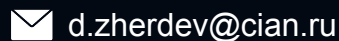
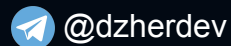
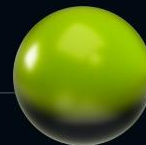


# BDUI: удовольствие или боль?



Дмитрий  
Жердев  
Циан



**Mobius**  
2023 Autumn





**BDUI**

Удовольствие или Боль?

A police officer in a dark uniform and cap is shouting with his mouth wide open. He is holding a silver badge in his right hand. The background is a solid blue color with several large, glossy blue spheres. A white speech bubble is positioned to the left of the officer, containing the text "My name is...".

My name is...

# Дмитрий Жердев

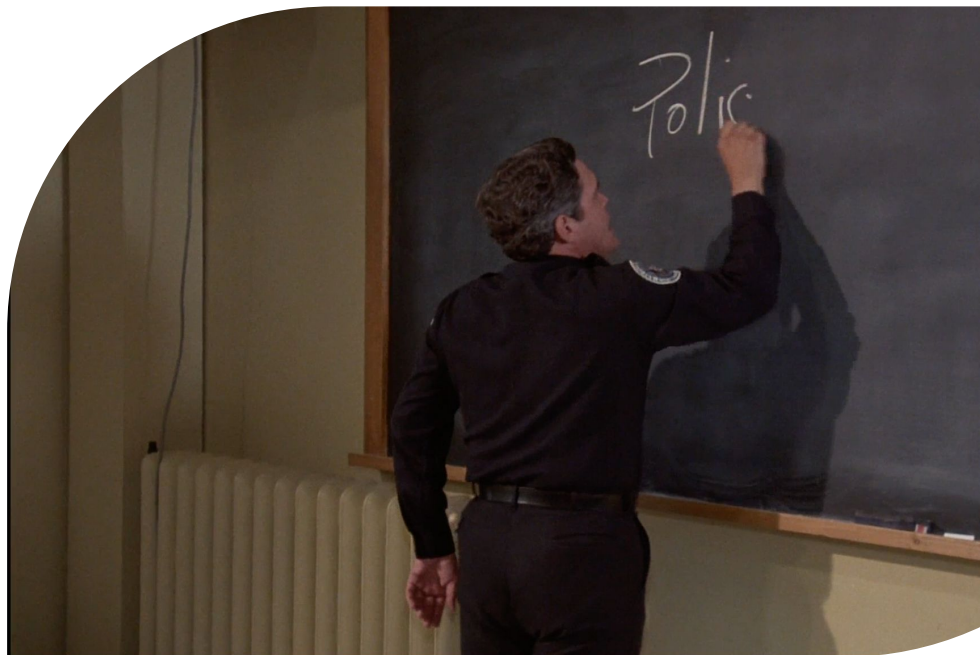
Mail: [d.zherdev@cian.ru](mailto:d.zherdev@cian.ru)

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



# Сегодня мы выясним

1. Что такое BDUI?
2. Как мы до такого докатились?
3. Как это работает? Что умеет?
4. Для кого сделано?
5. Что болело?
6. Понравилось ли нам?
7. Планируем ли продолжать?

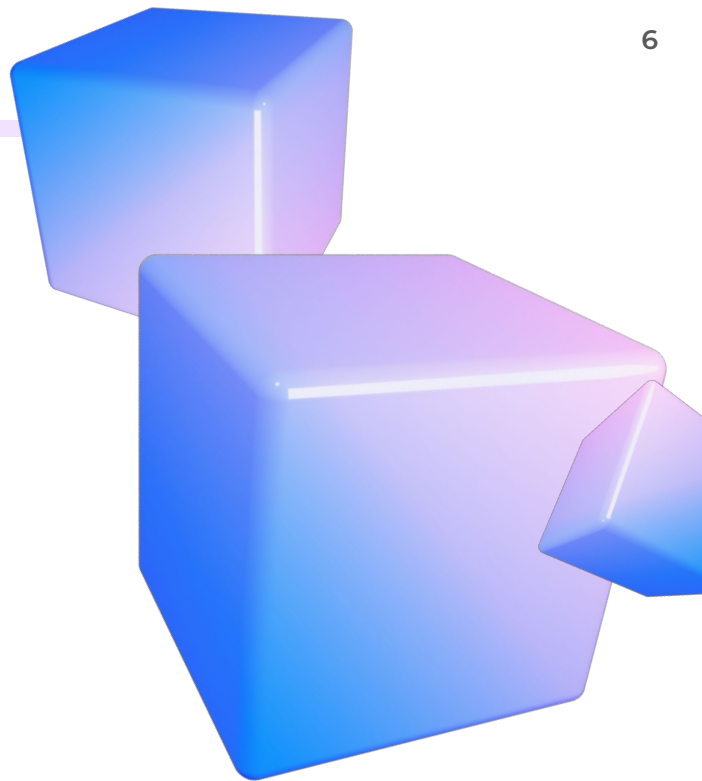


# Что такое BDUI?

**BDUI** — концепция, при которой:

- **сервер** управляет не только данными в приложении, но и формирует экраны, верстку, реакцию на взаимодействия и переходы
- **клиент** умеет рендерить

Позволяет обновлять поведение фичей на нескольких платформах сразу без релизного цикла.

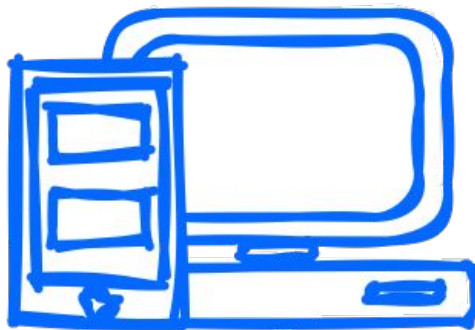


# Ценность BDUI

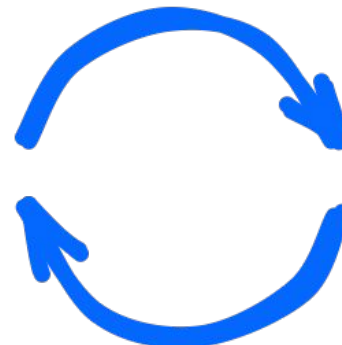
---



Снижение Time  
to Market



Создание UI  
для нескольких  
платформ



Обновление фичи без  
обновления  
приложений

# А меня-то зачем слушать?

Узнать ещё один реальный опыт и его нюансы

В нашей реализации есть нестандартные идеи





Как мы пришли к VDUI?



## Крупная PropTech компания

● **12**  
продуктовых  
направлений

● **1000+**  
сотрудников

Приложение Циан — сервис  
по покупке, продаже и аренде  
недвижимости.

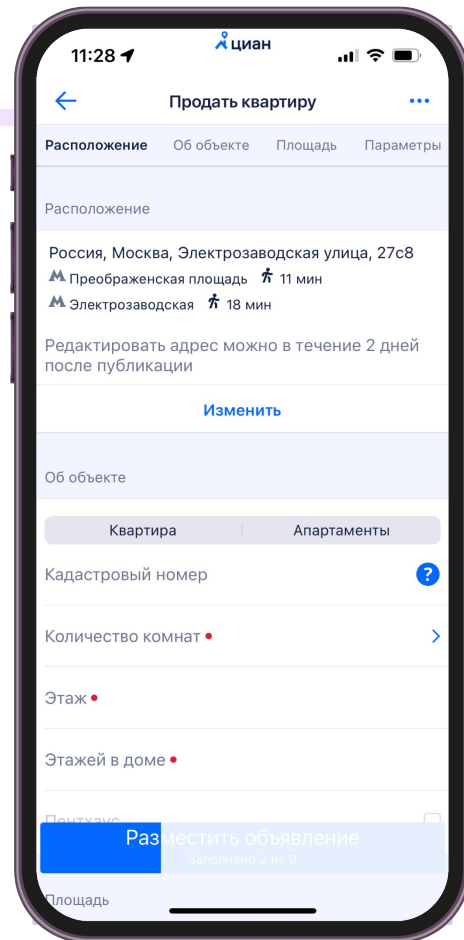
● **1,9 млн**  
актуальных объявлений  
ежемесячно

● **19 млн**  
уникальных пользователей  
ежемесячно

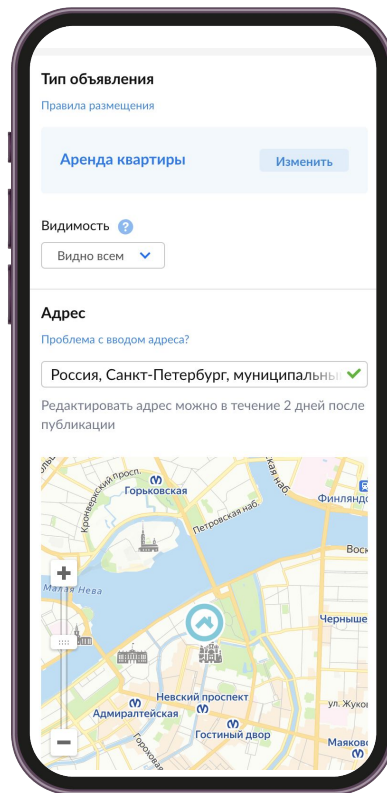
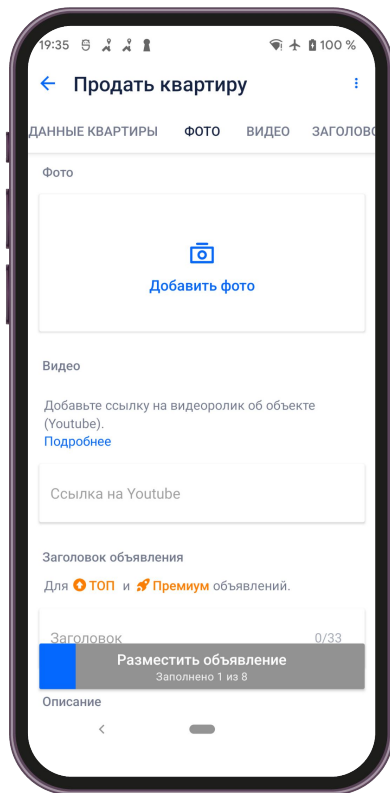
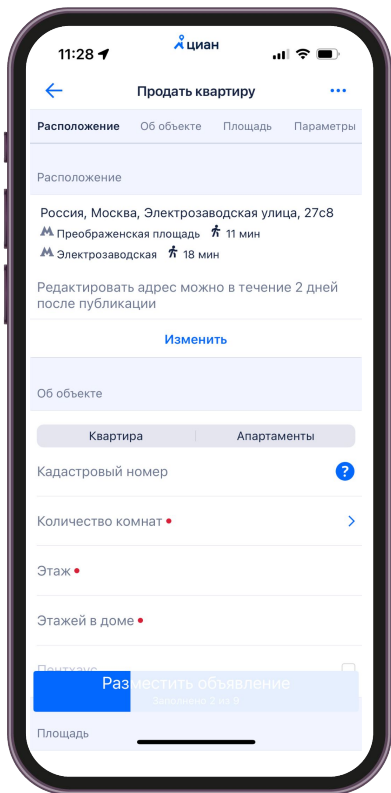
# Форма подачи объявления

**Конверсия в публикацию** —  
ключевая продуктовая  
метрика.

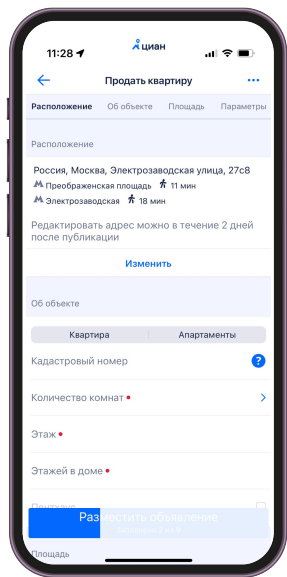
Текущая реализация: драйвер  
или якорь роста метрики?



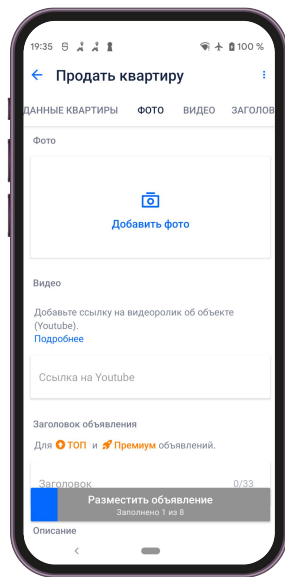
# Негативный User Experience



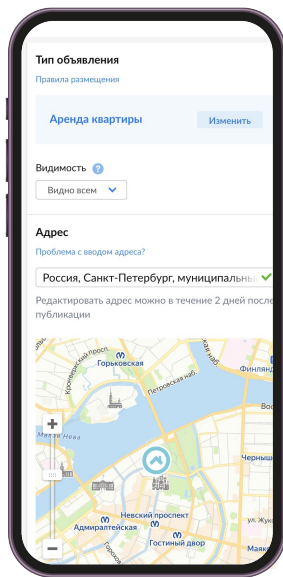
# Устаревший технологический стек



Objective-C



Java



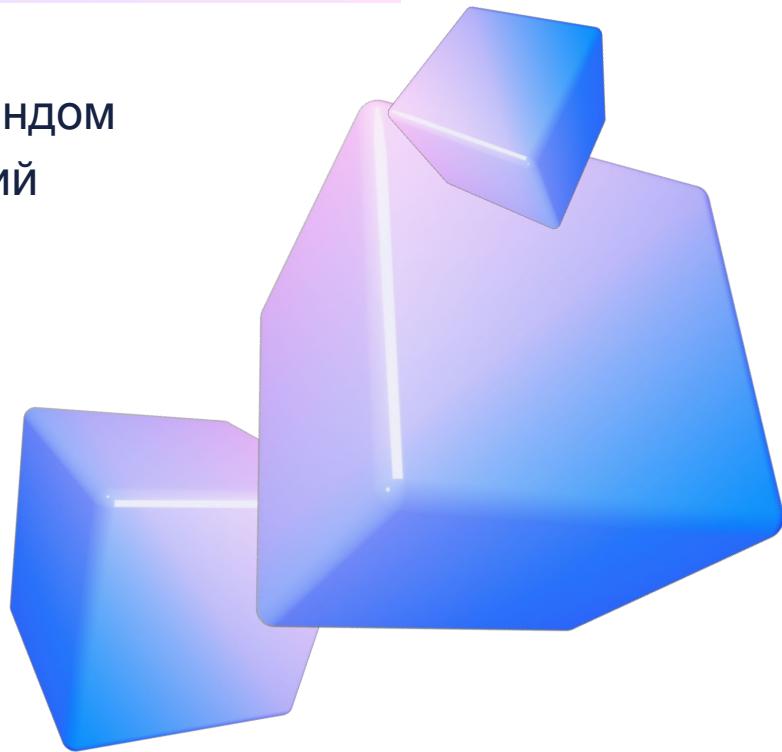
Angular 1.5

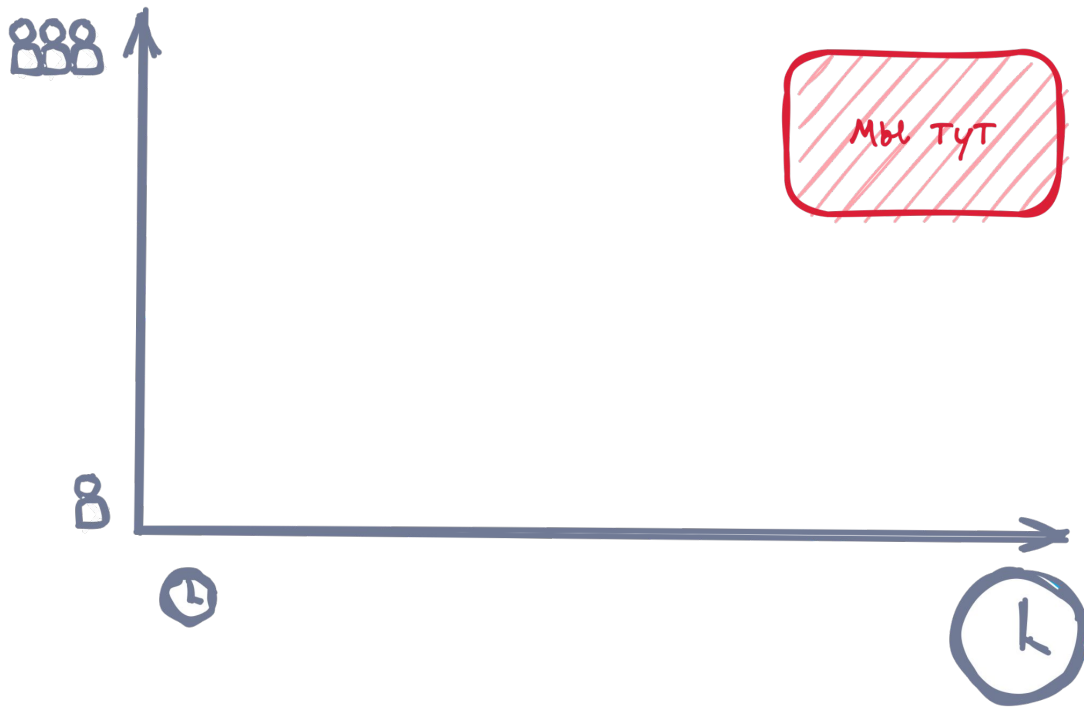
- Низкая стабильность продукта
- Высокая стоимость поддержки
- Негатив разработки
- Трудности с наймом

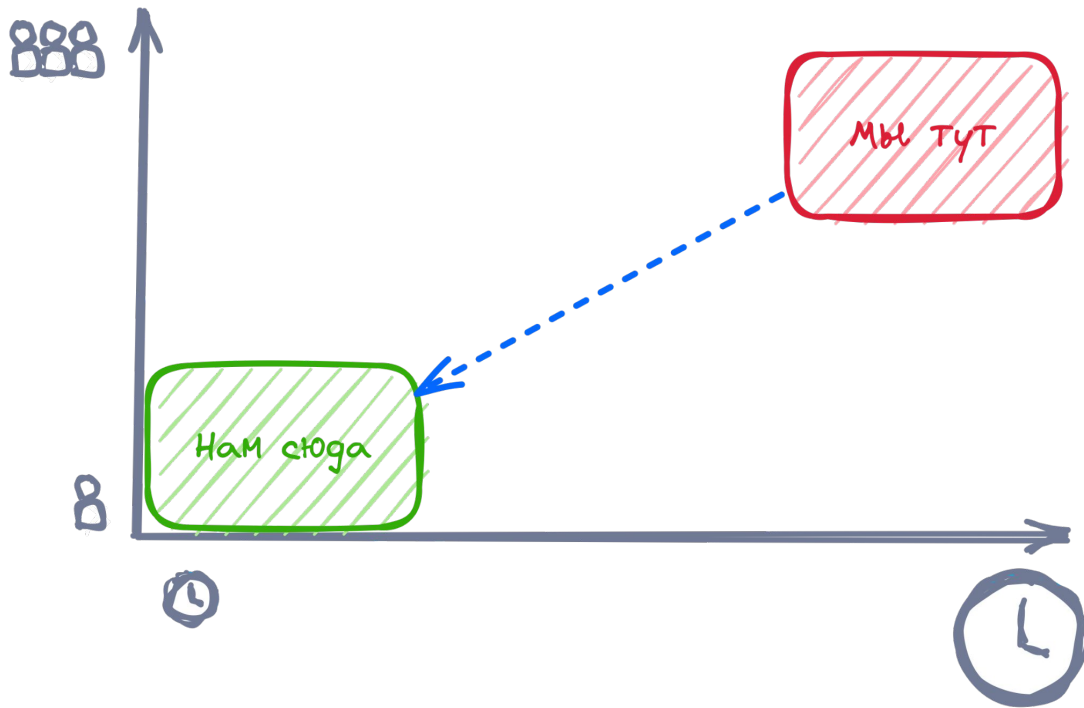
# Высокая стоимость поддержки

---

- Логика размазана между клиентом и бэкендом
- Необходимость всех стеков для изменений
- Волокита старых версий









Надо поработать!



