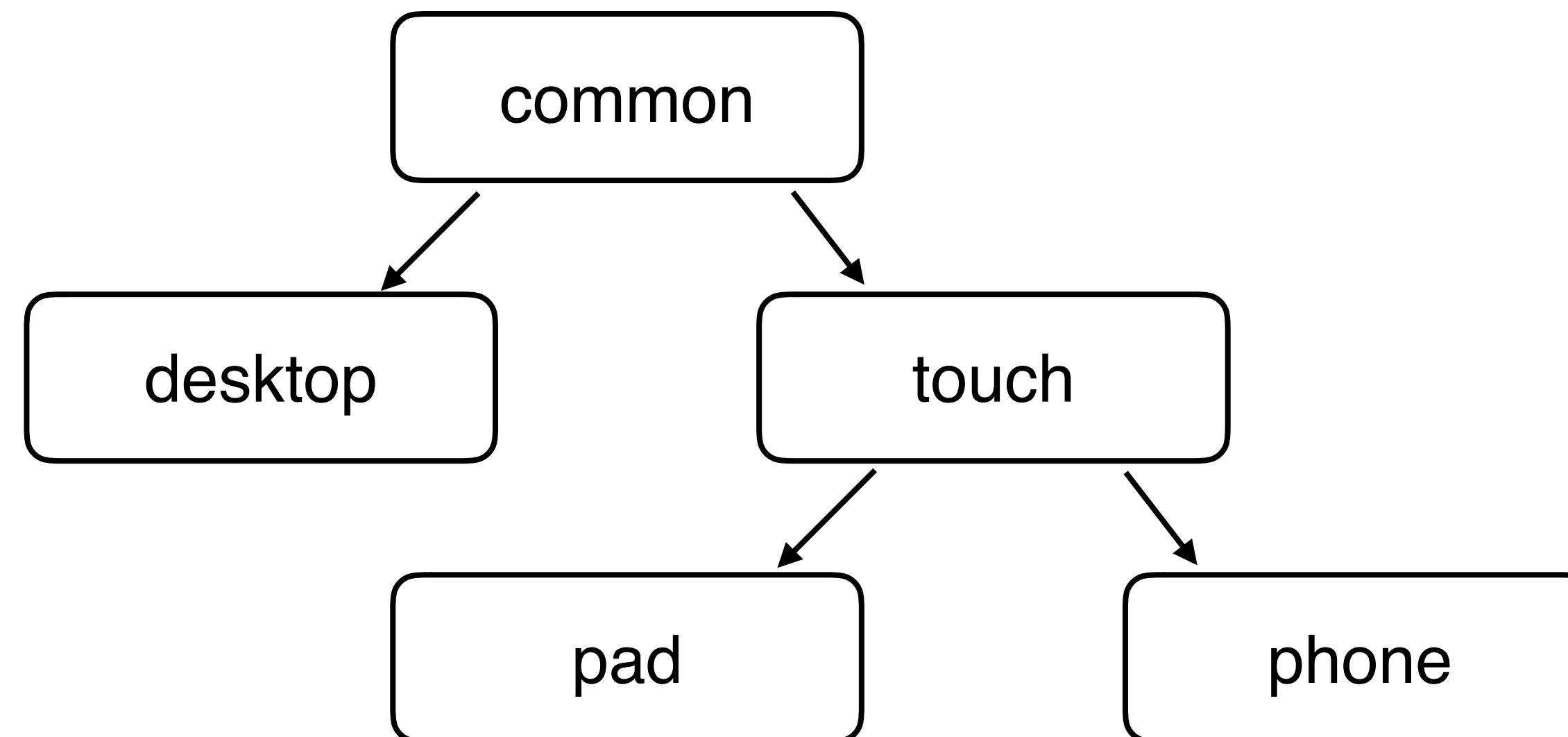
Для чего?

- | Общая библиотека и проекты
- | Платформы



Для чего?

