



ТИМ ЧАПТЫКОВ
vk.com/tc
[@chapykov](https://vk.com/@chapykov)



Куда растут библиотеки компонентов?

Тим Чаптыков

vk.com/tc

[@chaptykov](https://t.me/chaptykov)

Поле экспериментов


- VKUI (2017)
- VK Palette (2018)

Вопросы

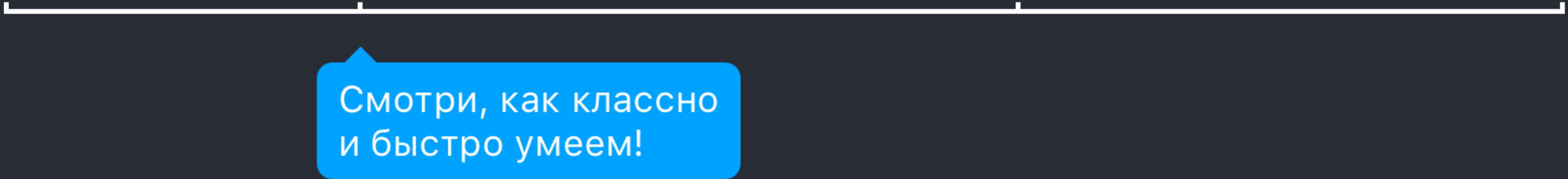
- Как сделать библиотеку компонентов хорошо?
- Как её развивать?
- Как сделать это дешево?




Жизненный цикл



Пишем с нуля.
Сделаем всё как надо.



Смотри, как классно
и быстро умеем!



Тут происходит
«жёсткая история»

A horizontal white line with three vertical tick marks is positioned in the lower half of the image. A blue callout box with a white border and a small triangle pointing upwards is located below the line, pointing to the second tick mark from the left. The text inside the box is white and reads 'Тут происходит «жёсткая история»'.



Не надо чинить, скоро
всё перепишем с нуля.

Начало

Простота



Универсальность



Гибкость



Надежность



Простота



Универсальность



Гибкость



Надежность



Через пол года

Простота



Универсальность



Простота



Универсальность



Простота



Универсальность



Простота



Универсальность



Простота



Универсальность



Простота



Универсальность



Простота



Универсальность



Простота

Универсальность



Простота

Универсальность



Простота

Универсальность



Простота

Универсальность



Простота

Универсальность



Гибкость

Надёжность





Тренд

Простота

Универсальность



Гибкость

Надёжность



Тренд

Универсальность важнее простоты

Надёжность важнее гибкости



Делаем сложно и дорого

А что будет, если покрутим иначе?