# Что нам нужно?

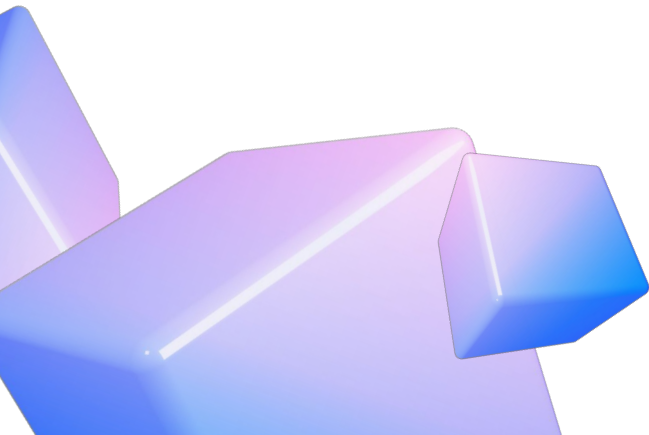
---

- современный платформенный стек
- предсказуемость оценок
- оптимальная стоимость изменений и поддержки
- высокий Bus Factor









Достигаются любой архитектурой

- консистентный UI
- синхронизация логики
- простота A/B экспериментов
- ускорение релизного цикла

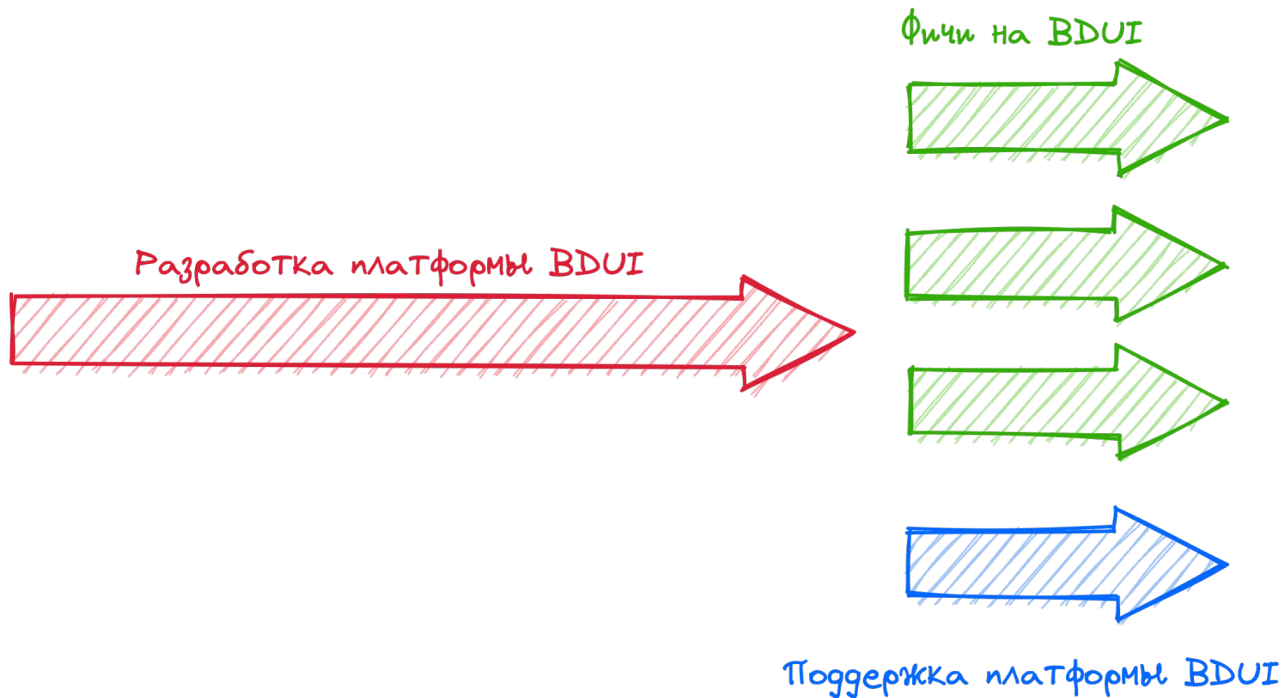
**Надо подумать!**



# Какие варианты?

Новая нативная реализация	WebView	BDUI
 Нативный UX	 Ненативный UX	 Нативный UX
 3 человека на фичу	2 человека на фичу	 1 человек на фичу
 TTM 2 недели	 TTM 1 день	 TTM 1 день

# BDUI дорог на старте



# С чего начали?

---

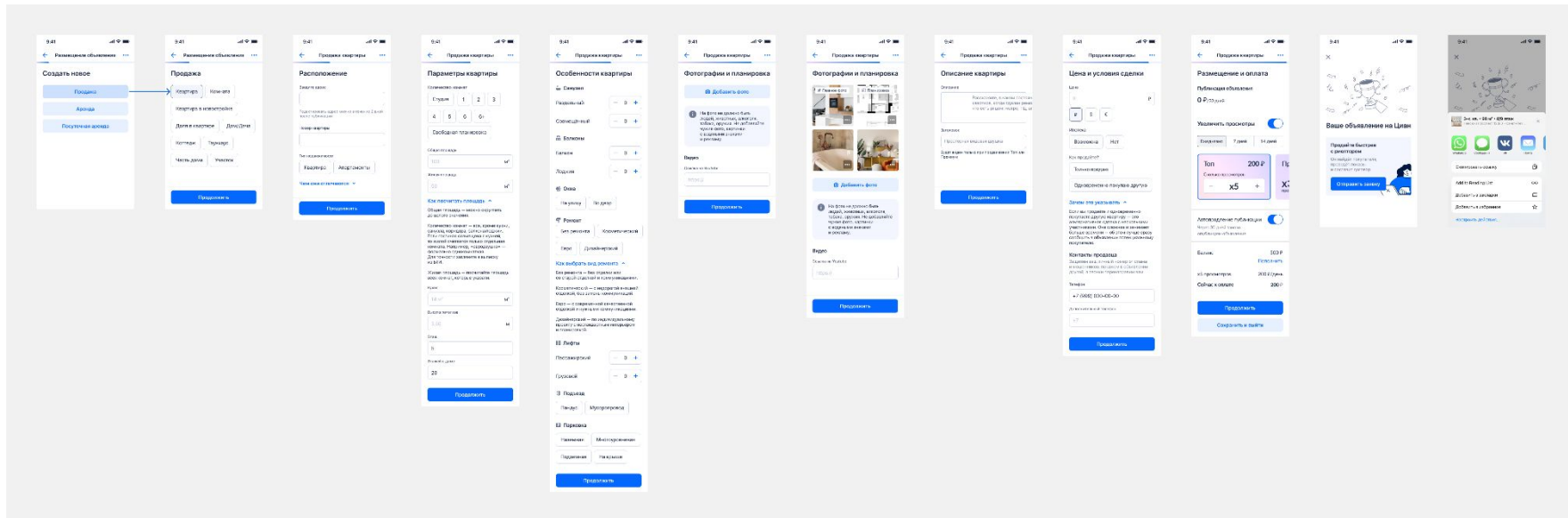
## Договорились с бизнесом:

- Дорогой старт
- Технические требования: универсальный, переиспользуемый для наших продуктовых вертикалей механизм
- Небольшая зернистость
- Использование дизайн-системы

Реализация универсальная в рамках компании, заточена под нас

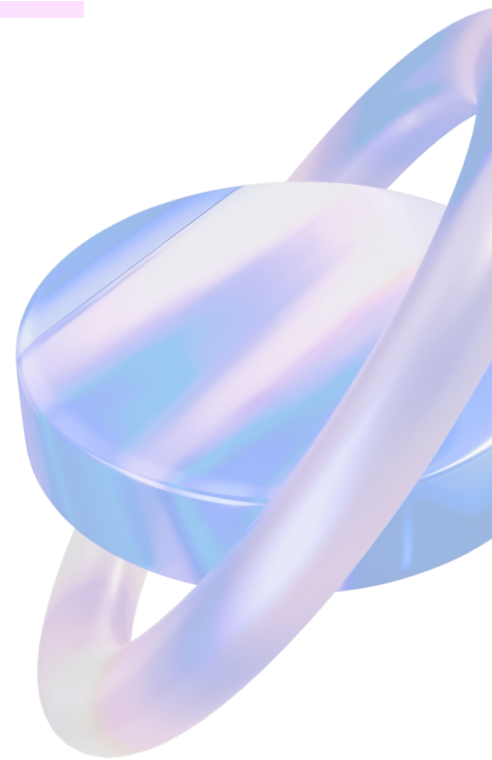
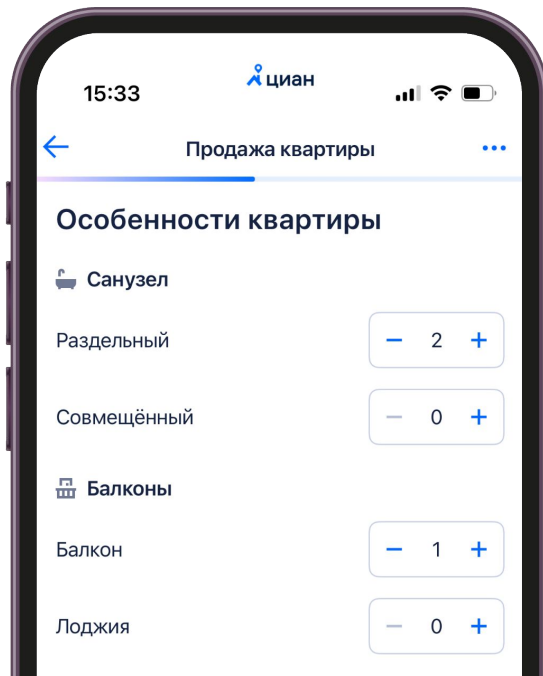
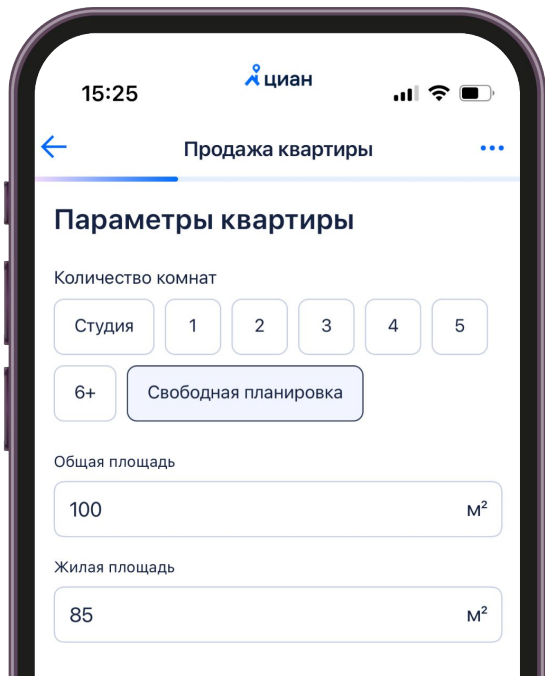


Хотите посмотреть?



# Новая форма

- Много инпутов
- Есть взаимосвязанные элементы





# Новая форма

## Есть форматирование

Бонус посреднику

10 555 555 ₺

₺ \$ € %

Телефон

+7 993 883-83-99

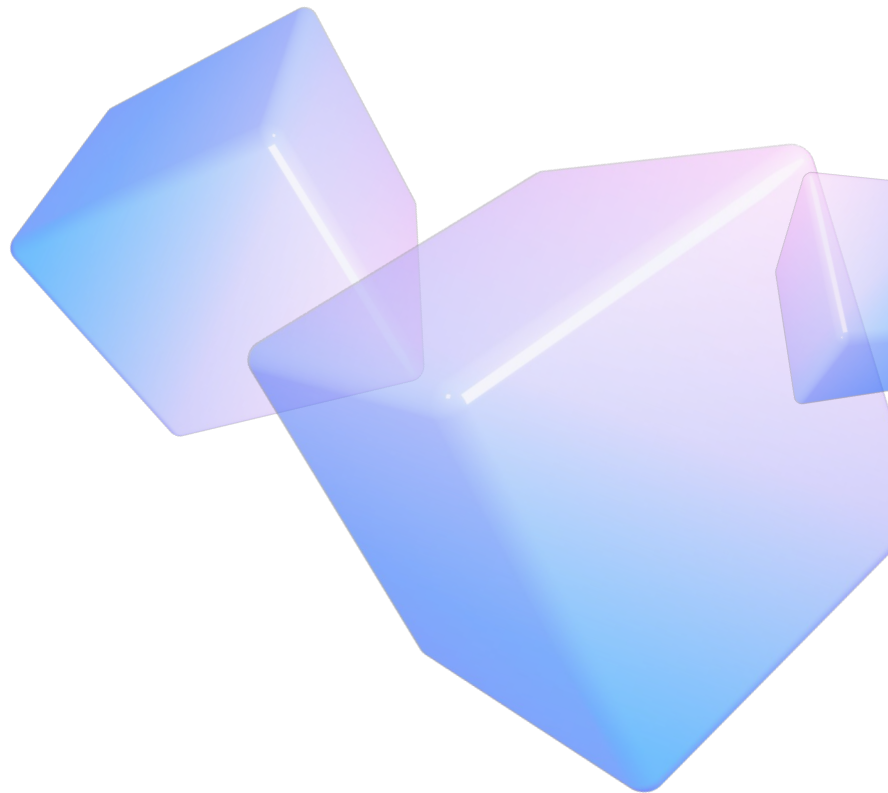
# Новая форма

Требуется валидация

Общая площадь

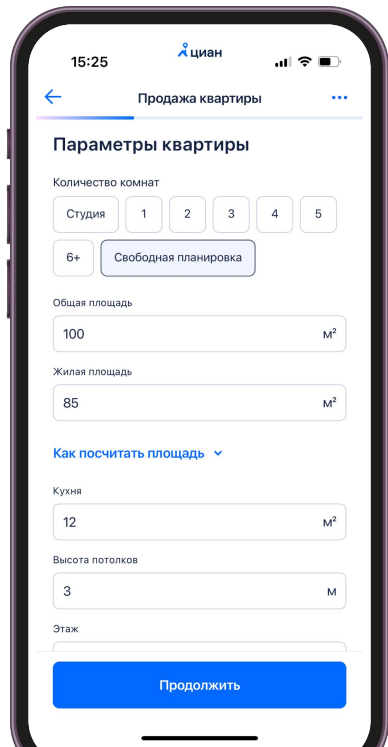
М<sup>2</sup>

Заполните площадь



# Новая форма

## Пошаговая, присутствуют переходы



15:25 | циан

← Продажа квартиры

### Параметры квартиры

Количество комнат

Студия 1 2 3 4 5

6+ Свободная планировка

Общая площадь

100 м<sup>2</sup>

Жилая площадь

85 м<sup>2</sup>

Как посчитать площадь ▾

Кухня

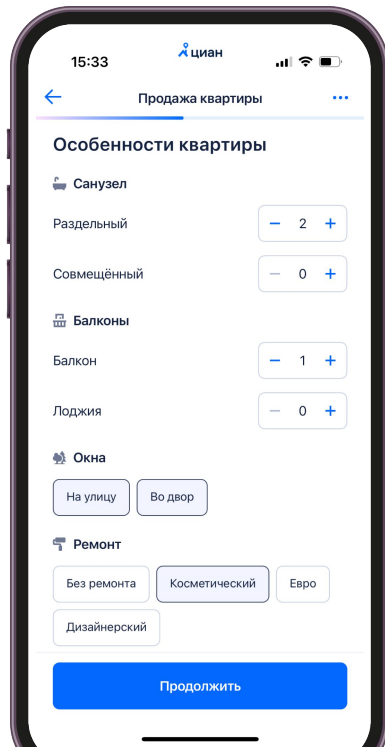
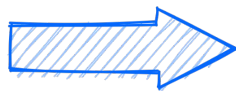
12 м<sup>2</sup>

Высота потолков

3 м

Этаж

Продолжить



15:33 | циан

← Продажа квартиры

### Особенности квартиры

Санузел

Раздельный - 2 +

Совмещённый - 0 +

Балконы

Балкон - 1 +

Лоджия - 0 +

Окна

На улицу Во двор

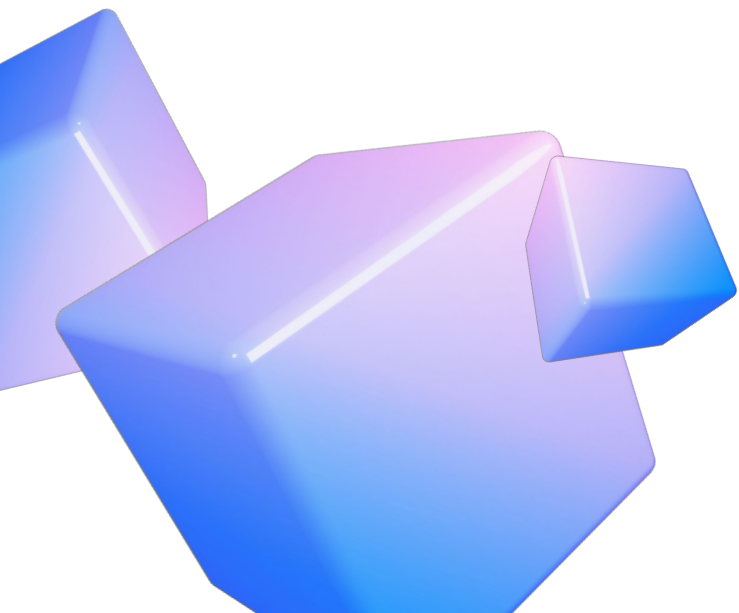
Ремонт

Без ремонта Косметический Евро

Дизайнерский

Продолжить

- Инпуты
- Взаимосвязанные элементы
- Форматирование
- Валидация
- Переходы



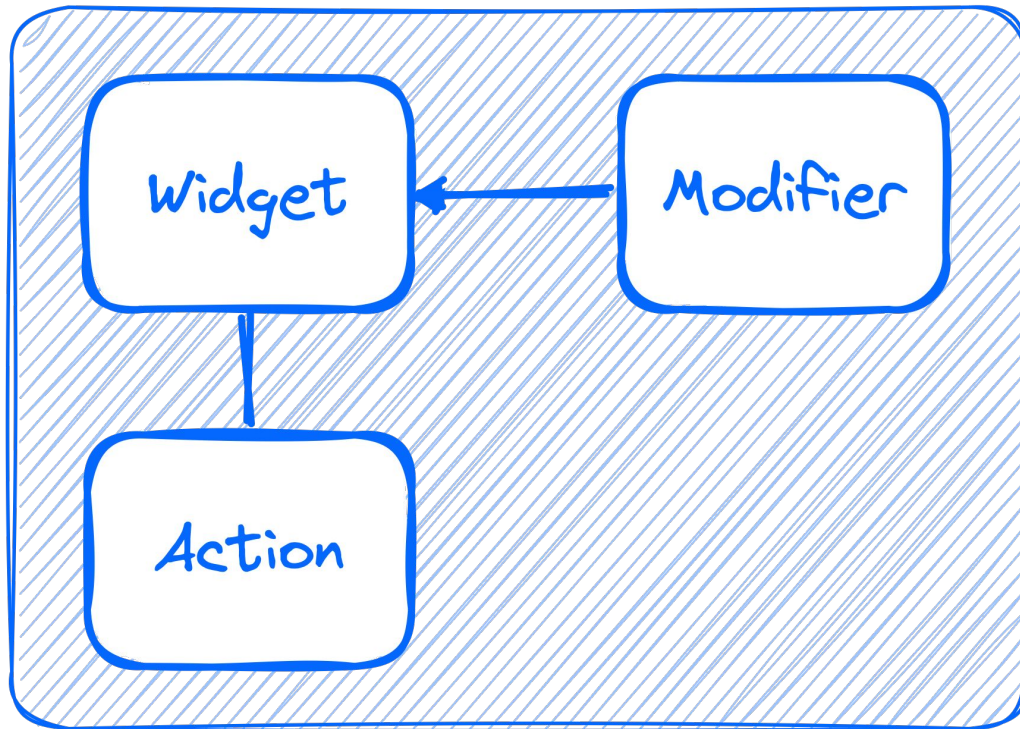
И что нам со всем этим делать?