- | Общая библиотека и проекты
- | Платформы

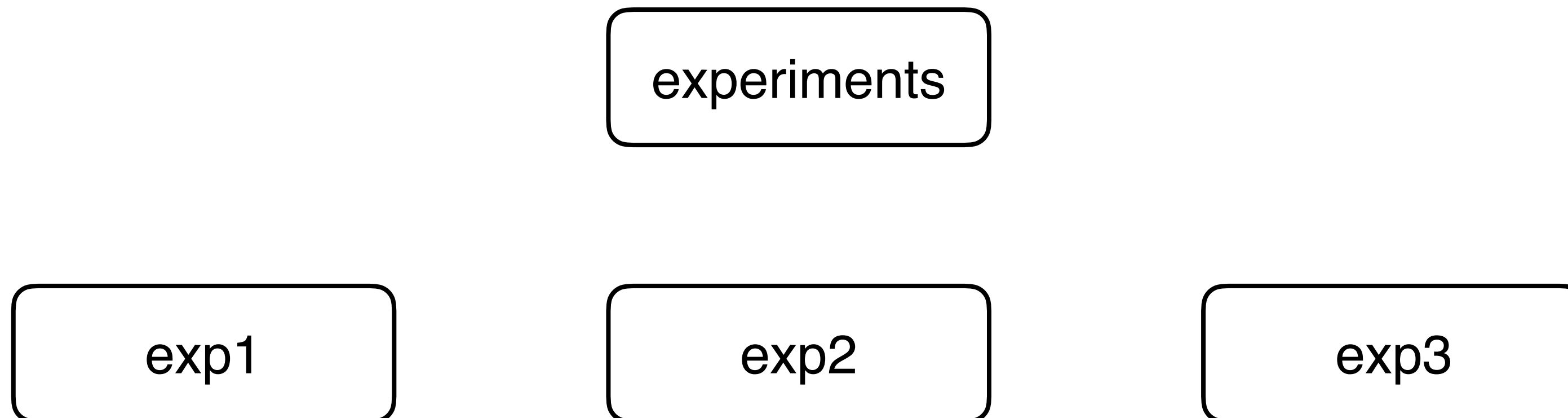


Для чего?