**Добавим простоты
и гибкости**

Добавим простоты

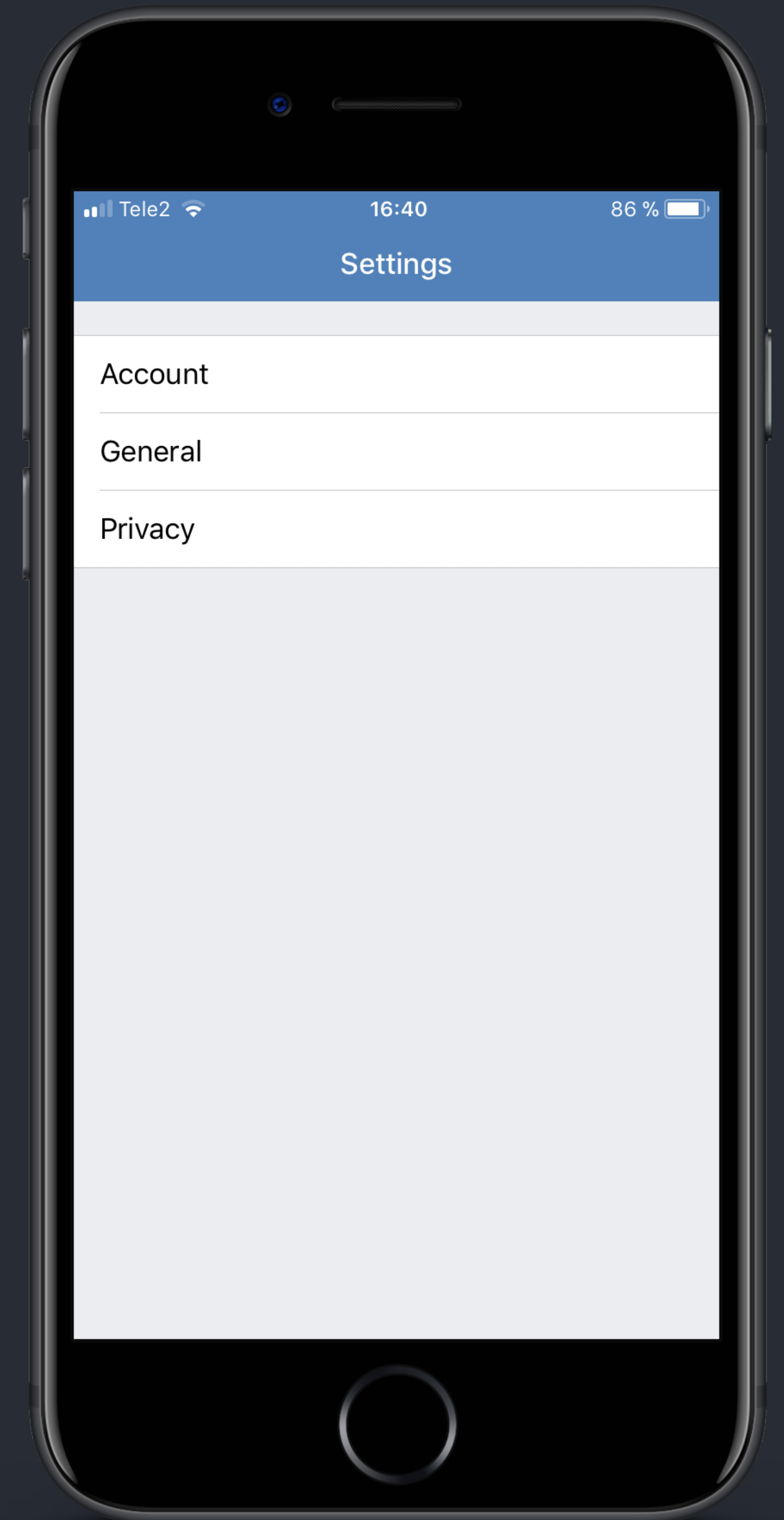
Декларативный интерфейс

HTML (1993)

```
<ul>  
  <li>Account</li>  
  <li>General</li>  
  <li>Privacy</li>  
</ul>
```

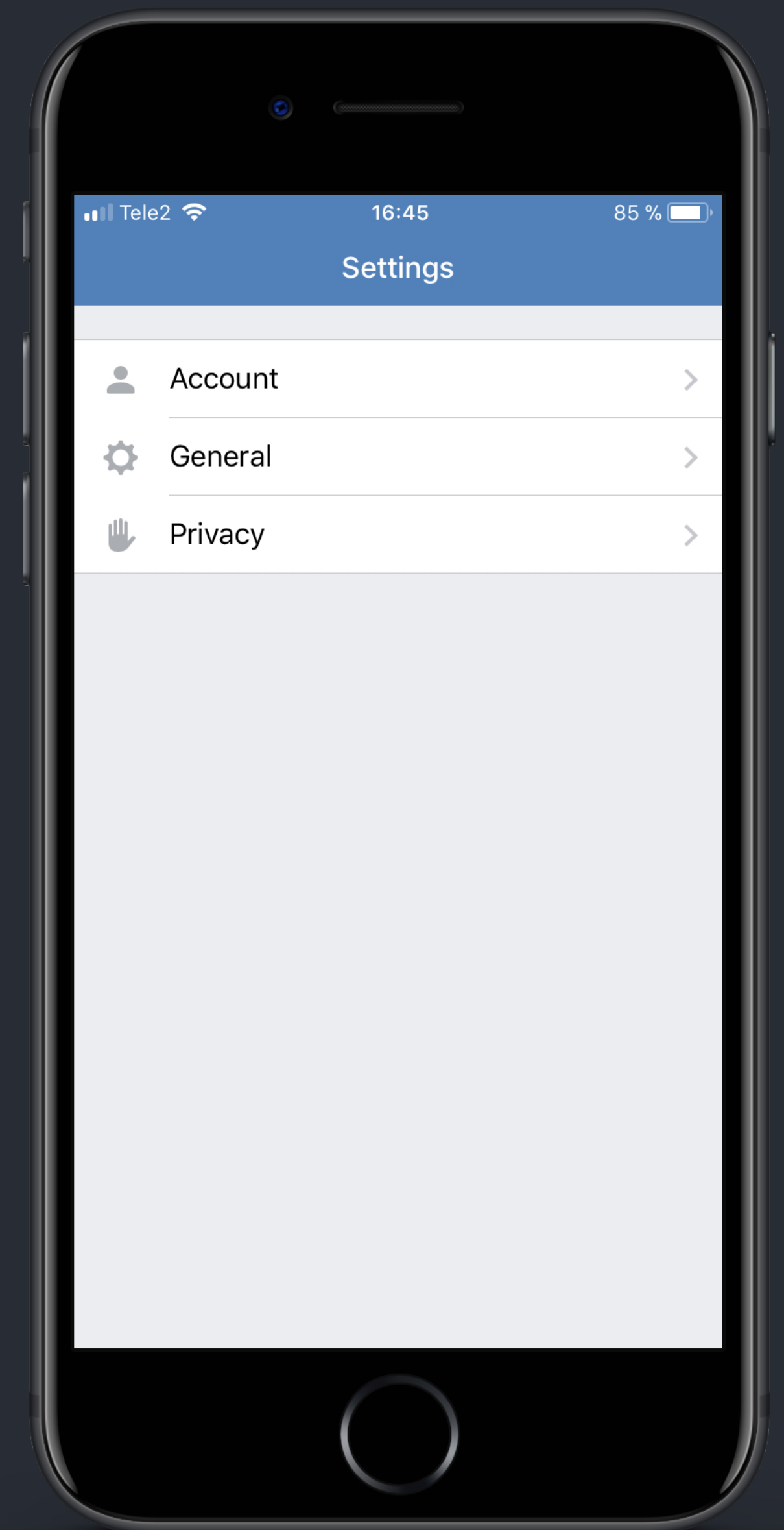
VKUI (2017)