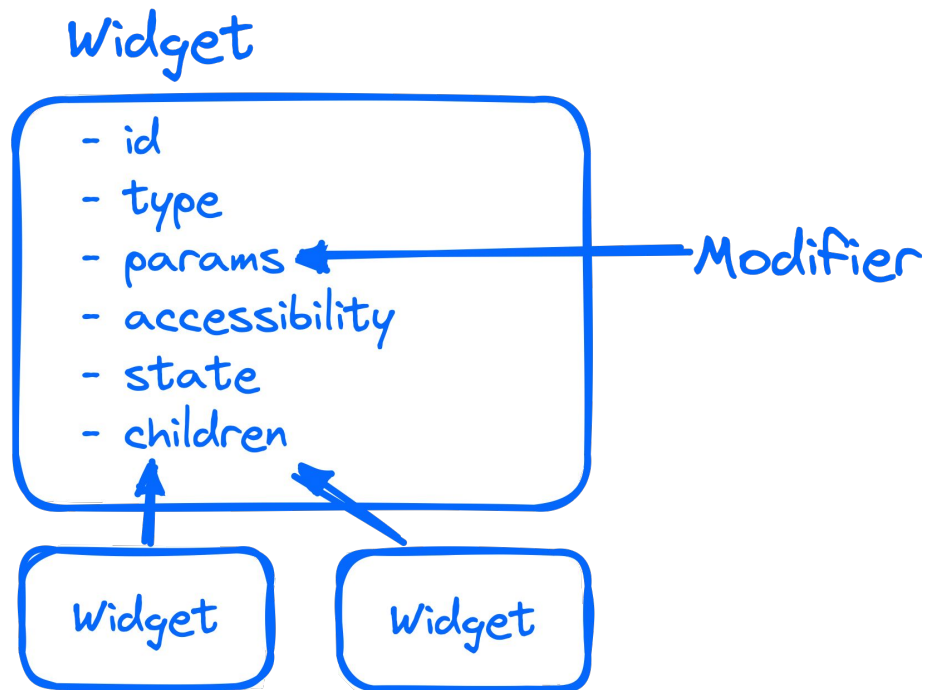


Давайте посмотрим,  
как мы выкрутились!



# JSON



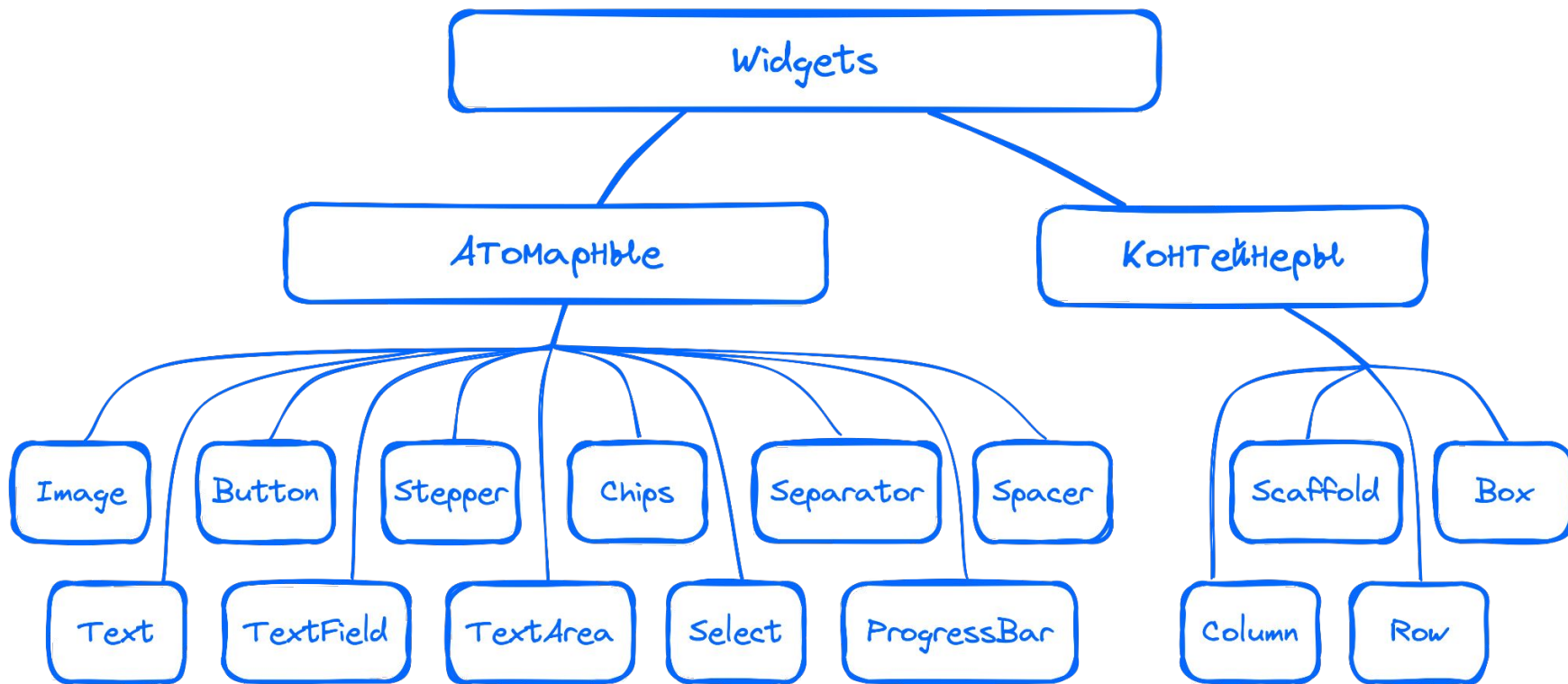




# Виджеты

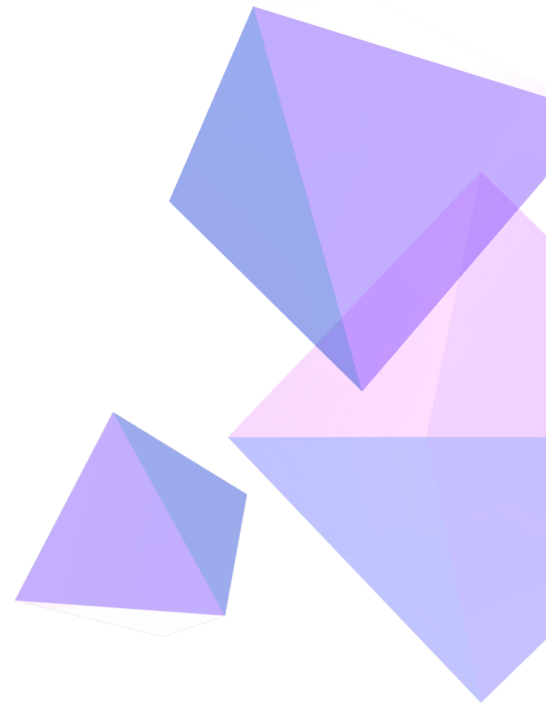
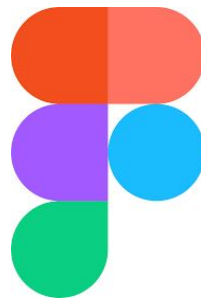
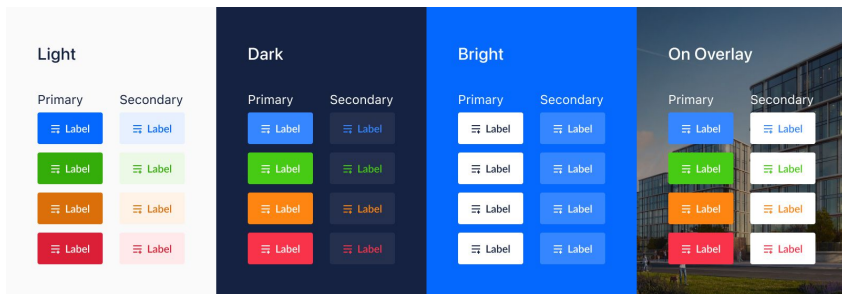
```
{  
  "id": "input1",  
  "type": "TextField",  
  "params": {  
    "configuration": {  
      "title": { "value": "Общая площадь" },  
      "inputSuffix": { "value": "м²" }  
    }  
  }  
}
```

Общая площадь



# Дизайн-система

- Готовая спецификация определила контракт
- Готовая реализация
- Отлаженные edge-кейсы



# Модификаторы

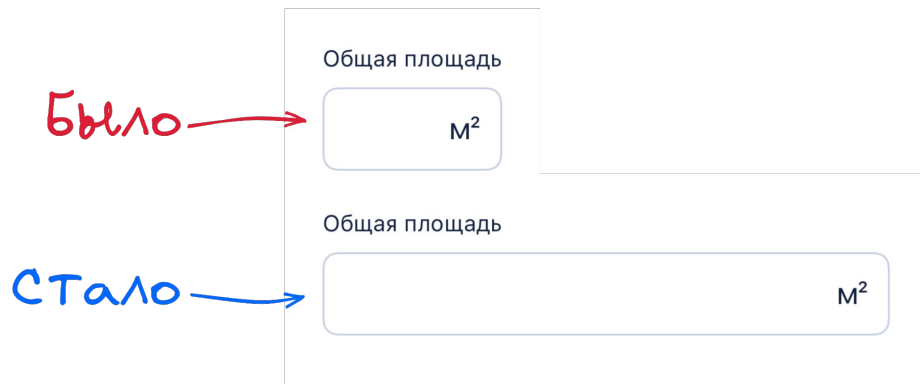
---

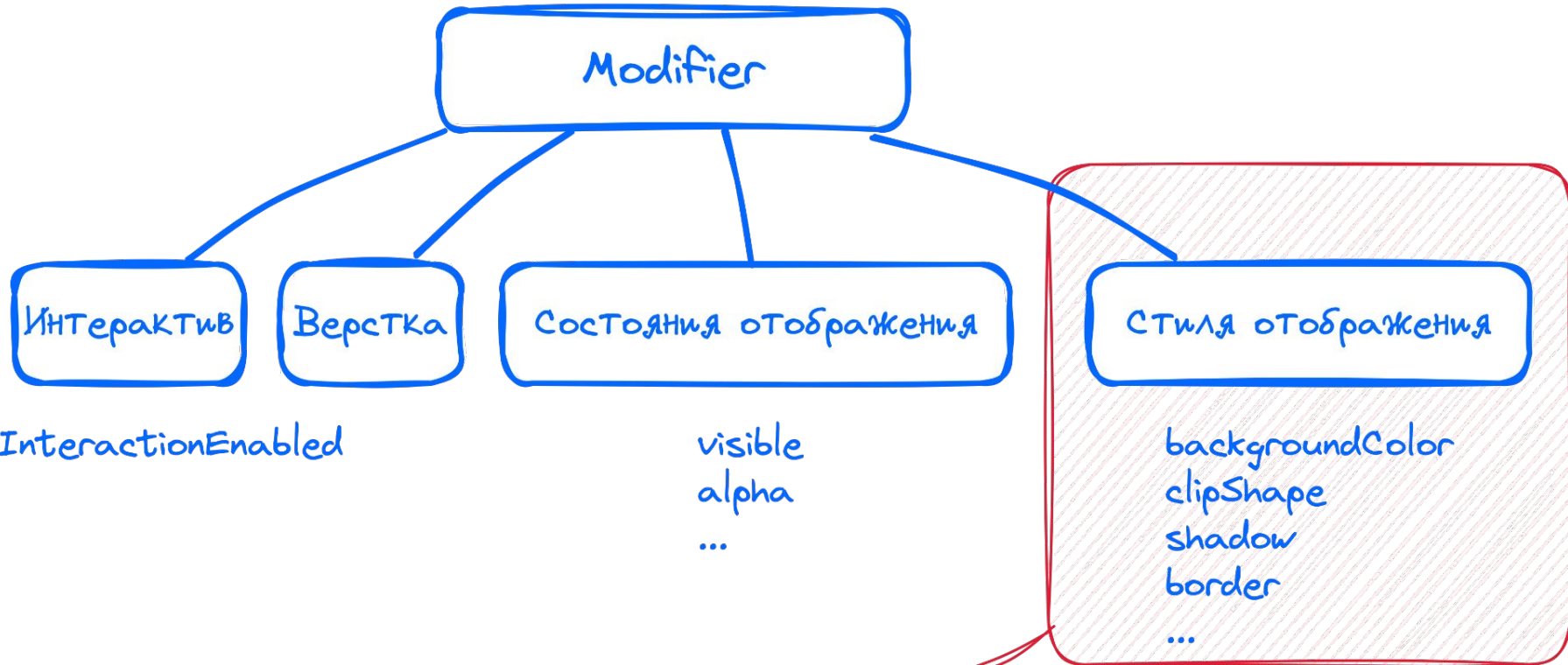
## Modifier

- type
- name
- value
- params

# Модификаторы

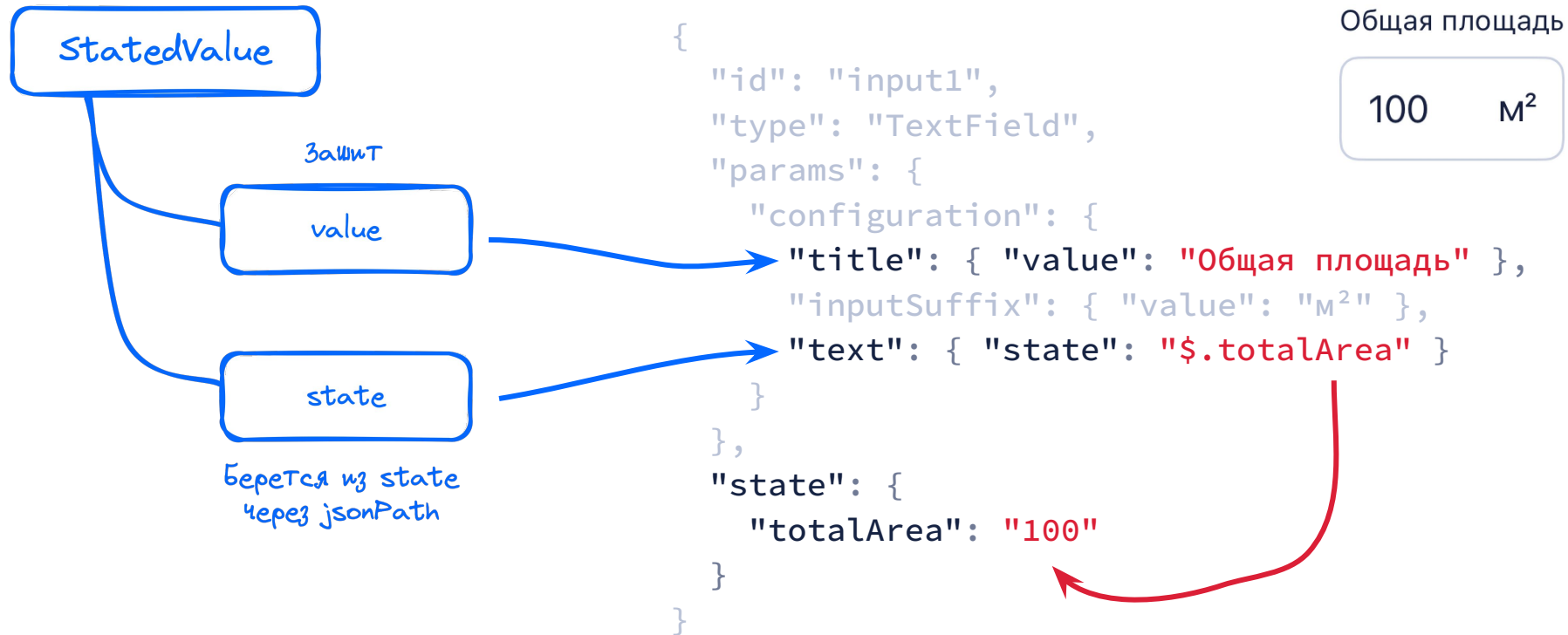
```
{
  "id": "input1",
  "type": "TextField",
  "params": {
    "configuration": {
      "title": { "value": "Общая площадь" },
      "inputSuffix": { "value": "м²" }
    }
  }
  "modifier": [{
    "name": "width",
    "type": "FloatLayout",
    "value": { "value": 300 }
  }]
}
```