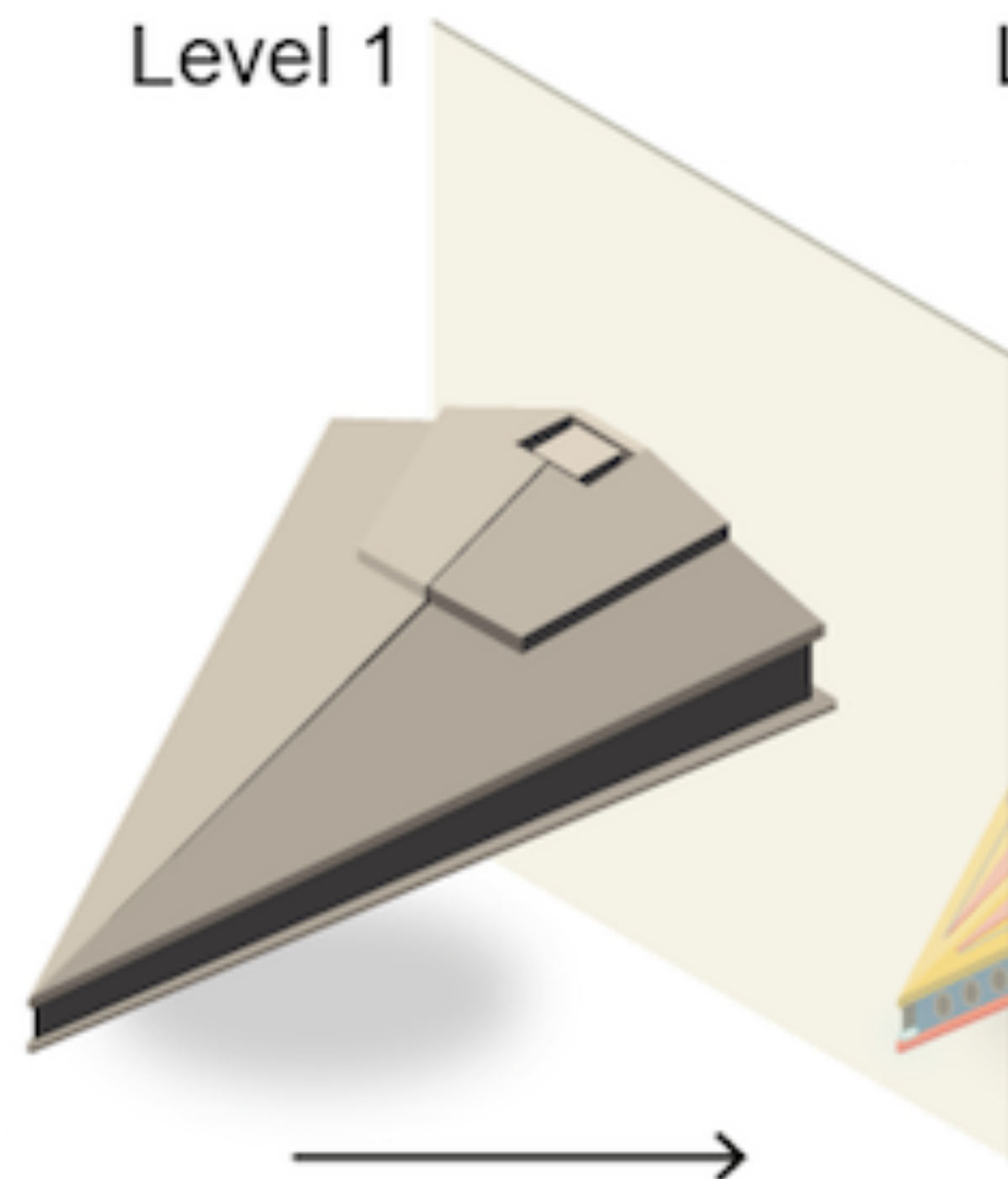
- | Общая библиотека и проекты
- | Платформы
- | Дизайн