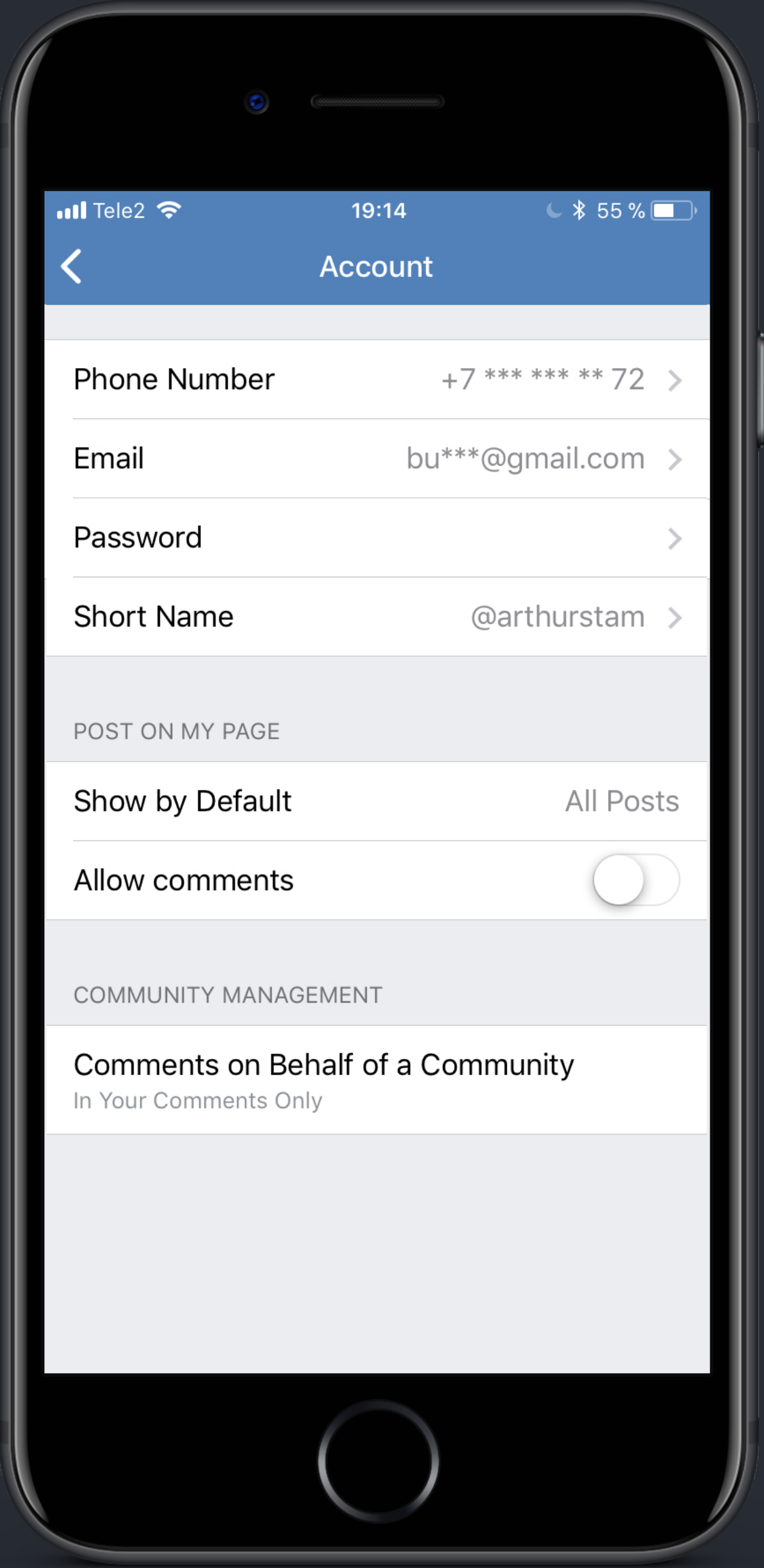
```
<List>
  <ListItem>Account</ListItem>
  <ListItem>General</ListItem>
  <ListItem>Privacy</ListItem>
</List>
```



Add props

```
<ListItem  
  expandable={true}  
  onClick={expandSomething}  
  icon={userIcon}  
>  
  Account  
</ListItem>
```