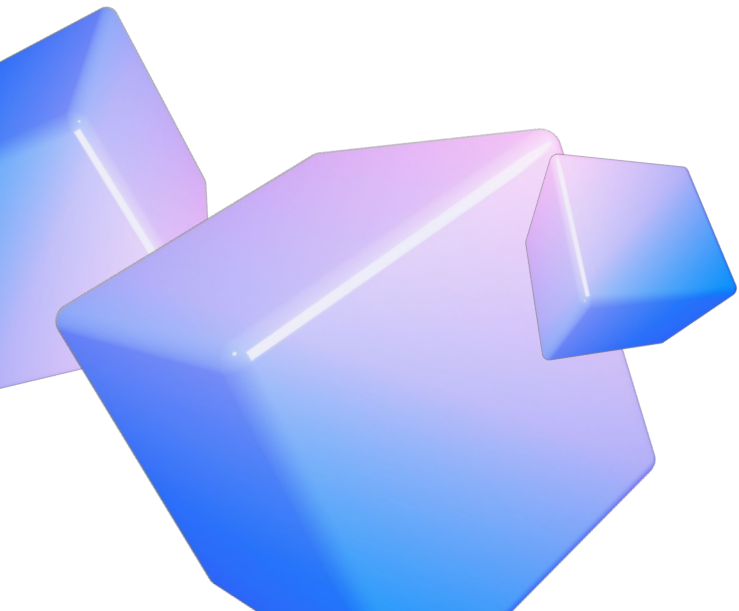


Поддерживают не все виджеты

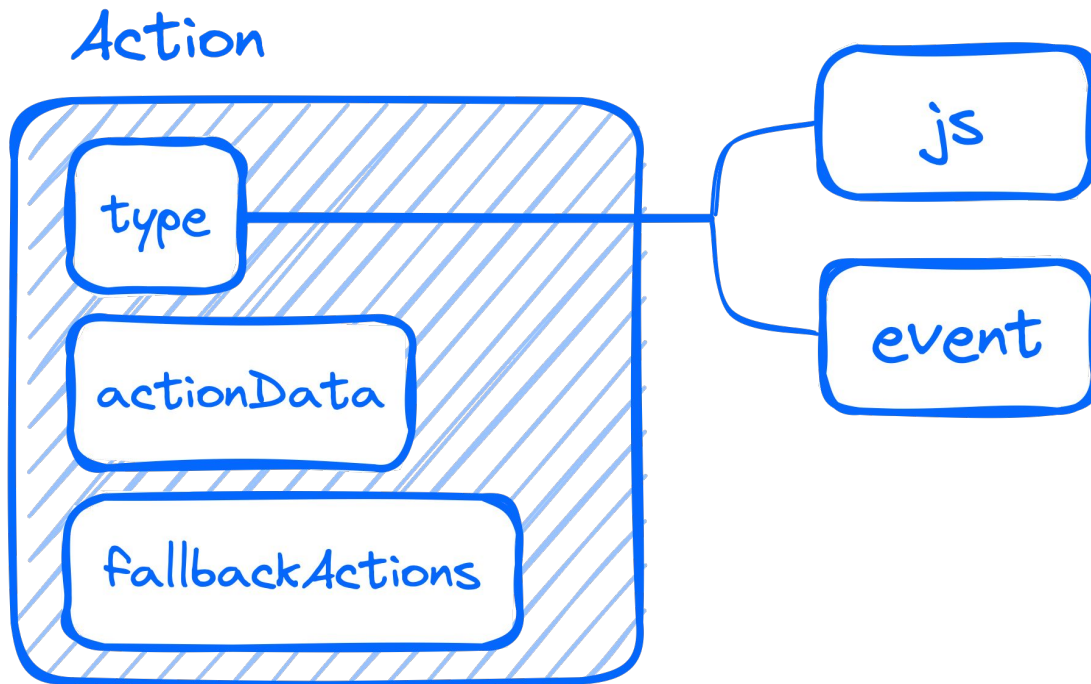
# Состояния



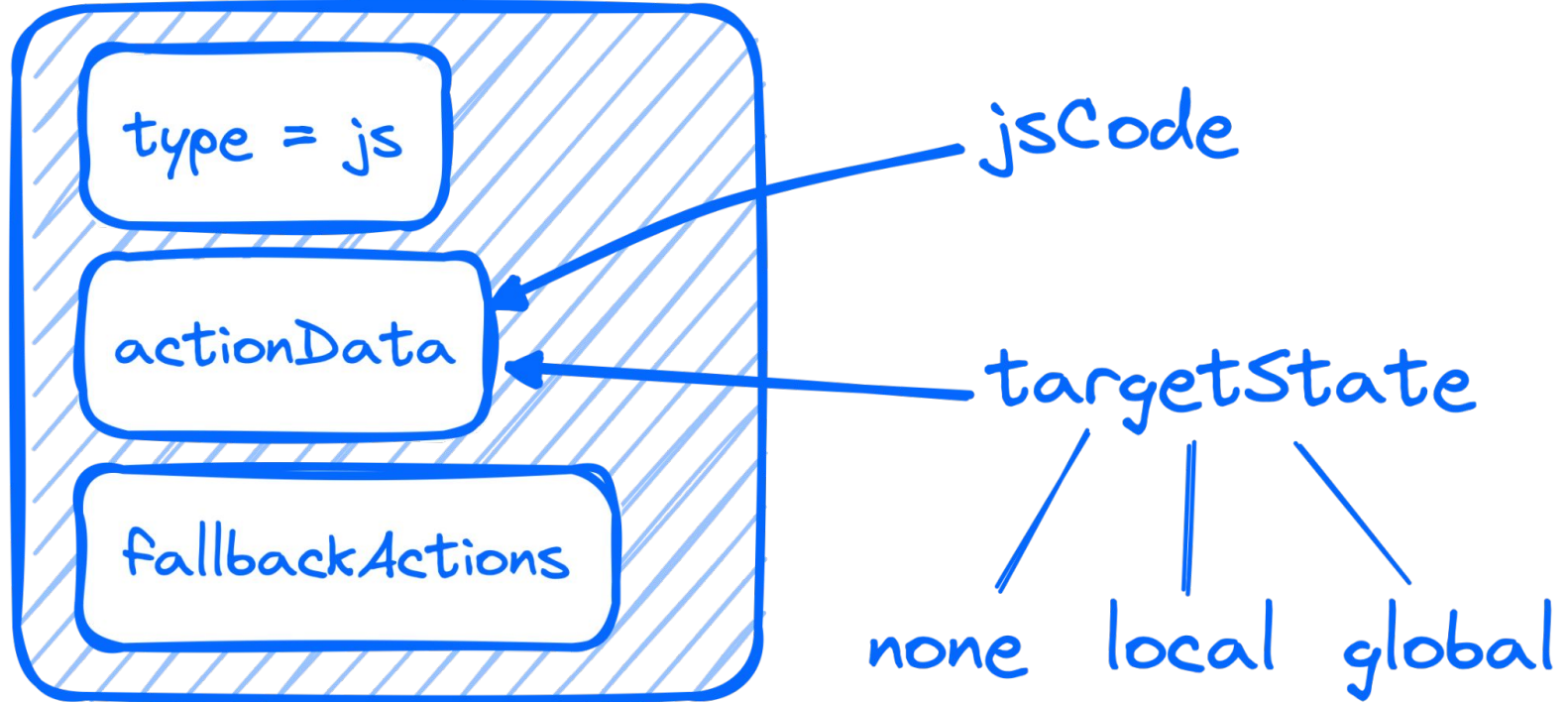
- **Инпуты**
- Взаимосвязанные элементы
- Форматирование
- Валидация
- Переходы







# JS Action



# JSAction

```
{
  ...
  "params": {
    "configuration": {
      ...
      "text": { "state": "$totalArea" }
      "onTextEdit": [{
        "type": "js",
        "actionData": {
          "targetState": "local"
          "jsCode":
        }
      }]
    }
  }
  "state": {
    "totalArea": "100"
  }
}
```

Общая площадь

 M<sup>2</sup>

```
function main(globalState, localState, args)
{
  let newState = localState;
  newState['totalArea'] = args['inputText'];
  return newState;
}
```

# Взаимосвязанные элементы

- Состояние элемента зависит от действий
- От состояния одного элемента зависит состояние другого

15:13 Циан

← Продажа квартиры

6+ Свободная планировка

Общая площадь

100 м²

Жилая площадь

м²

Как посчитать площадь ▾

Кухня

м²

Высота потолков

м

Этаж

2

Этажей в доме

5

Продолжить

## Как посчитать площадь ▾



```
function main(globalState, localState, args) {  
  let newState = globalState;  
  newState['infoExpanded'] = !newState['infoExpanded'];  
  return newState;  
}
```

```
"state": {  
  "infoExpanded": false  
}
```

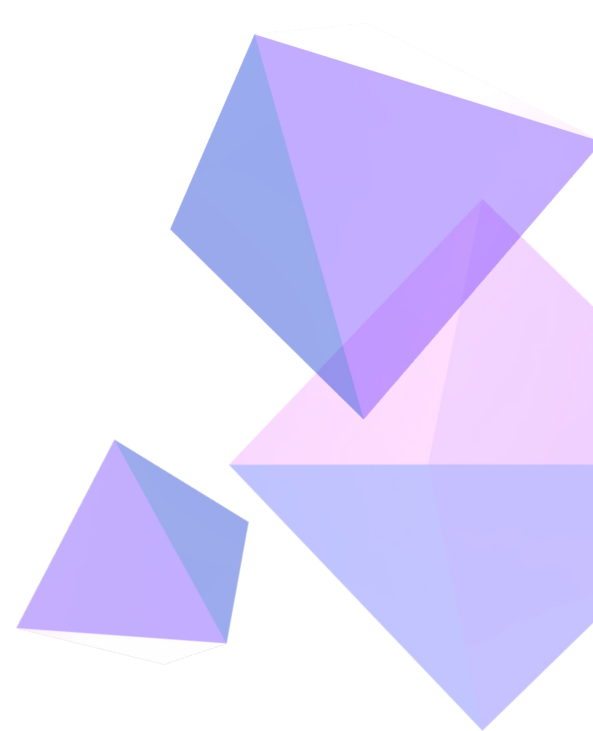


Общая площадь — можно округлить до целого значения.

↓  
*json*

```
{
  "id": "info",
  "type": "Text",
  "params": {
    "text": { "value": "Общая площадь - ..." },
    "modifier": [{
      "type": "Bool",
      "name": "visible",
      "value": { "state": "$.infoExpended" }
    }]
  }
}

"state": {
  "infoExpended": true
}
```



15:13 Циан

← Продажа квартиры ...

6+ Свободная планировка

Общая площадь

100 м<sup>2</sup>

Жилая площадь

м<sup>2</sup>

Как посчитать площадь ▾

Кухня

м<sup>2</sup>

Высота потолков

М

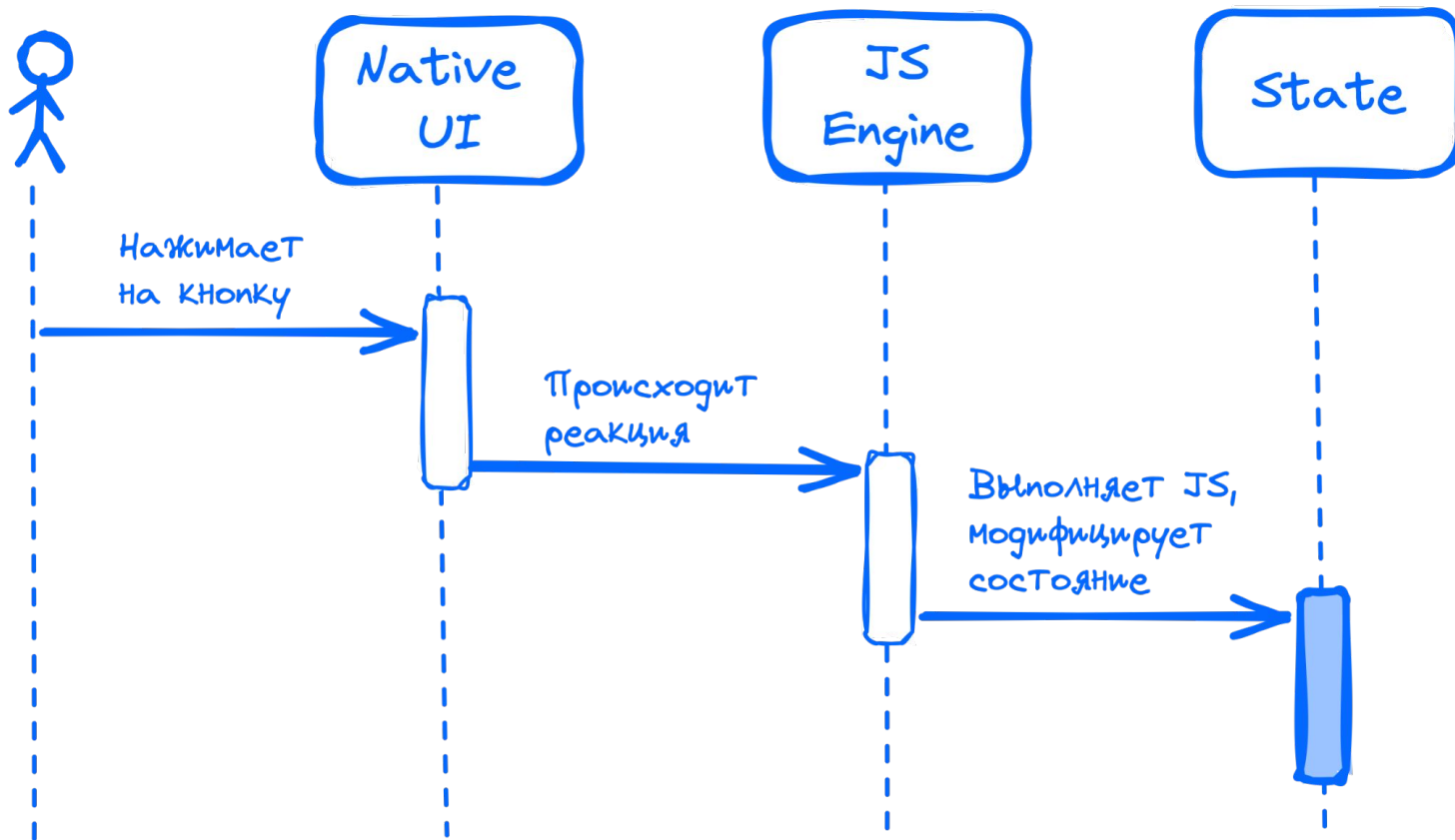
Этаж

2

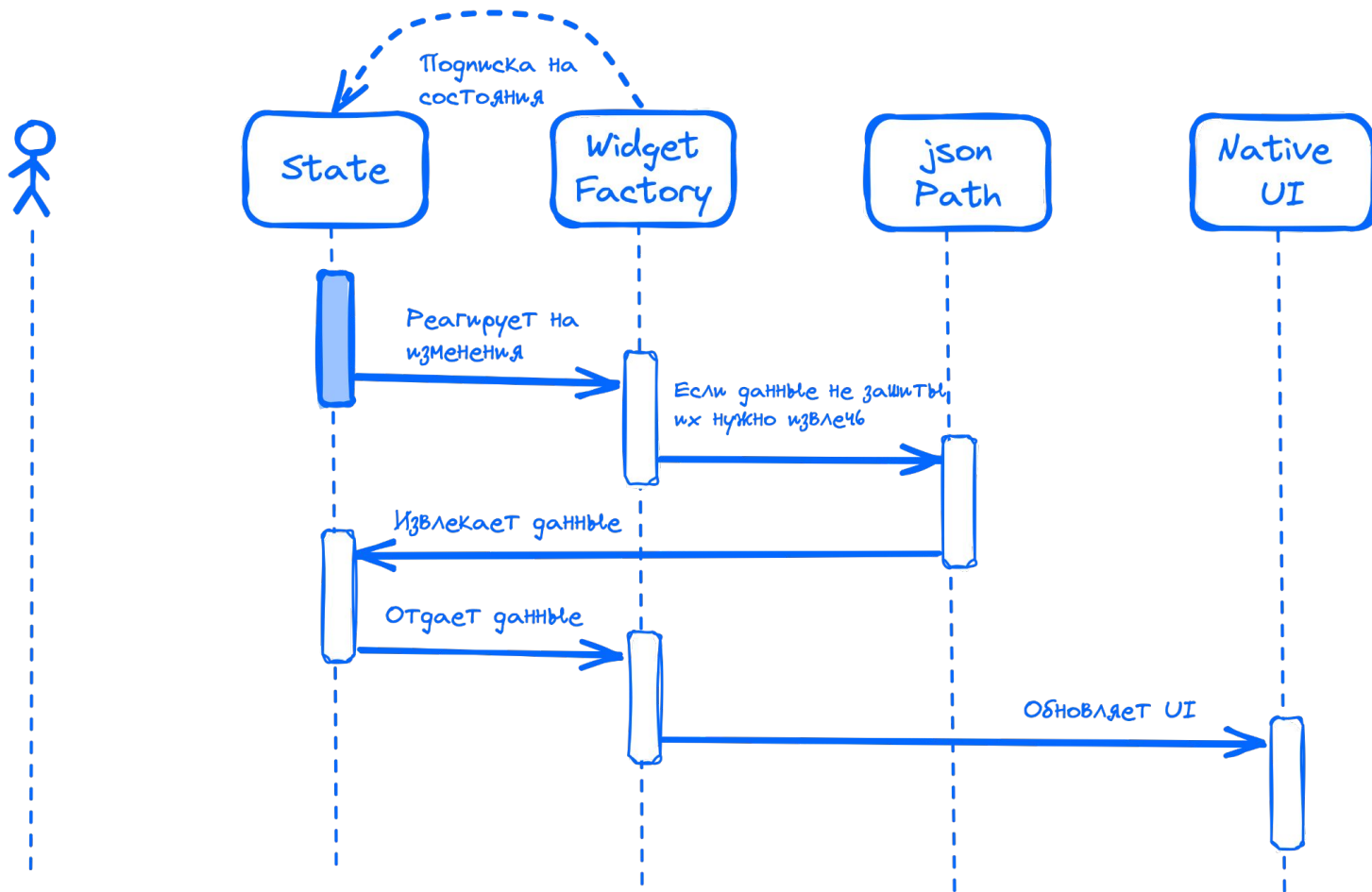
Этажей в доме

5

Продолжить







# Почему JS?

---

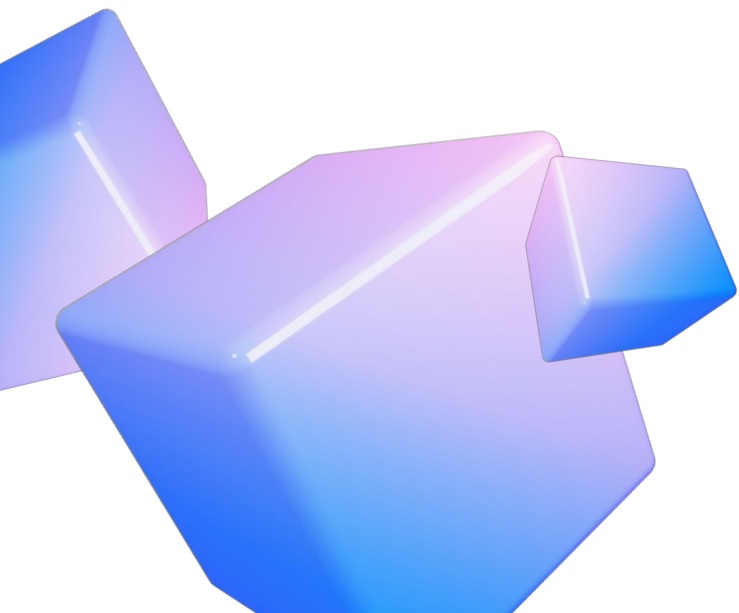
## JS:

- + В связке со state образует универсальный механизм
- + Минимум логики на клиентах
- + Изменения без правок на клиентах

## Conditions/Регулярки:

- Новое условие — новый тип Conditions
- Логика на клиентах
- Изменения с правками на клиентах

- Инпуты
- Взаимосвязанные элементы
- Форматирование
- Валидация
- Переходы

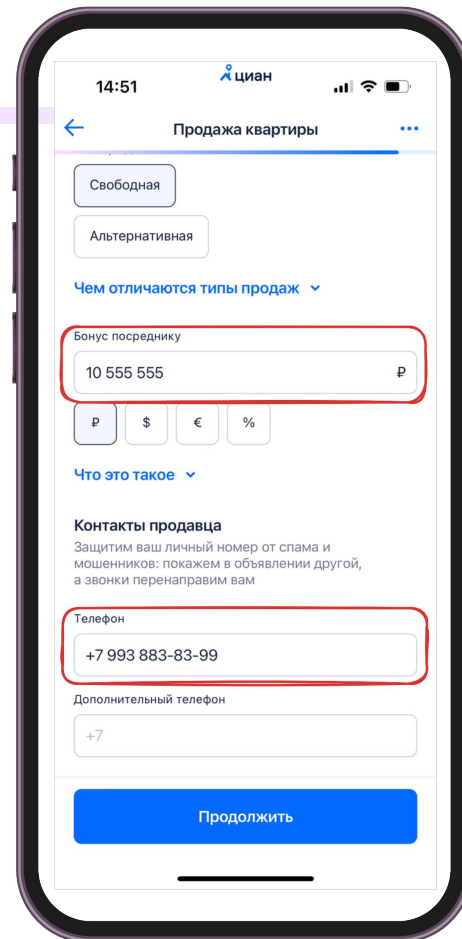


# Форматирование

## Требуется инпутам

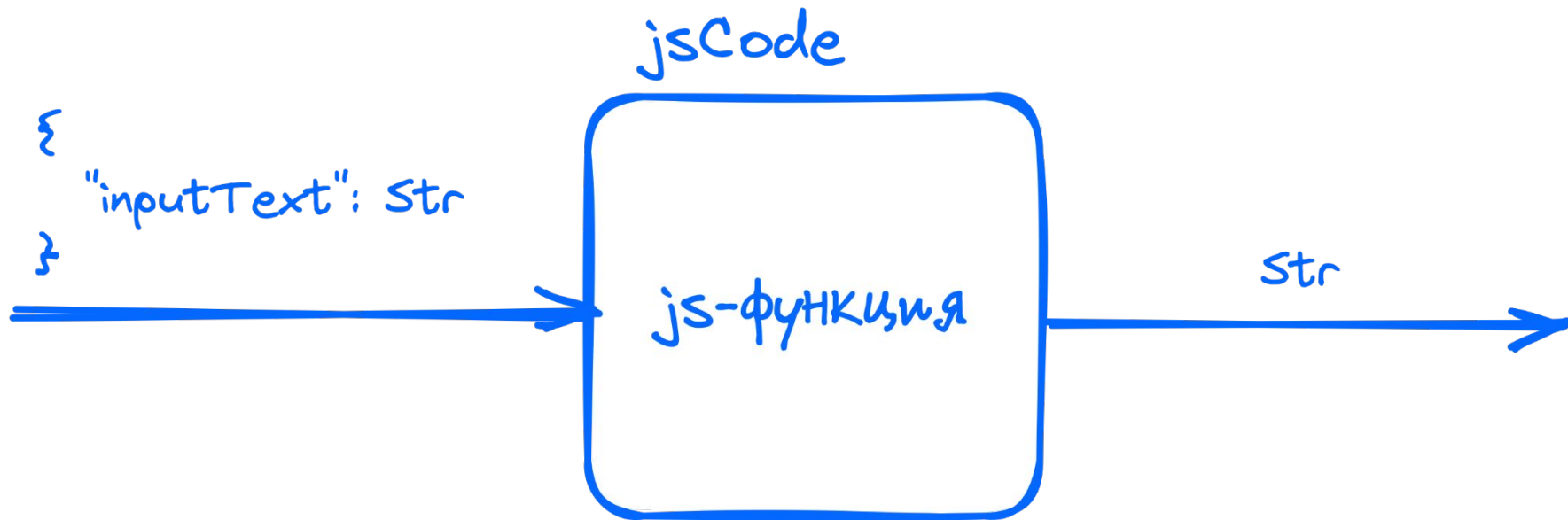
Поле `inputFormatter` — для стандартных:

- price
- priceFractional
- phone



# Форматирование

Поле **onTextChanged** — для кастомных



# JSAction

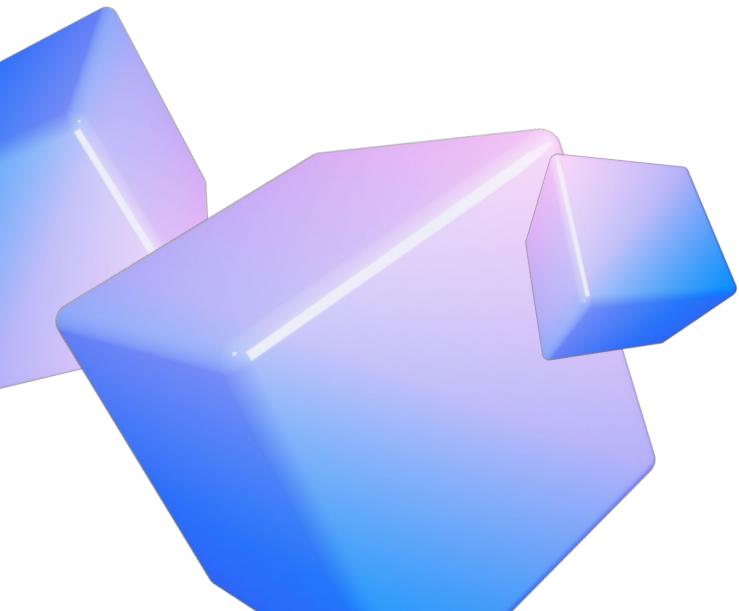
```
{
  "id": "cadastralNumberField",
  "type": "TextField",
  "params": {
    "configuration": {
      "title": { "value": "Кадастровый номер" },
      "onTextChanged": {
        "jsCode":
      }
    }
  }
}
```

Кадастровый номер

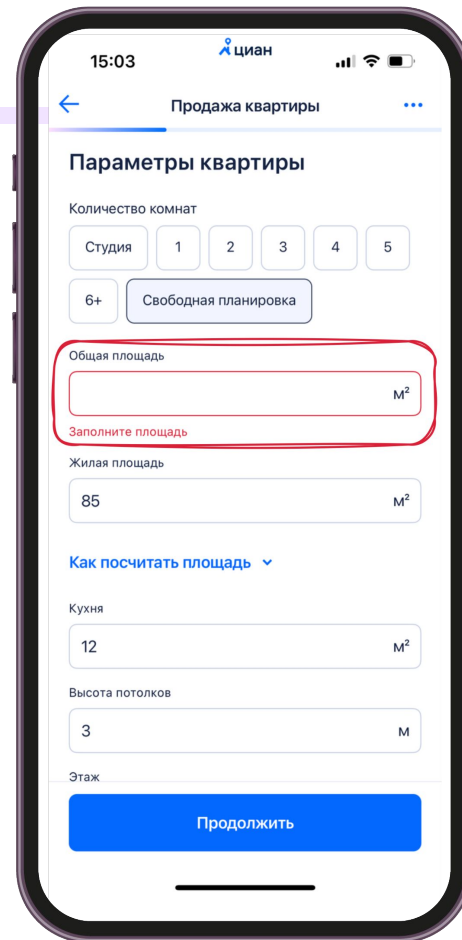
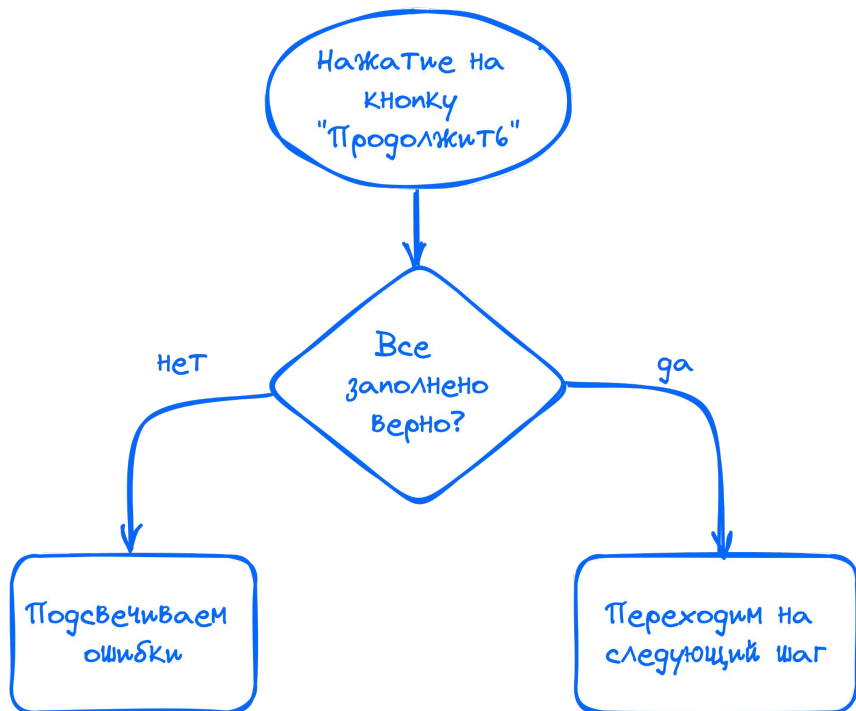
77:03:0003012:1067

```
function main(args) {
  const text = args['inputText'];
  const regex = /[0-9:]/g;
  const formattedText = text.match(regex);
  if (formattedText) {
    return formattedText.join('');
  } else {
    return '';
  }
}
```

- Инпуты
- Взаимосвязанные элементы
- Форматирование
- Валидация
- Переходы

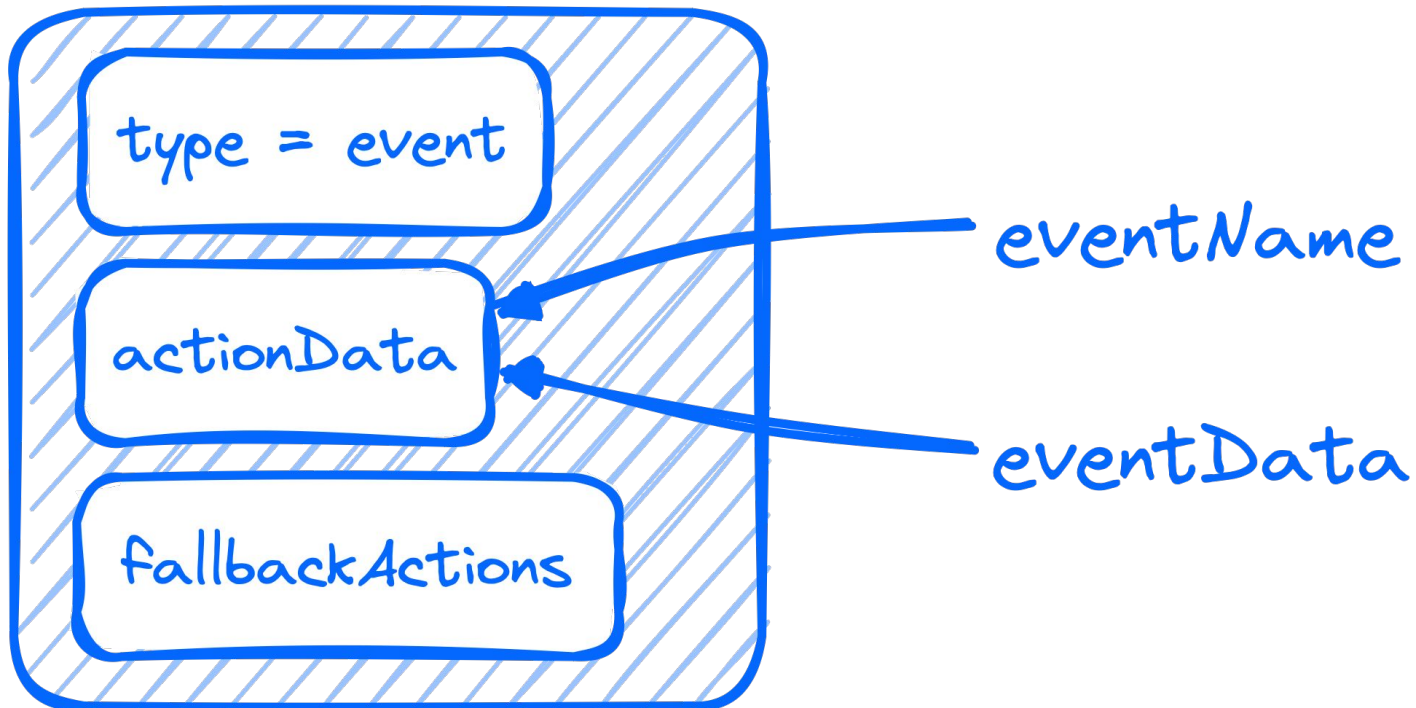


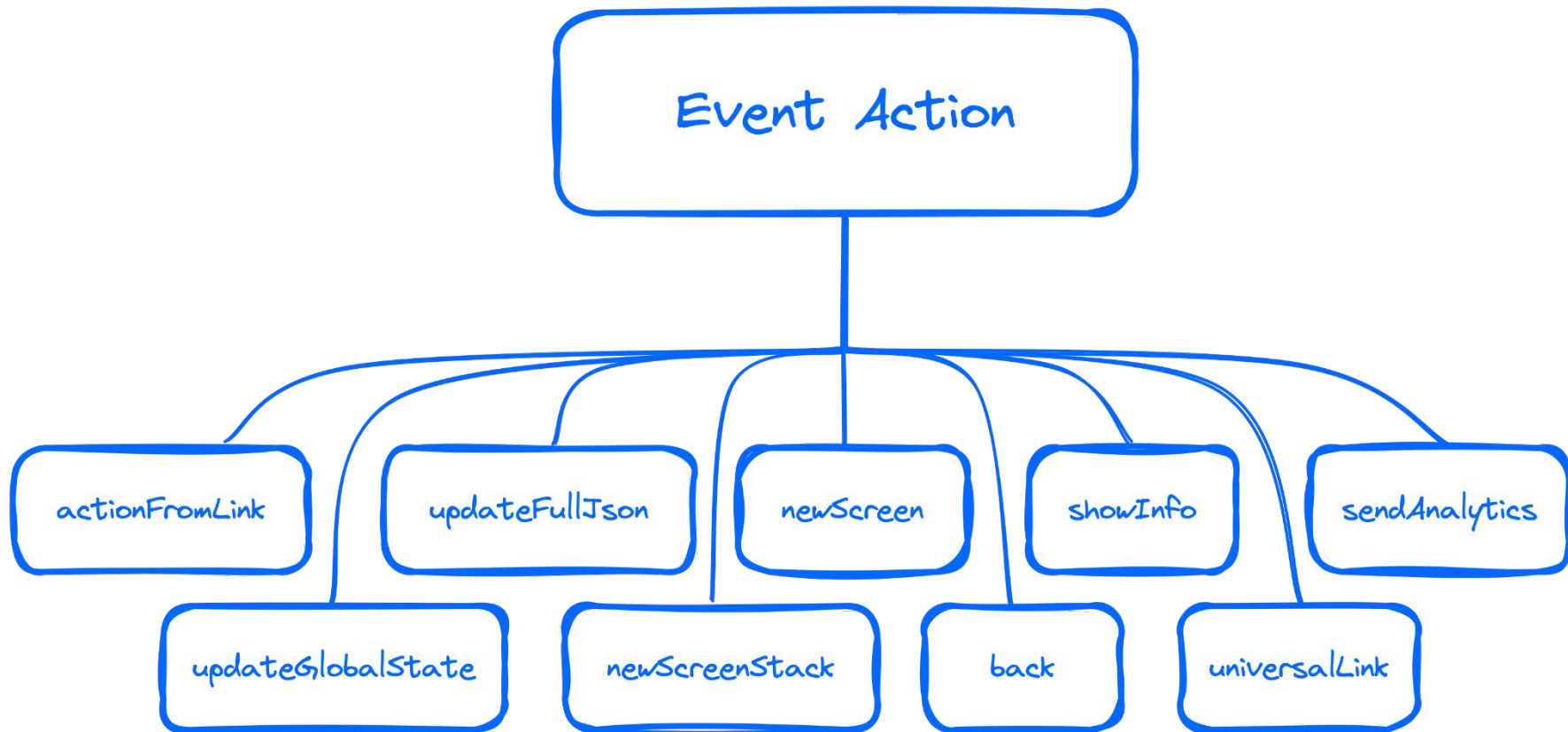
# Валидация





# Event Action





# Валидация

## Сделать запрос:

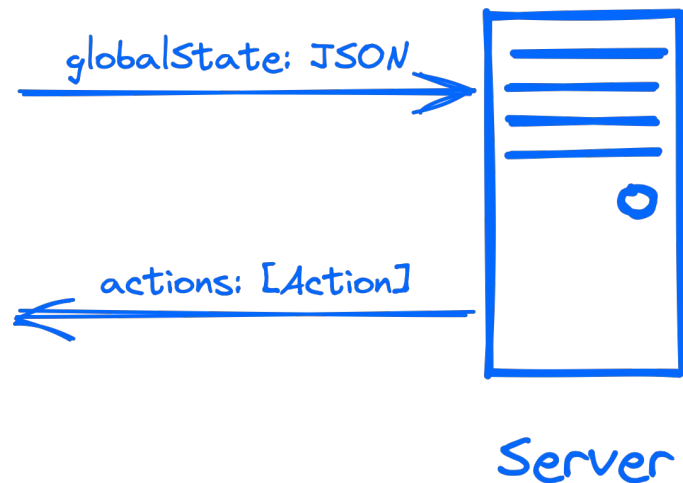
```
{
  "type": "event",
  "actionData": {
    "eventName": "actionFromLink",
    "eventData": {
      "link": "https://api.cian.ru/.../next-step"
    }
  }
}
```

## Показать состояние загрузки:

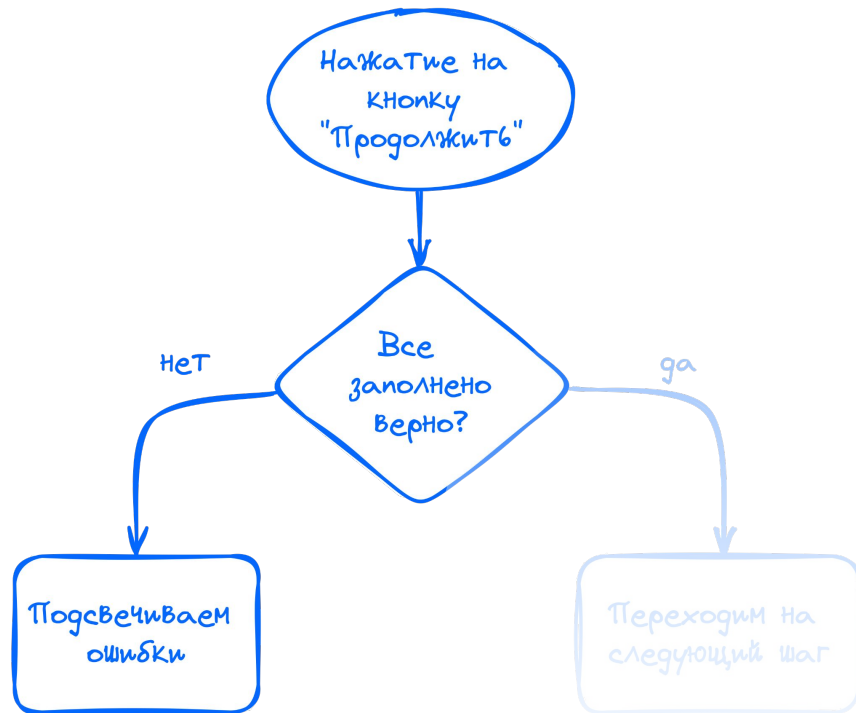
```
{
  "type": "event",
  "actionData": {
    "eventName": "updateGlobalState",
    "eventData": {
      "unionStrategy": "merge",
      "newState": {
        "enabled": false,
        "loading": true
      }
    }
  }
}
```

# Клиент-серверное взаимодействие

- Есть встроенный механизм клиент-серверного взаимодействия
- Рекурсивный полиморфный контракт (виджеты выстраиваются в иерархию)
- Единый формат



# Ошибка валидации



# Ошибка валидации

```
{
  "actions": [{
    "type": "event",
    "actionData": {
      "eventName": "updateGlobalState",
      "eventData": {
        "unionStrategy": "merge",
        "newState": {
          "enabled": true,
          "loading": false,
          "totalAreaError": "Заполните площадь"
        }
      }
    }
  ]
}
```

*Server* →

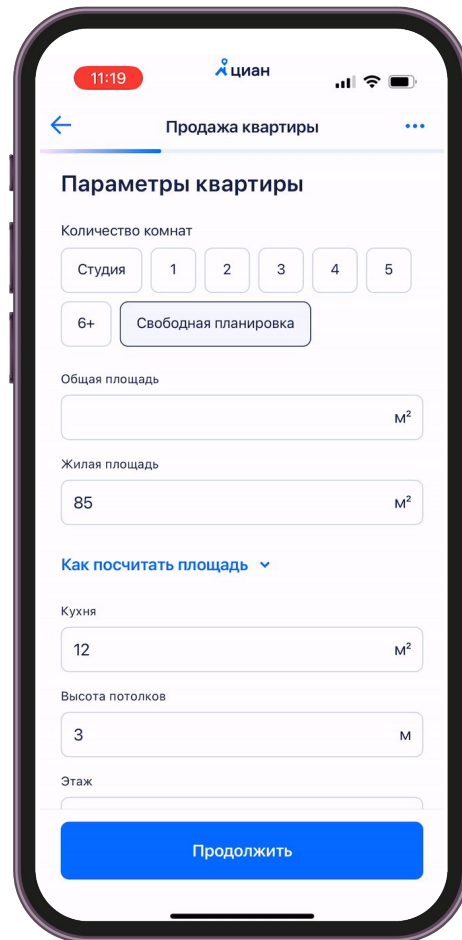
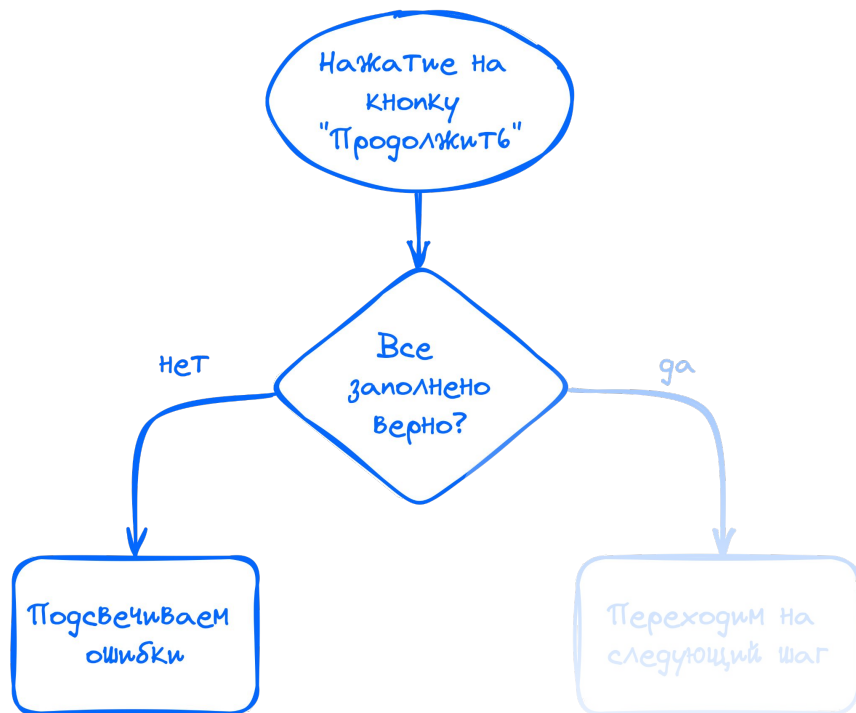
# Ошибка валидации

```
{
  "id": "input1",
  "type": "TextField",
  "params": {
    "configuration": {
      "title": { "value": "Общая площадь" },
      "invalidTextState": { "state": "$.totalAreaError" }
    }
  }
}
```