Для чего?

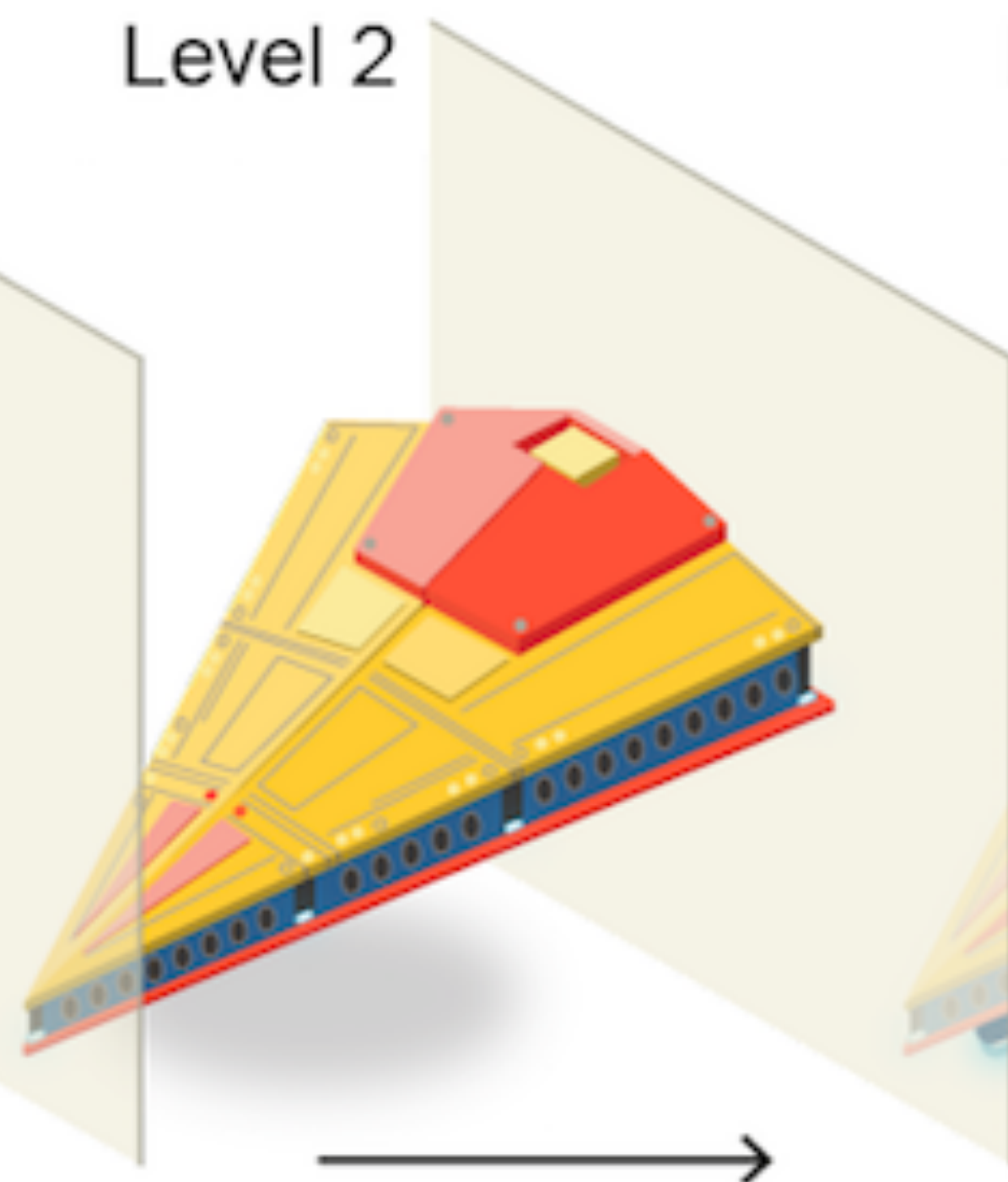
- | Общая библиотека и проекты
- | Платформы
- | Дизайн
- | Эксперименты