Account

- Phone Number +7 *** ** 72 >
- Email bu***@gmail.com >
- Password >
- Short Name @arthurstam >

POST ON MY PAGE

- Show by Default All Posts
- Allow comments ☐

COMMUNITY MANAGEMENT

- Comments on Behalf of a Community
In Your Comments Only






VK Palette (2018)

```
<List>  
  <ListItem>Account</ListItem>  
  <ListItem>General</ListItem>  
  <ListItem>Privacy</ListItem>  
</List>
```

Назад

Управление


✕

	Могут приглашать участников в беседу	Все участники
	Могут редактировать информацию беседы	Все участники
	Могут менять закреплённое сообщение	Все участники
	Могут добавлять администраторов	Только создатель
	Тип чата	Служебный

Отмена


Сохранить


Информация




Пони и радуга

3 участника


 Показать вложения


 Ссылка на беседу


 Управление

Все участники 3

Администраторы 1




 Добавить участников



Тимофей Чаптыков


был в сети 13 минут назад

Создатель беседы



Амина Омарова

была в сети 35 минут назад



Катерина Тарасова

online

Покинуть беседу

Добавим гибкости

- Разработка сложных фиच становится проще
- Вероятность выстрелить в ногу выше

Добавим гибкости

- Разработка сложных фиच становится проще
- Вероятность выстрелить в ногу выше



Насколько это страшно?

Развязываем руки



Дизайнер снова сделал
двадцать пикселей
вместо десяти...



Введём жёсткую систему с шагом в четыре пикселя.

Будем проводить воспитательную беседу с теми, кто от неё отступит.

А потом напишем препуш-хук, чтобы в мастер не попадало ни одно число, которое не делится на четыре.

А МОЖНО ТАК

```
<div style={{ marginBottom: 10 }}>  
  <Button>Account</Button>  
</div>  
<Button>General</Button>
```

Или так

```
<Button style={{ marginBottom: 10 }}>  
  Account  
</Button>  
<Button>General</Button>
```

Чаще всего так

```
<Button className="MyBlockName">  
  Account  
</Button>  
<Button>General</Button>
```


Кастомизация

- className
- style

Решает большинство задач кастомизации компонентов

**Разработчик лучше знает
свою проблему**

className и style

—надёжность +гибкость

Rest props

Пример

```
<input  
  onChange={this.props.onChange}  
  value={this.props.value}  
>
```

Понадобился blur...

```
<input  
  onChange={this.props.onChange}  
  onBlur={this.props.onBlur}  
  value={this.props.value}  
>
```

...и focus

```
<input  
  onChange={this.props.onChange}  
  onBlur={this.props.onBlur}  
  onFocus={this.props.onFocus}  
  value={this.props.value}  
>
```

А как же disabled?

```
<input  
  onChange={this.props.onChange}  
  onBlur={this.props.onBlur}  
  onFocus={this.props.onFocus}  
  value={this.props.value}  
  disabled={this.props.disabled}  
>
```


Спустя месяц...

```
<input
  onChange={this.props.onChange}
  onBlur={this.props.onBlur}
  onFocus={this.props.onFocus}
  onKeyDown={this.props.onKeyDown}
  onKeyUp={this.props.onKeyUp}
  value={this.props.value}
  disabled={this.props.disabled}
  readOnly={this.props.readOnly}
  style={this.props.style}
  className={this.props.className}
  id={this.props.id}
  placeholder={this.props.placeholder}
/>
```

И где-то рядом PropTypes

```
static propTypes = {  
  onChange: PropTypes.func,  
  onBlur: PropTypes.func,  
  onFocus: PropTypes.func,  
  onKeyDown: PropTypes.func,  
  onKeyUp: PropTypes.func,  
  value: PropTypes.string,  
  disabled: PropTypes.bool,  
  readonly: PropTypes.bool,  
  style: PropTypes.object,  
  className: PropTypes.string,  
  id: PropTypes.string,  
  placeholder: PropTypes.string  
};
```



Их еще достаточно много

...restProps

```
const { onChange, ...restProps } = this.props;

return (
  <input
    {...restProps}
    onChange={this.onChange}
  />
);
```