Общая площадь

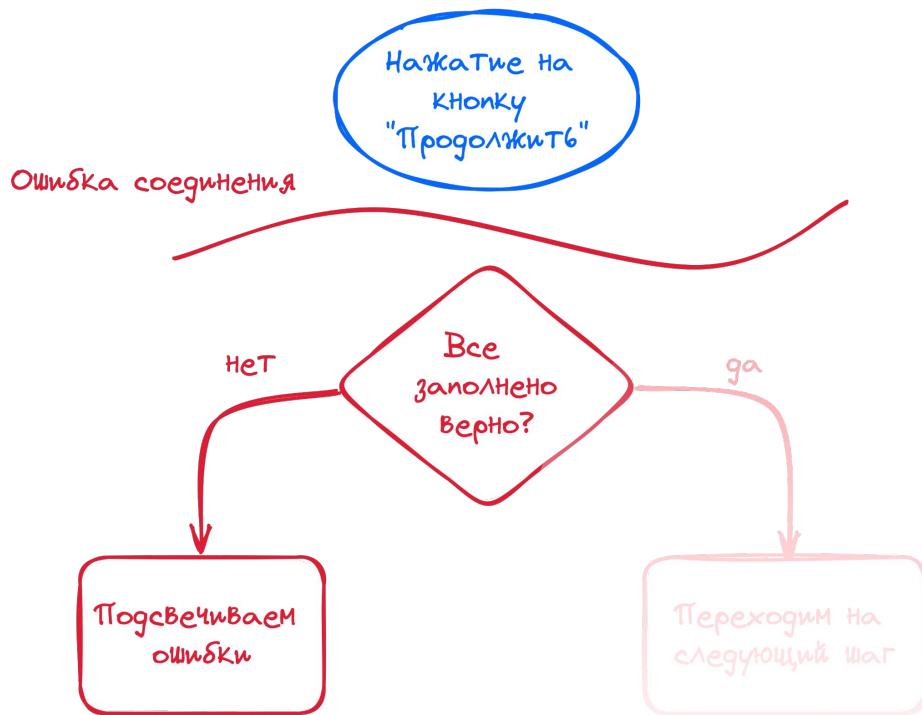
М<sup>2</sup>

Заполните площадь





# А что, если ошибка соединения?

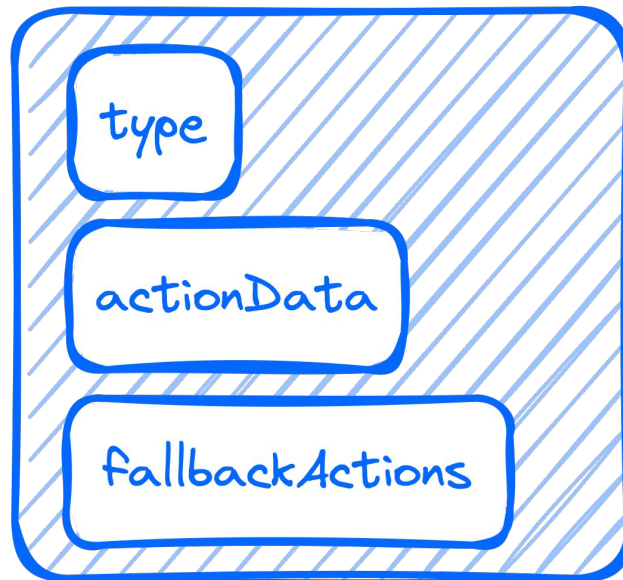


## Требуется:

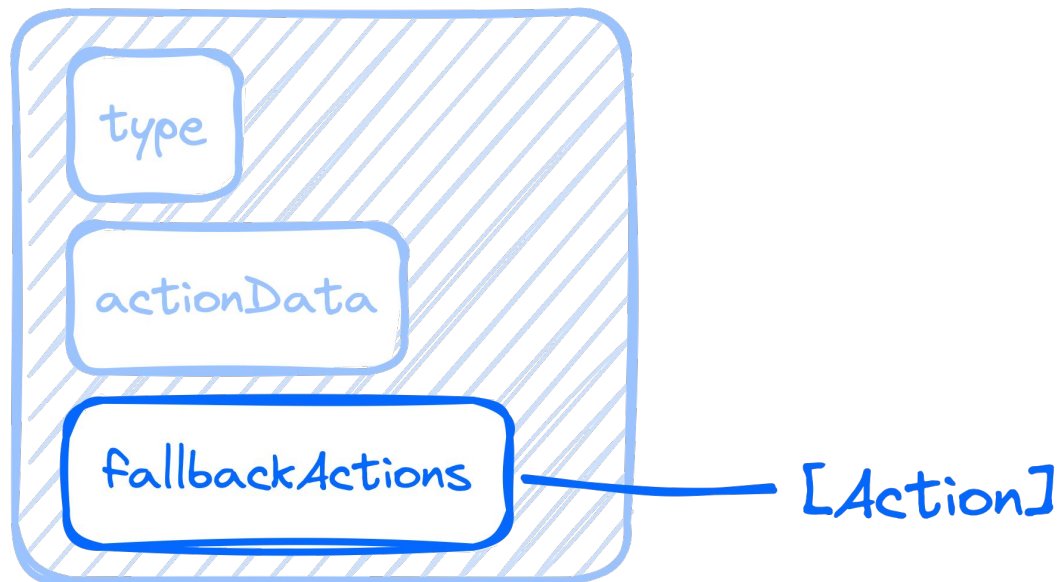
- показать информер об ошибке
- сбросить состояние загрузки

## Как это сделать?

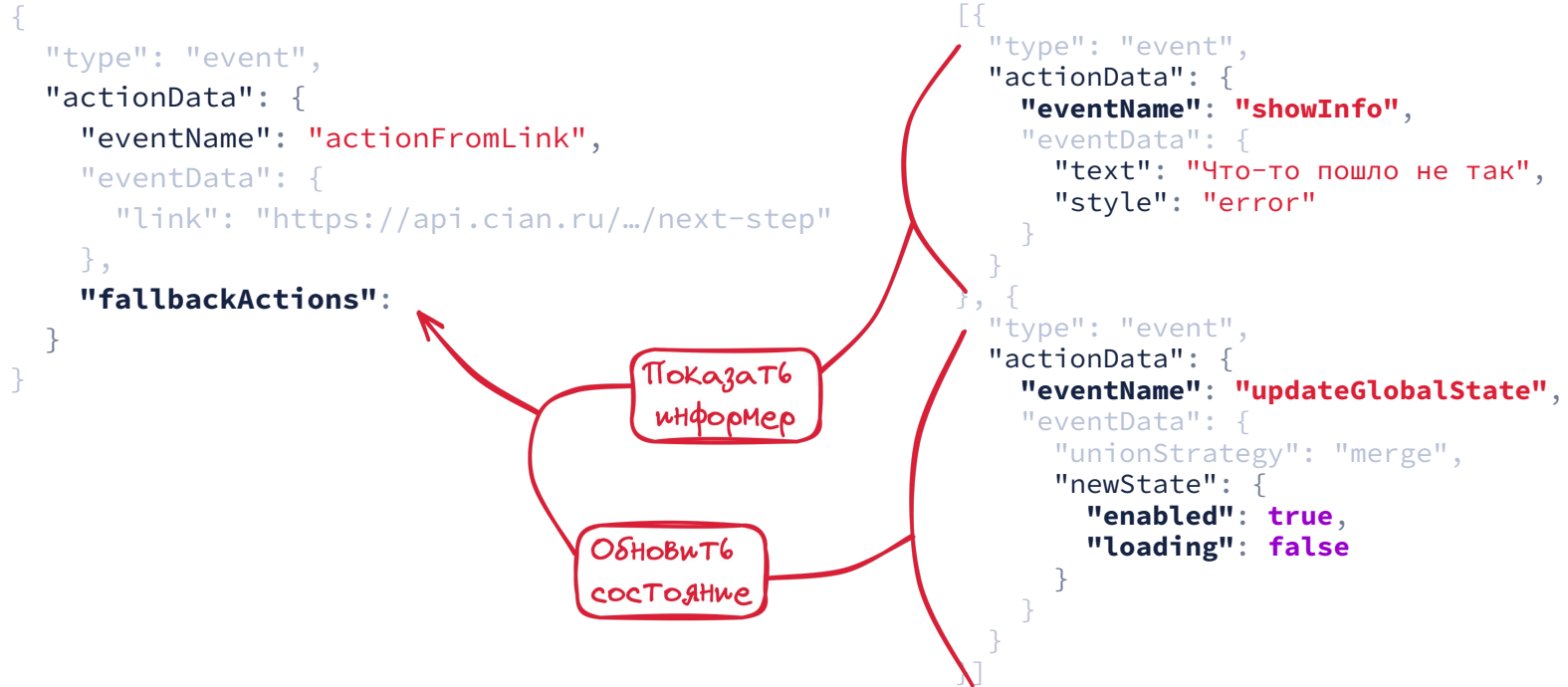
## Action



Action



# Обработка ошибки соединения



13:43 циан

← Продажа квартиры ...

### Параметры квартиры

Количество комнат

Студия 1 2 3 4 5

6+ Свободная планировка

Общая площадь

М<sup>2</sup>

Жилая площадь

85 М<sup>2</sup>

Как посчитать площадь ▾

Кухня

12 М<sup>2</sup>

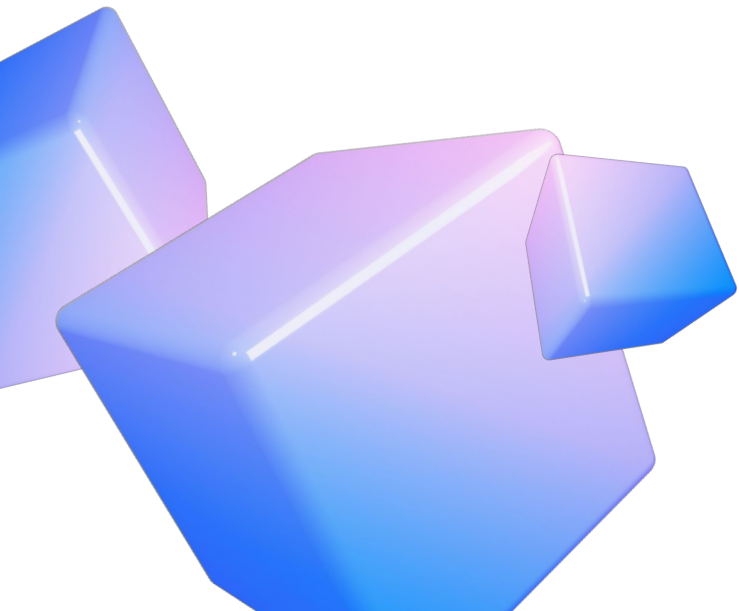
Высота потолков

3 М

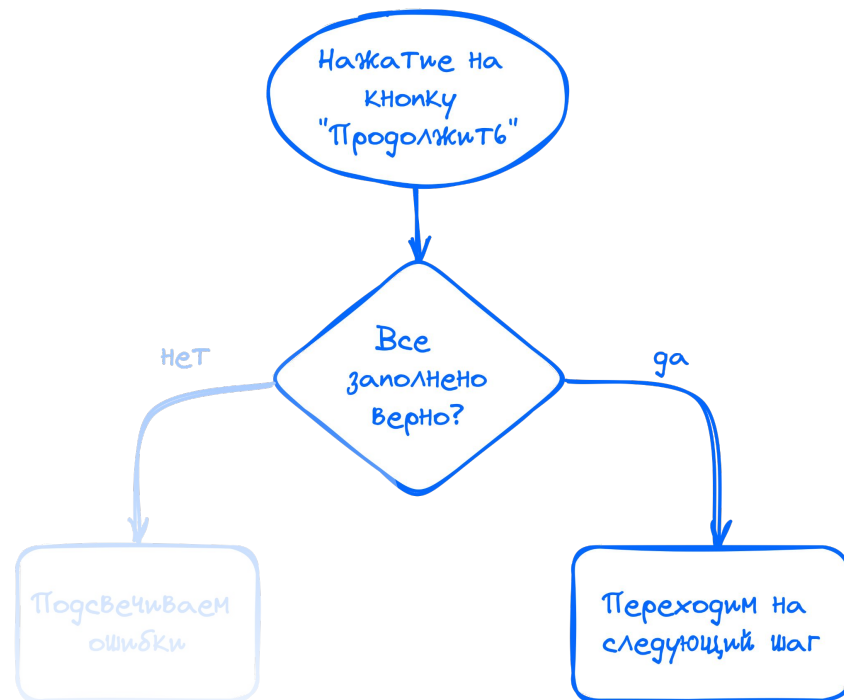
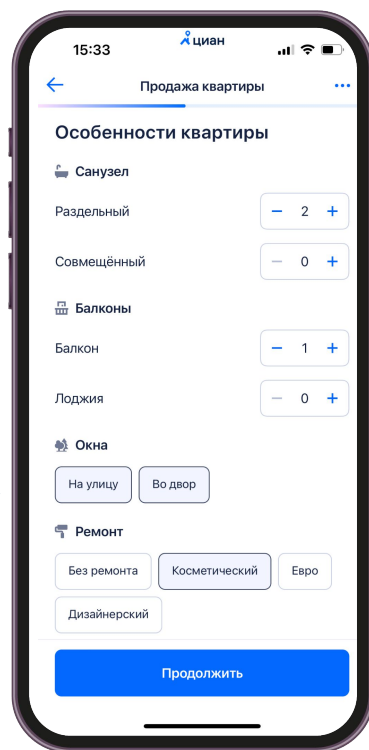
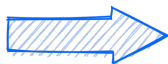
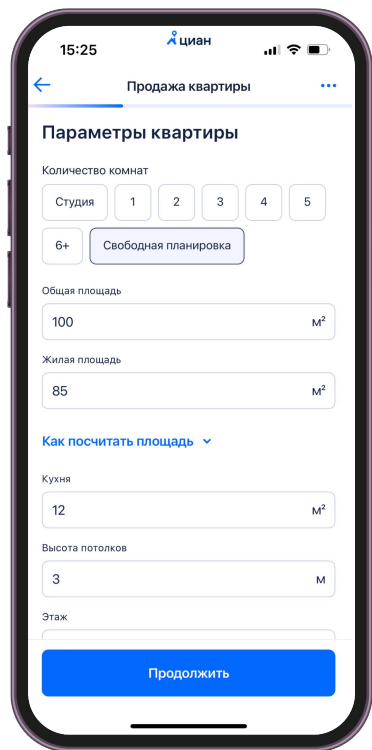
Этаж

Продолжить

- Инпуты
- Взаимосвязанные элементы
- Форматирование
- Валидация
- Переходы



# Переход



# Переход

```
{
  "actions": [{
    "type": "event",
    "actionData": {
      "eventName": "newScreen",
      "eventData": {
        "stack": "horizontal",
        "json": <JSON следующего экрана>
      }
    }
  ]
}
```

*Server* →



14:41 циан

← Продажа квартиры ...

### Параметры квартиры

Количество комнат

Студия 1 2 3 4 5

6+ Свободная планировка

Общая площадь

100 м<sup>2</sup>

Жилая площадь

85 м<sup>2</sup>

Как посчитать площадь ▾

Кухня

12 м<sup>2</sup>

Высота потолков

3 м

Этаж

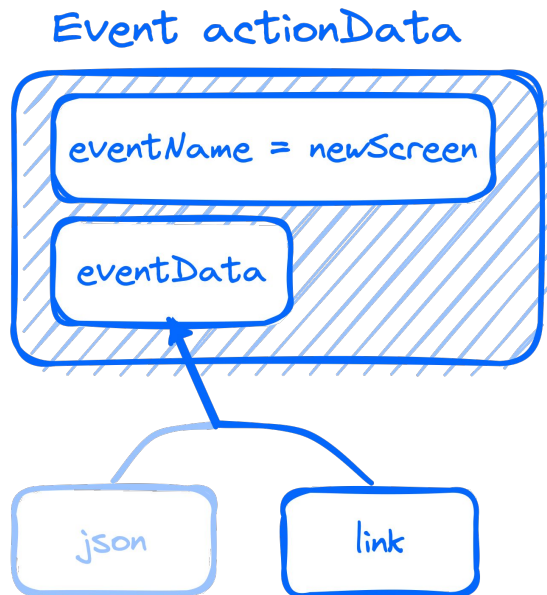
Продолжить

А можно ли открыть экран без  
предзагруженных данных?