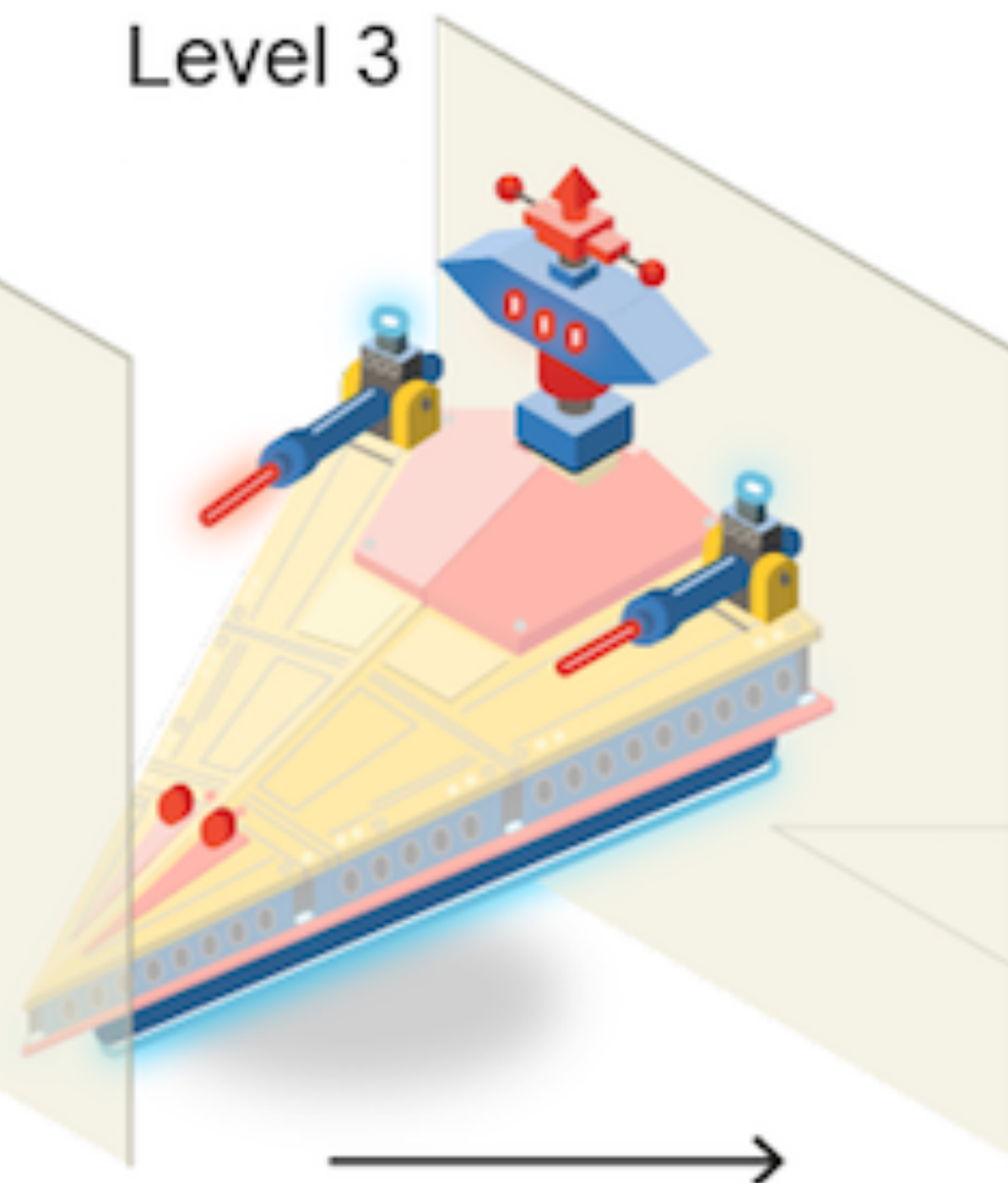
Level 1



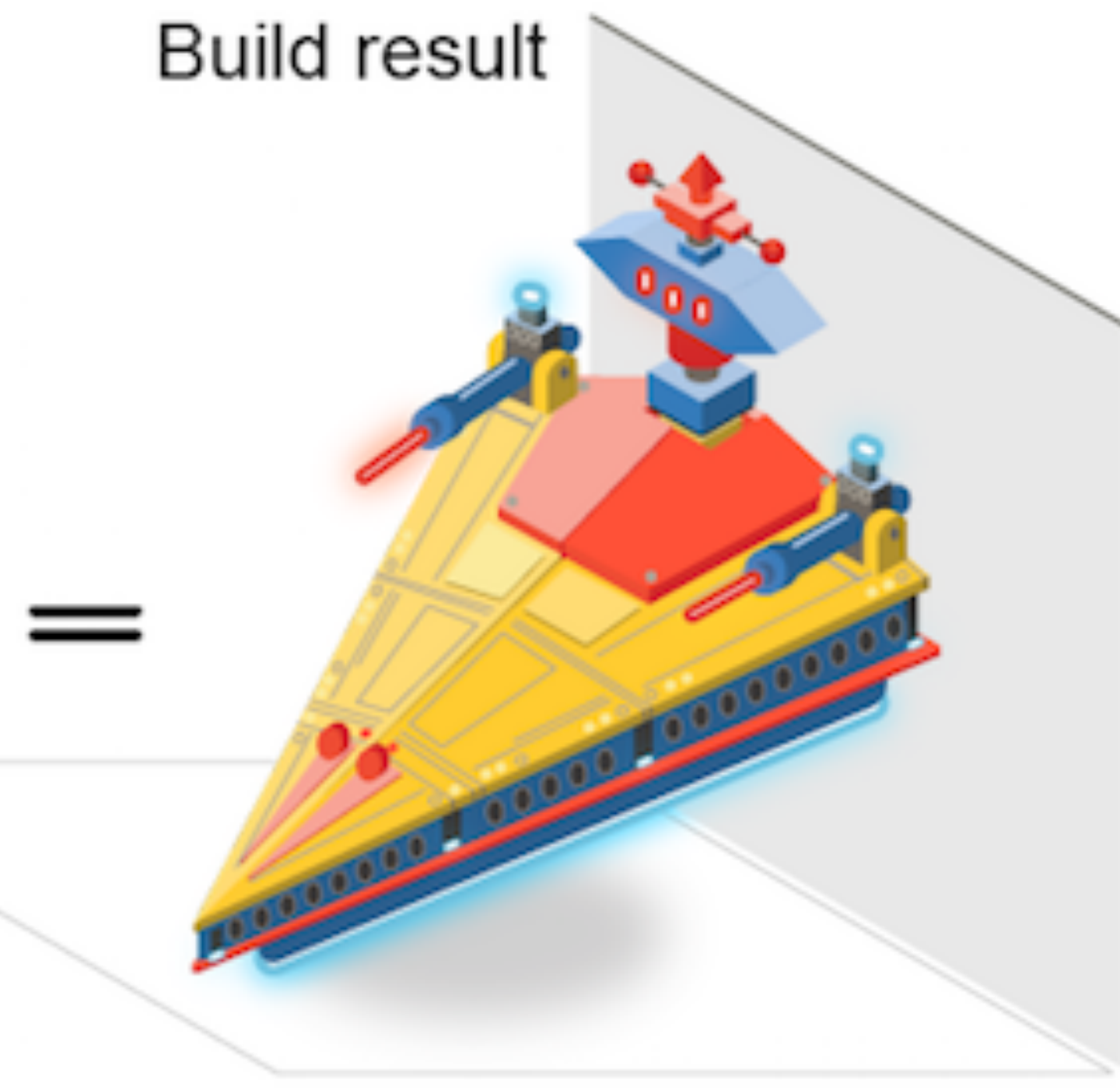
Level 2



Level 3



Build result



Как это работает

```
/* level1/b1/b1.css */  
.block1 {  
    color: red;  
}
```

```
/* level2/b1/b1.css */  
.block1 {  
    color: green;  
    font-size: 24px;  
}
```

Уровни переопределения

```
[  
    'path/to/level1',  
    'path/to/level2'  
]
```

Уровни переопределения

```
/* styles.css */
```

```
@import path/to/level1/block1/block1.css;
```

```
@import path/to/level2/block1/block1.css;
```

Уровень переопределения: bem-react-core

```
// blocks/Link/Link.js
import decl from 'bem-react-core';

export default decl({
  block: 'Link',
  tag: 'a',
  attrs({ url }) {
    return { href: url };
  }
});
```

<Link url="...">...

...

Уровень переопределения: bem-react-core

```
// blocks/Link/Link.js
// a11y/blocks/Link/Link.js
import decl from 'bem-react-core';

export default decl({
  block : 'Link',