...restProps

—надёжность +гибкость

Props / Children

Props / Children

```
<Button label="Press me" />
```

// VS

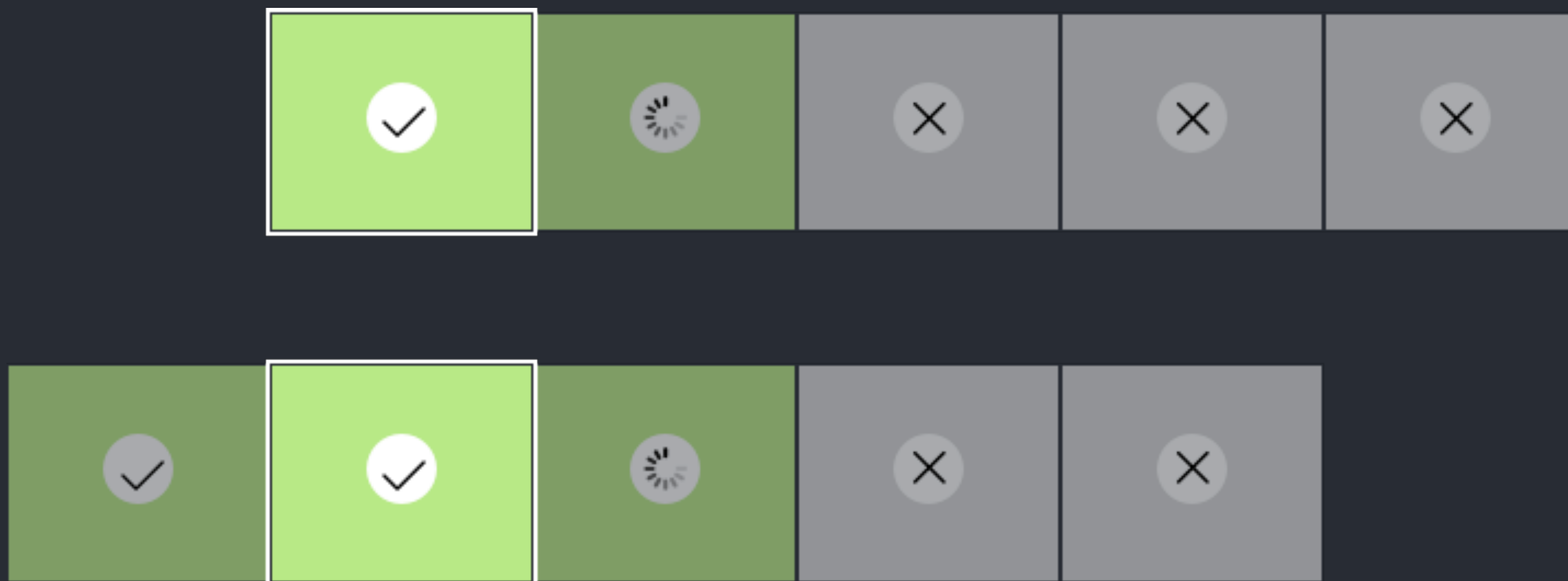
```
<Button>Press me</Button>
```

Props

```
<Gallery  
  data={ [  
    { source: 'image1.jpg', altText: 'Image 1' },  
    { source: 'image2.jpg', altText: 'Image 2' },  
    { source: 'image3.jpg', altText: 'Image 3' }  
  ] }  
>
```


Children

```
<Gallery>  
    
    
    
</Gallery>
```



**Контроль загрузки
и оптимизация?**

Приводим к массиву

```
React.Children.toArray(props.children);
```

```
// OR
```

```
[].concat(props.children);
```

Показываем только видимые

```
children.filter(isVisible)
```

Гибкость

```
<Gallery>
  <div>
    <h1>Hello, World!</h1>
  </div>
  
  
</Gallery>
```

Страшно?



Они же там
такого наворотят

Это как раз то, что нужно

Доверяйте коллегам

Добавим простоты и гибкости

- Несколько простых компонентов лучше одного универсального
- Избегаем повторений за счет переиспользования простых компонентов

Button + loader

Кнопка со статусом загрузки

Обычное
состояние

Загрузка



Сохраняет
ширину

Интерфейс компонента

```
<Button loading={true}>  
  Кнопка  
</Button>
```

Добавим новый проп?

—простота +универсальность

Создадим новый компонент

```
ButtonWithProgress.propTypes = {  
  /**  
   * Наследуем свойства кнопки  
   */  
  ...Button.propTypes,  
  
  /**  
   * Показать индикатор загрузки  
   */  
  loading: PropTypes.bool,  
};
```

СОЗДАДИМ НОВЫЙ КОМПОНЕНТ

```
function ButtonWithProgress(props) {  
  const { className, loading, children, ...nativeProps } = props;  
  const classnames = cx('ButtonWithProgress', className, {  
    'ButtonWithProgress--loading': loading  
  });  
  
  return (  
    <Button {...nativeProps} className={classnames}>  
      <span className="ButtonWithProgress__content">{children}</span>  
      {loading && (  
        <Progress  
          inverted={props.appearance === 'primary'}  
          className="ButtonWithProgress__progress"  
        />  
      )}  
    </Button>  
  );  
}
```


СОЗДАДИМ НОВЫЙ КОМПОНЕНТ

```
.ButtonWithProgress {  
    position: relative;  
}  
.ButtonWithProgress--loading .ButtonWithProgress__content {  
    color: transparent;  
}  
.ButtonWithProgress__progress {  
    position: absolute;  
    top: 0;  
    right: 0;  
    bottom: 0;  
    left: 0;  
    margin: auto;  
}
```