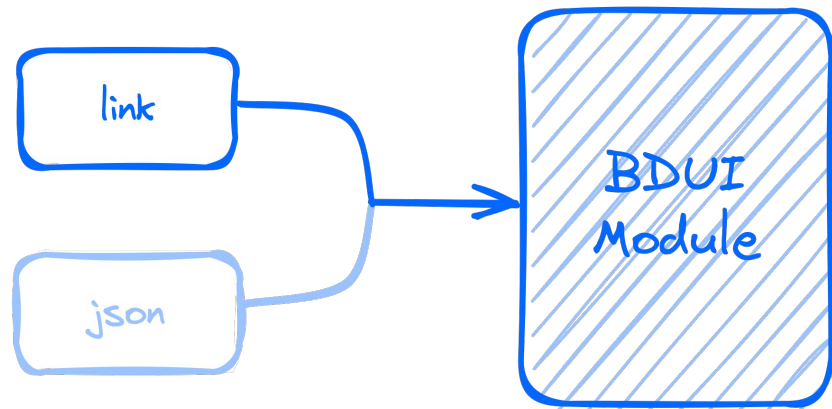


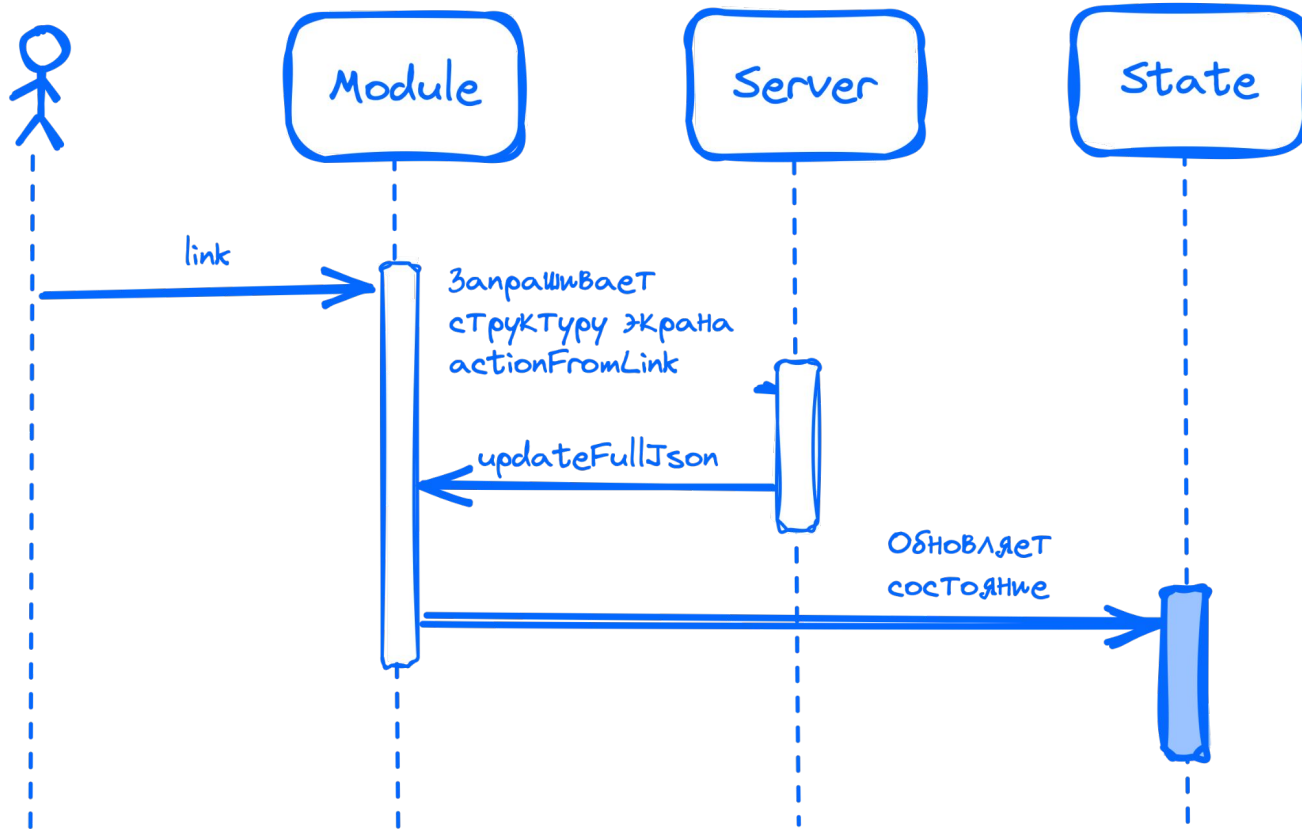
# Переход с последующей загрузкой

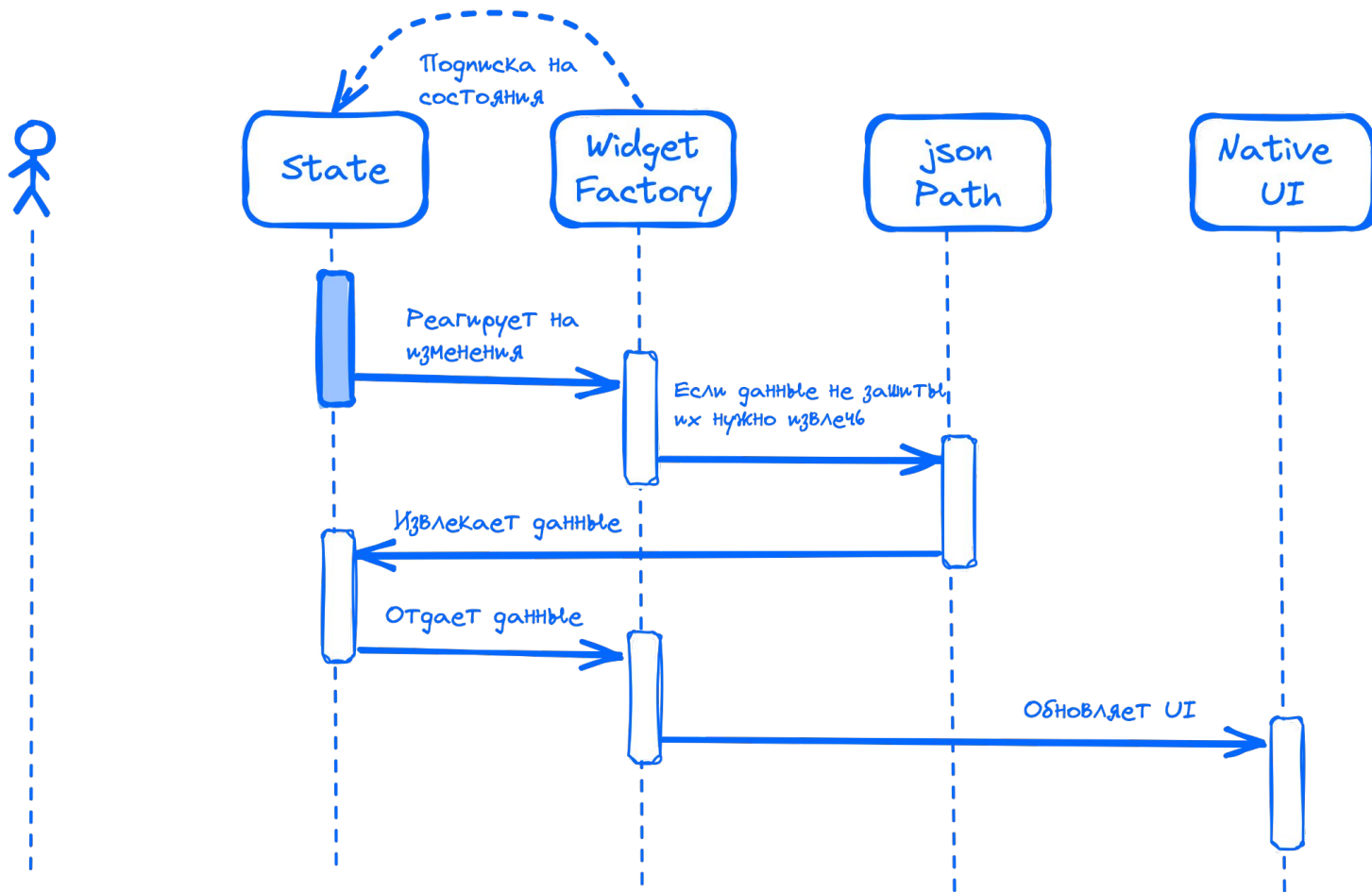
С другого экрана BDUI



Создание модуля из натива

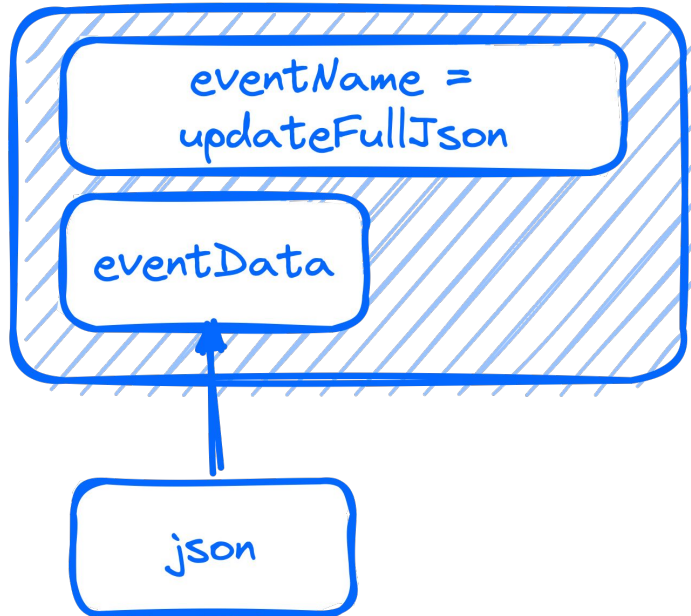






# Event Action updateFullJson

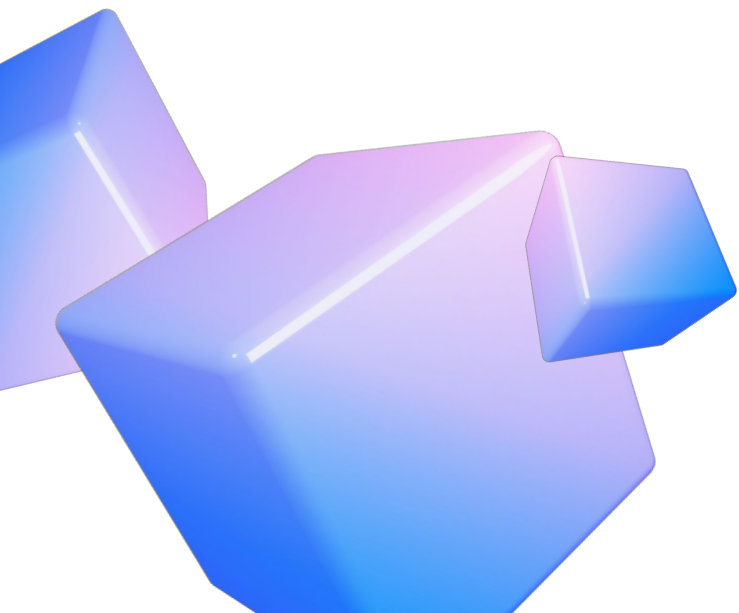
Event actionData

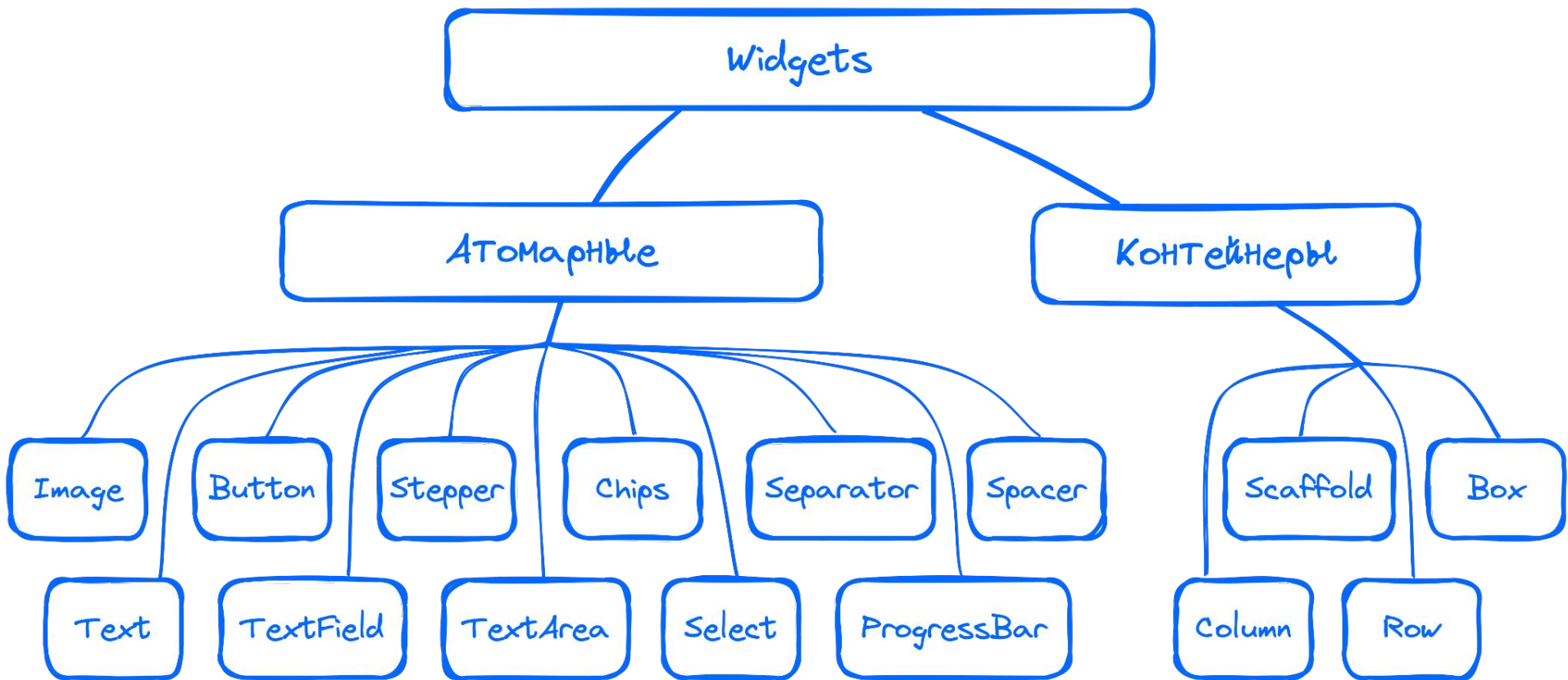


Используется для:

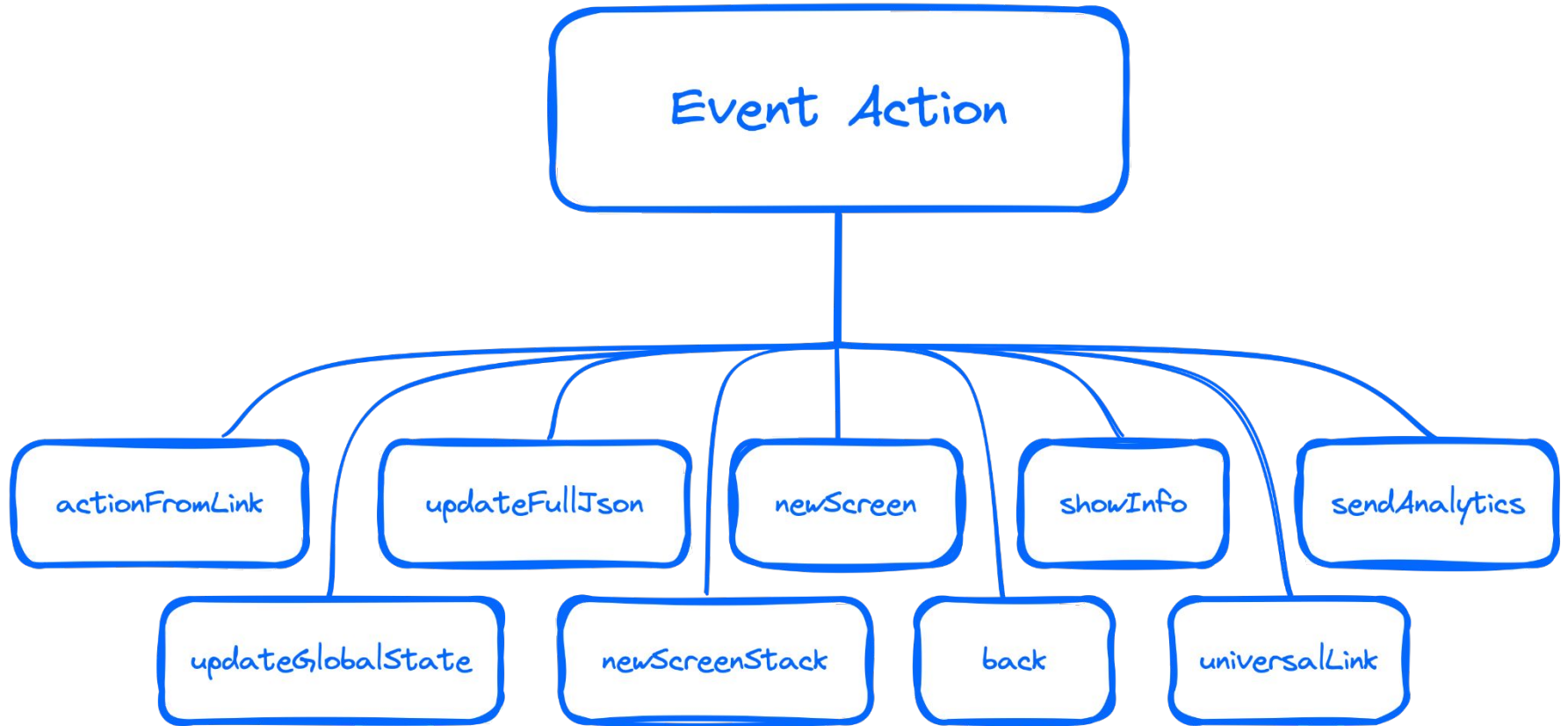
- первой отрисовки экрана
- последующей перерисовки

- Инпуты
- Взаимосвязанные элементы
- Форматирование
- Валидация
- Переходы









А вдруг не хватит?



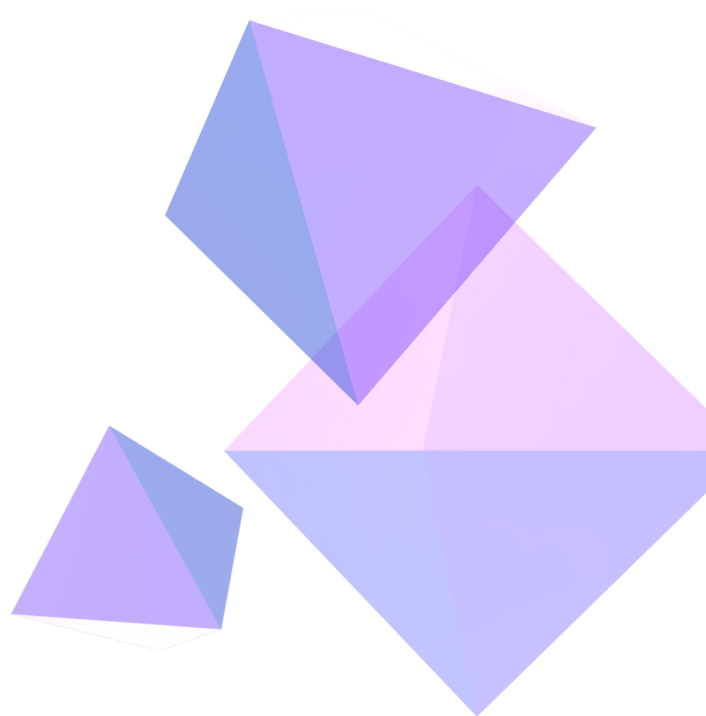
# Вертолёты

---

## Это компоненты, которые:

- не входят в библиотеку стандартных компонентов BDUI
- не универсальны, заточены под конкретный продуктовый кейс
- разрабатываются и поддерживаются продуктовыми командами

Можно инжектировать в BDUI-модуль при его создании



# Вертолёты-виджеты

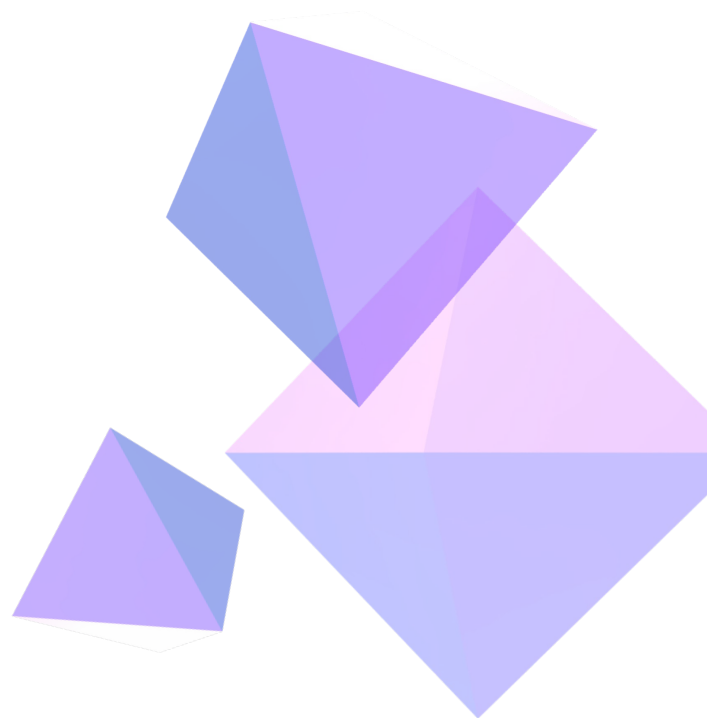
## WidgetFactory

```
func canMakeWidget(model) -> Bool  
func makeWidget(model,  
dependencies)
```