  attrs() {
    return { ...this.__base.apply(this, arguments), role : 'link' };
  }
})
```

<Link url="...">...

...

#CBBTV





Do you see the light?



[github.com/awinogradov/bem-](https://github.com/awinogradov/bem-react-stub)
[react-stub](https://github.com/awinogradov/bem-react-stub)



github.com/bem/bem-xjst/

Итого



Итого

- | БЭМ не только для CSS!
- | Зачем нужно что-то кроме обычного React-компонента?
 - › местами меньше кода за счёт стандартного render()
 - › переопределения по маленьким составным частям
 1. модификаторы
 2. уровни переопределения
 - › БЭМ, если вы привыкли



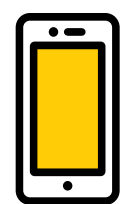


Work In Progress!

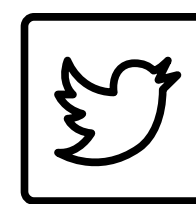
- | bem.info
- | github.com/bem/bem-react-core
- | github.com/bem/webpack-bem-loader
- | github.com/bem/bem-react-components



Контакты



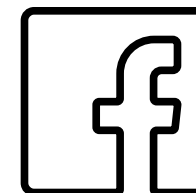
+7 (978) 096 88 87



@tadatuta



tadatuta@yandex-team.ru



i.tadatuta