Результат

- Фича есть
- 2 простых компонента вместо 1 сложного
- Переиспользование кода
- Размер бандла

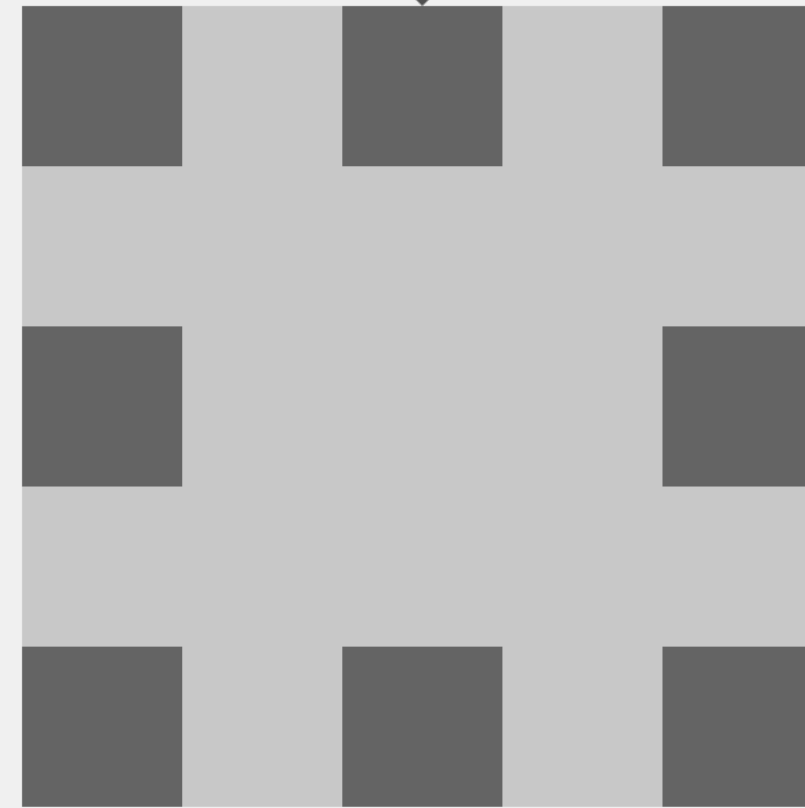
Разбиение на маленькие компоненты

—универсальность +простота

Tooltip

Простой тултип

position: top, align: center



Интерфейс компонента

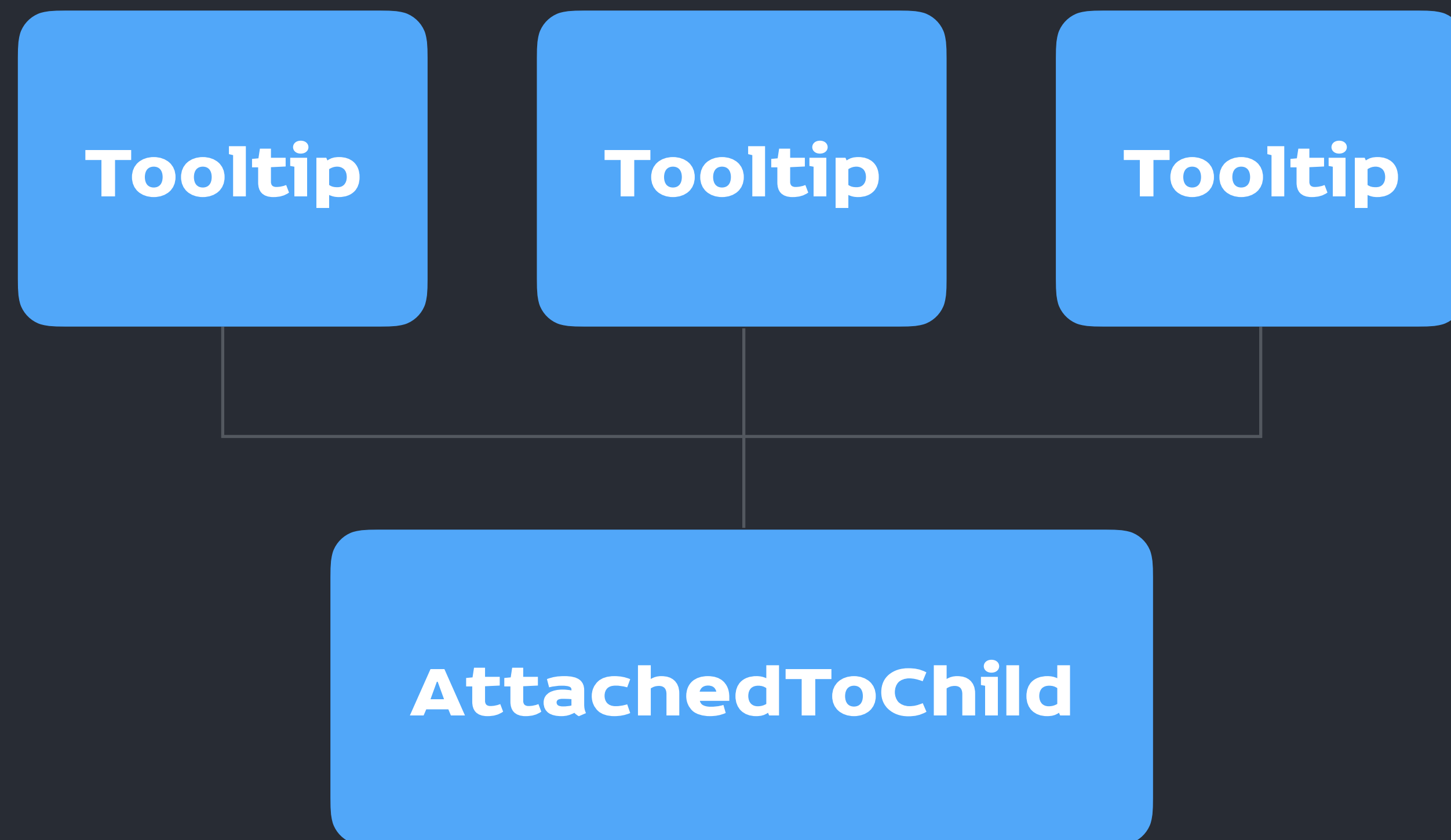
```
<Tooltip text="Text" position="t" align="right">  
  <SomeContent />  
</Tooltip>
```

ВОЗМОЖНОСТИ

- Отрисовывается в портале в body
- Позиционируется к элементу
- 8 вариантов выравнивания

Если хочется сложнее

- Другие варианты оформления
- Форматирование, сложный контент
- Другая логика появления и исчезновения



Не дублирование, а конкуренция



Заключение



 **YAMAHA**

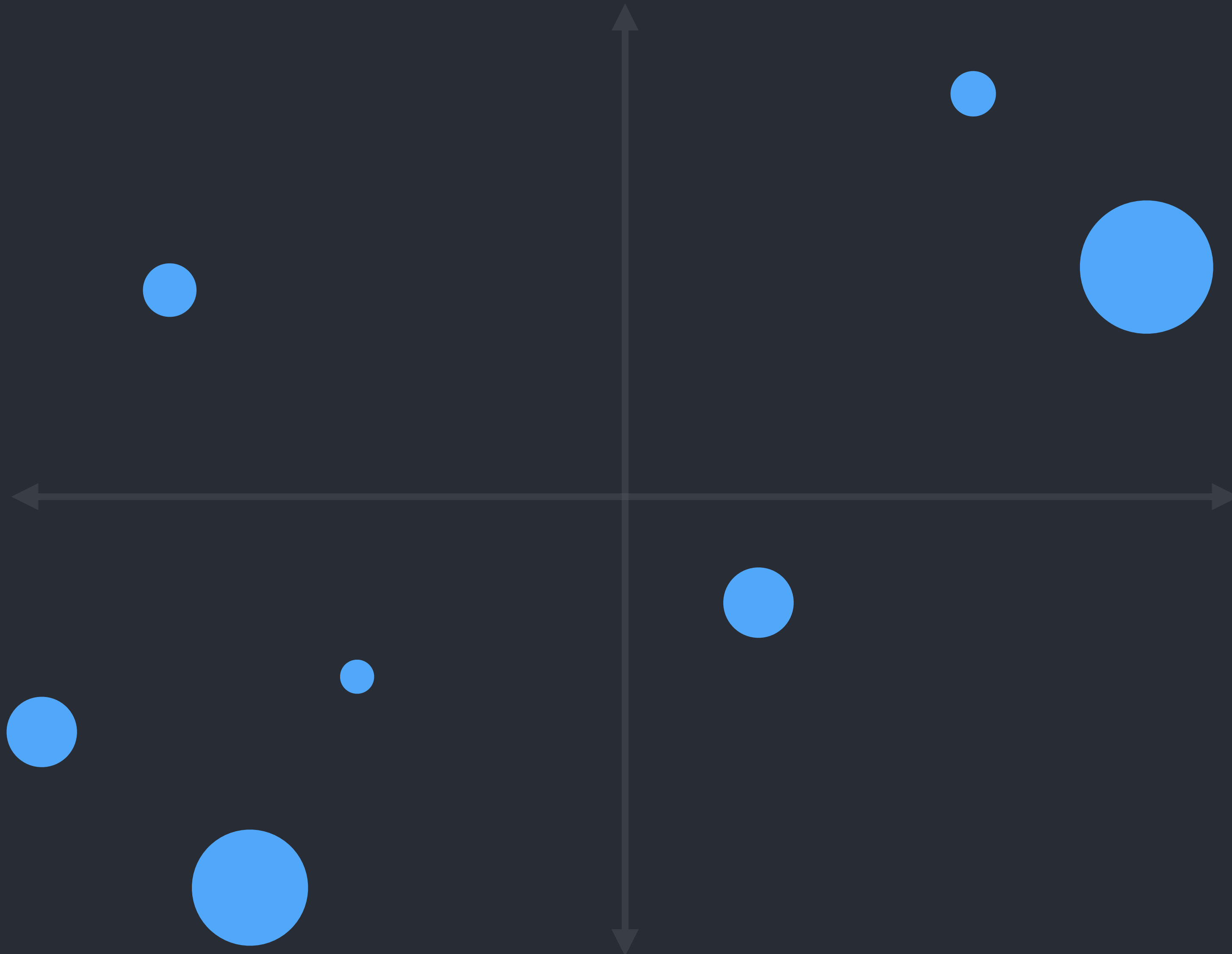
**Разные компоненты могут быть
настроены по-разному**

Универсальность

Гибкость

Надёжность

Простота



Универсальность

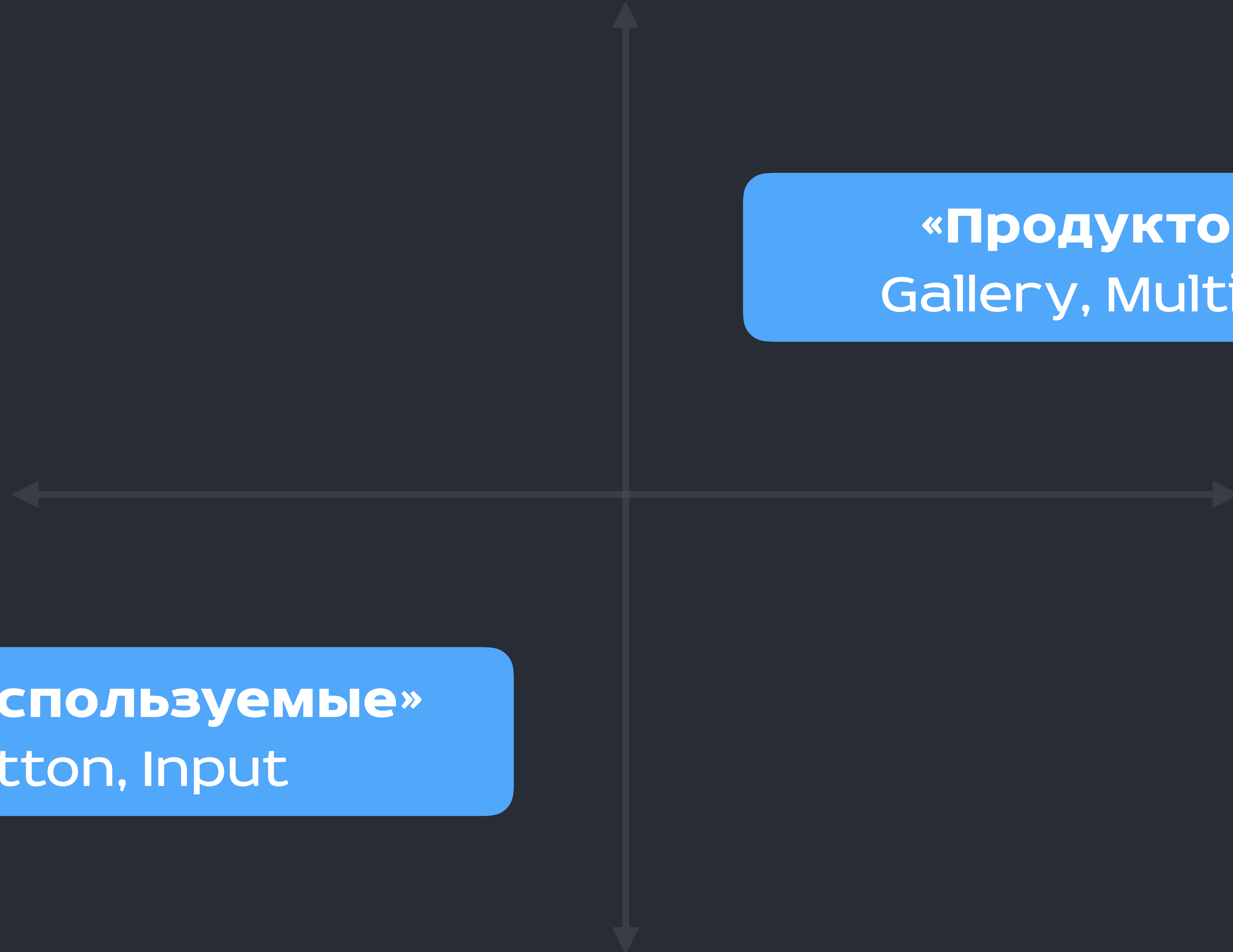
«Продуктовые»
Gallery, Multiselect

Гибкость

Надёжность

«Переиспользуемые»
Button, Input

Простота



Начните отсюда



ТИМ ЧАПТЫКОВ
vk.com/tc
[@chaptykov](https://t.me/chaptykov)