CoreWidgetFactory

ActionFactory

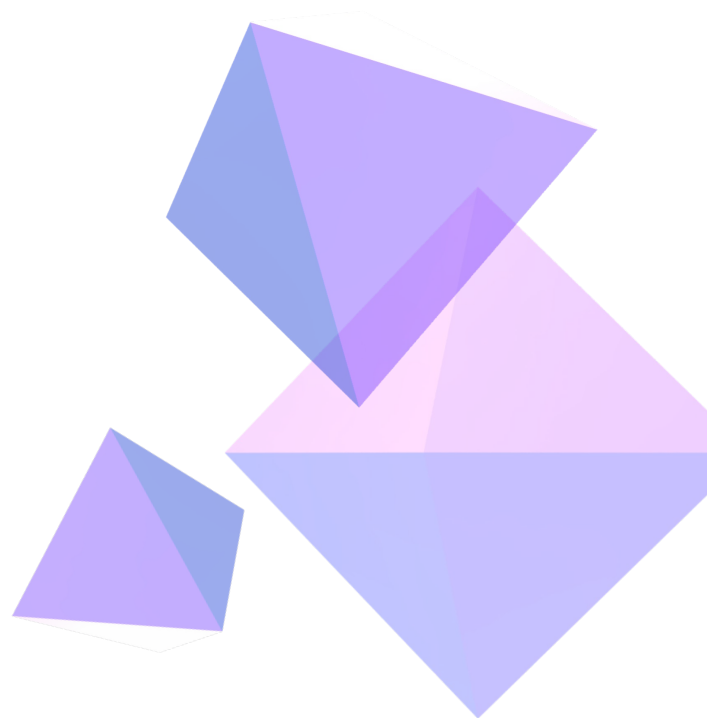
ModifierFactory



# Вертолёты-действия

## EventHandler

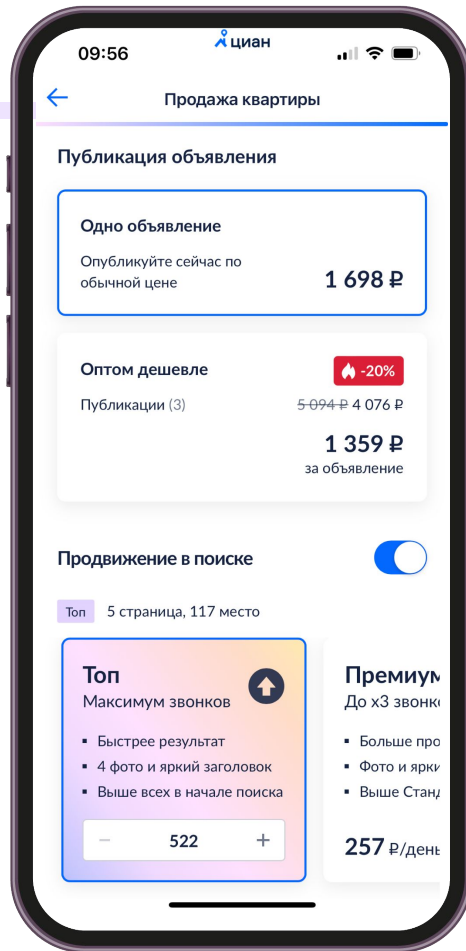
```
func canHandle(event) -> Bool  
func handle(event,  
            globalState)
```



# Вертолёты

## Форма выбора услуг:

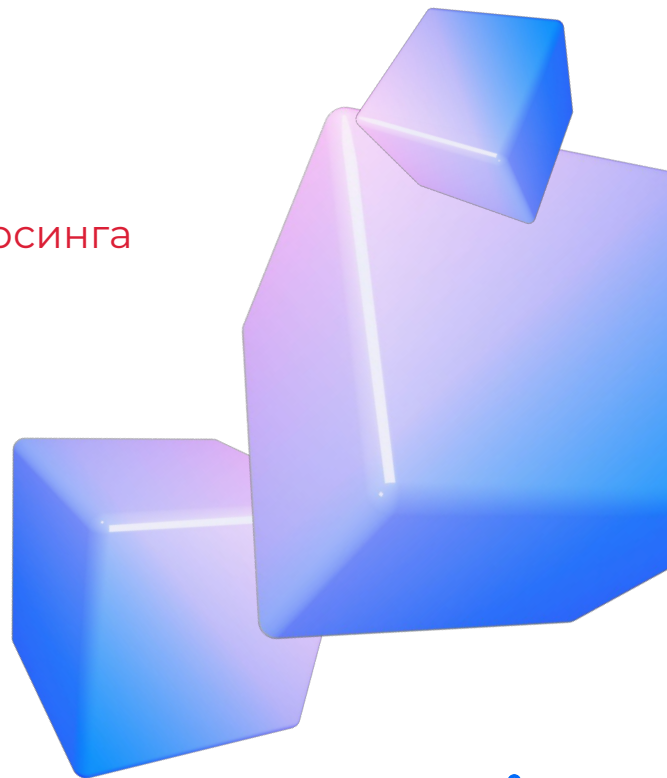
- **бэкенд:** инжектируется огромный микросервис
- **фронтенд:** инжектируется экран другого несвязанного модуля



# Стратегия ошибок

---

- Незнакомый **виджет** — ошибка парсинга
- Незнакомый **экшен** — ошибка парсинга
- Незнакомый **модификатор** — игнорируется
- Битое или пустое обязательное поле — ошибка парсинга
- Битое опциональное поле — ошибка парсинга



# Версионирование

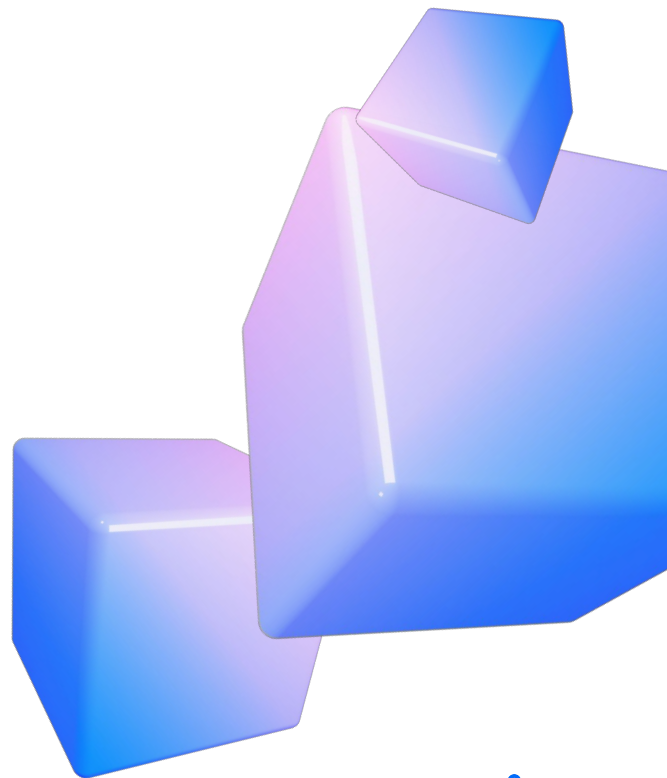
## Общая версия BUI в формате `<major>.<minor>`:

- совокупна из версий Core-компонентов: виджетов, модификаторов, экшенов
- вертолёты HE влияют на версию
- поднимаем major в случае обратно несовместимых изменений

## Версии синхронны между iOS и Android

## Дополнительные возможности версионирования:

- ориентироваться на версию приложения
- увеличить версию API-ручки





# Версионирование

---

Feature A

BDUI Module v1.0

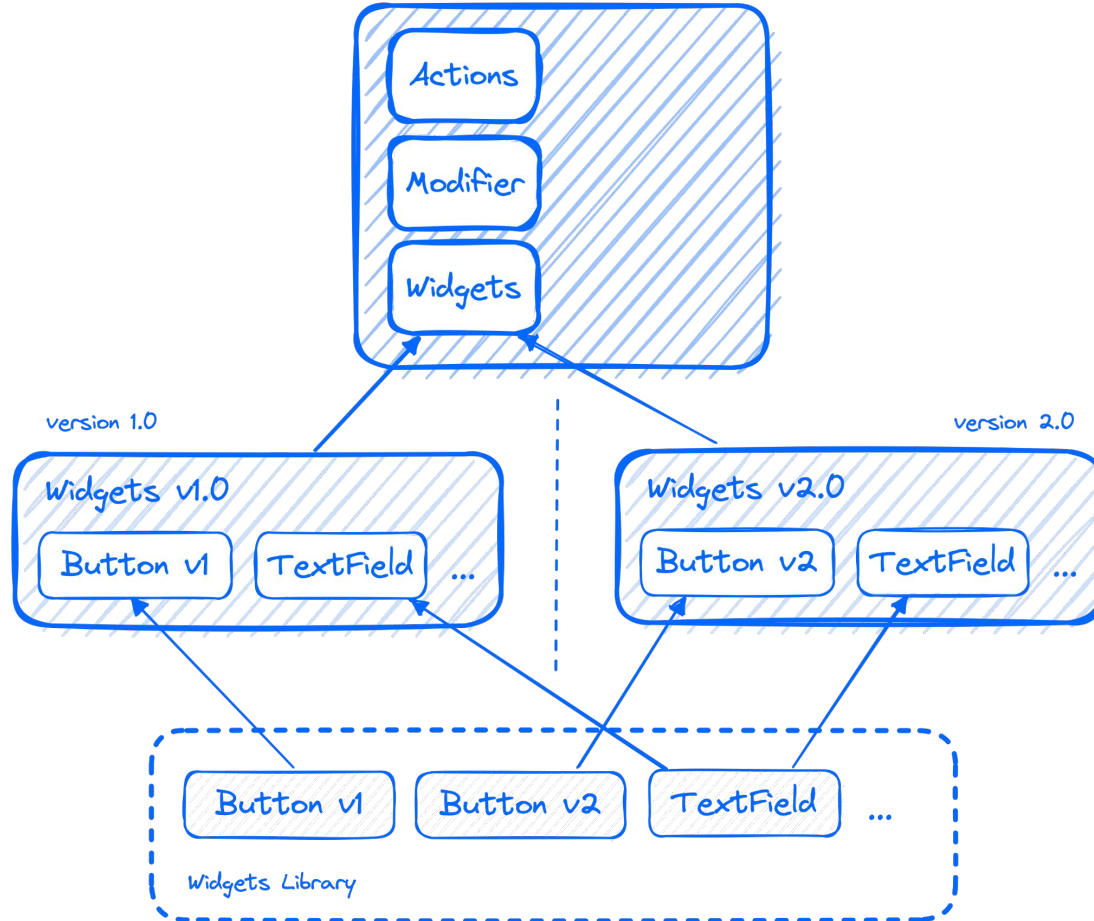
Feature B

BDUI Module v2.0

Зачем так?

- нет возможности синхронно обновлять все микросервисы
- снижаем риск заафферить существующие фичи

# BDUI Module

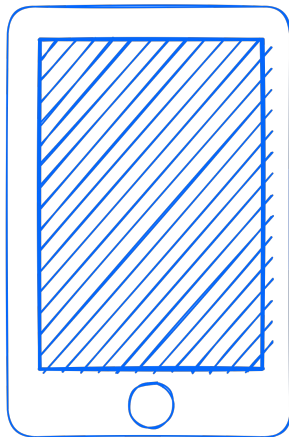


# Что умеет?

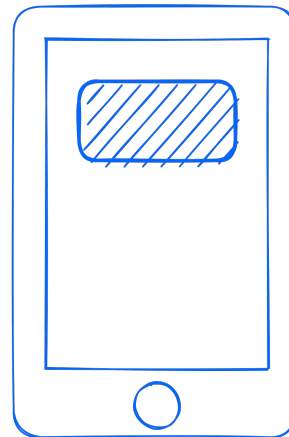
- Построение UI
- Валидация
- Навигация
- Аналитика
- Клиент-серверное взаимодействие
- Вертолёты



# Модуль может быть



Отдельным экраном



Вьюшкой, встроенной в натив

# Технологии

---

## Android

 Kotlin

 UI: Jetpack Compose

 JS: V8

## iOS

 Swift

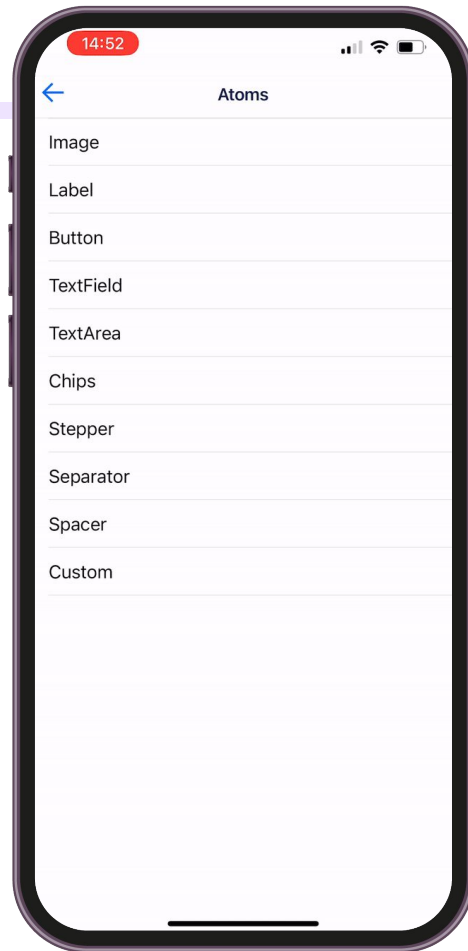
 UI: UIKit

 JS: JavaScriptCore

# Приложение-песочница

## Виджеты:

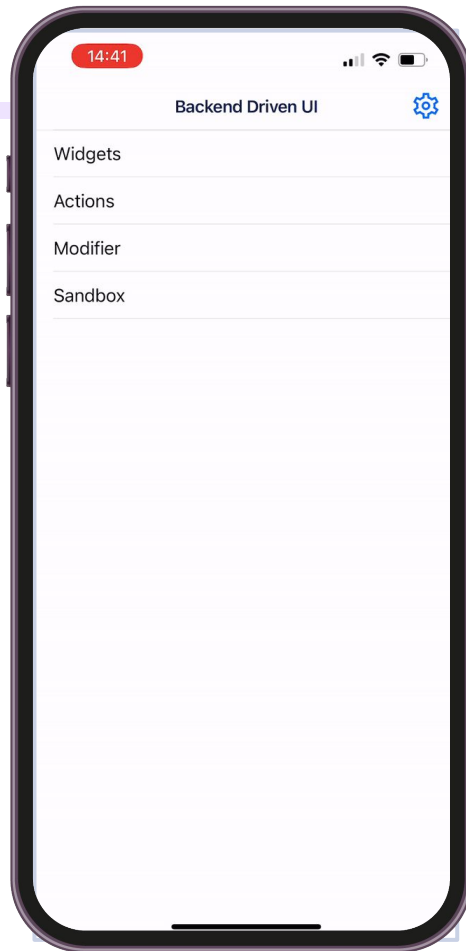
- сконфигурировать под макет
- получить JSON-описание



# Приложение-песочница

## Модули:

- ввести JSON в текстовое поле
- дернуть ручку API
- загрузить из любого удаленного JSON-файла
- Hot Reload при редактировании локального файла



# Что болело?

---

Набрались терпения 😁

**iOS:** повторить реализацию поведения Compose

**Android:** наверстать отставание в дизайн-системе

**Backend:**

- Библиотеки по спецификации OpenAPI 2.0
- Dev-testing
- Нет готового механизма тестировать JS-код
- Сильное перераспределение работ в сторону бэкенда





Понравилось ли нам?

Поначалу дорого и больно, но...



# Посмотрим видео

Сейчас провожу интервью с риелтором. Я его спрашиваю про форму подачи, а он всё хвалит, что стало удобнее и понятнее. Подсказки там всякие появились.

The screenshot shows the 'Продажа квартиры' (Apartment Sale) form in the Циан mobile app. The form is titled 'Расположение' (Location) and includes the following fields and options:

- Адрес объекта** (Object Address): Россия, Москва, Электrozаводская улица, 27с8. A note below indicates: 'Редактировать адрес можно в течение 2 дней после публикации' (You can edit the address within 2 days after publication).
- Номер квартиры** (Apartment Number): An empty text input field.
- Кадастровый номер** (Cadastral Number): 00:00:0000000:0000.
- Зачем это указывать** (Why indicate this?): A dropdown menu.
- Тип недвижимости** (Property Type): Two buttons: 'Квартира' (Apartment) and 'Апартаменты' (Apartments).
- Чем они отличаются** (How they differ): A dropdown menu.
- Продолжить** (Continue): A large blue button at the bottom.

# Посмотрим технические метрики



## 99 перцентиль:

**124 мс** — среднее время TTI экрана

**4,87 мс** — среднее время выполнения JS

**8,45 мс** — среднее время обновления UI

# Итоги первых релизов

Актуальный стек технологий

# 75%

задач — на бэкенде

# 61% → 64%

Прирост конверсии в публикацию

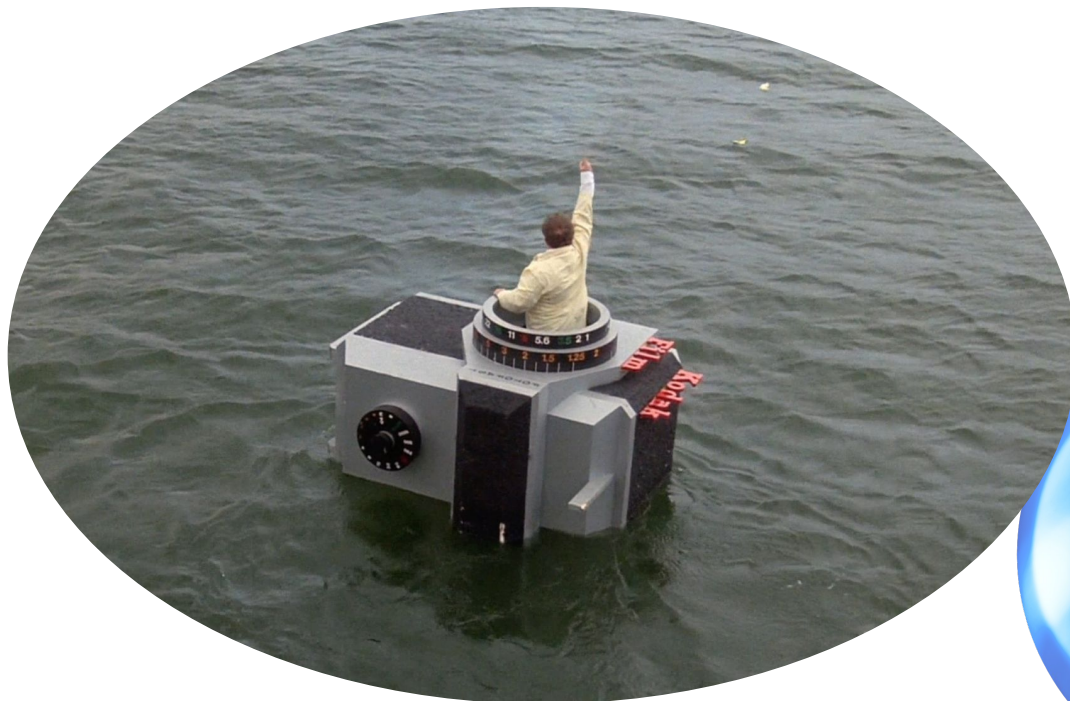
Время до  
production



1 дня для  
75% задач



**Наша цель  
достигнута!**



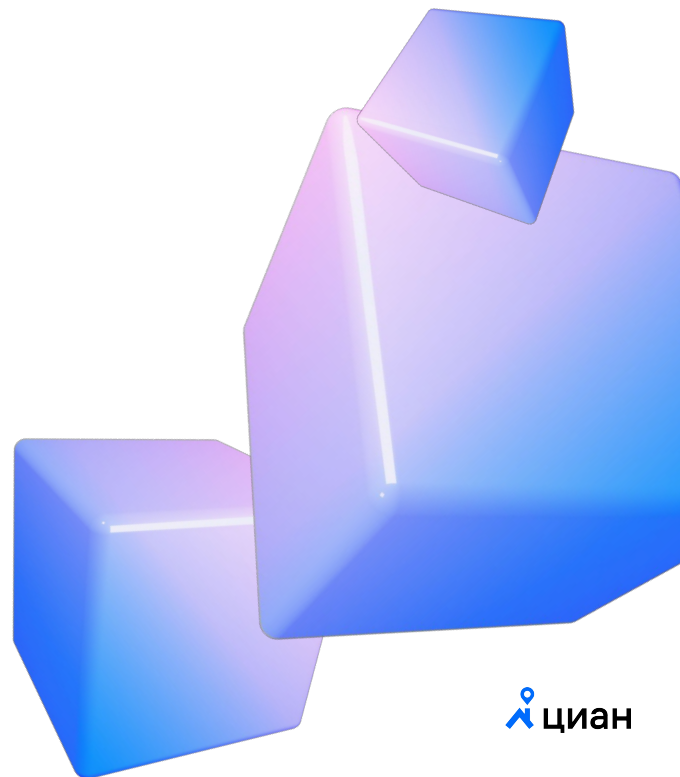
Планируем ли продолжать?



# Да или нет VDUI в фиче?

---

Продукту требуется максимально быстрый TTM — **ДА**



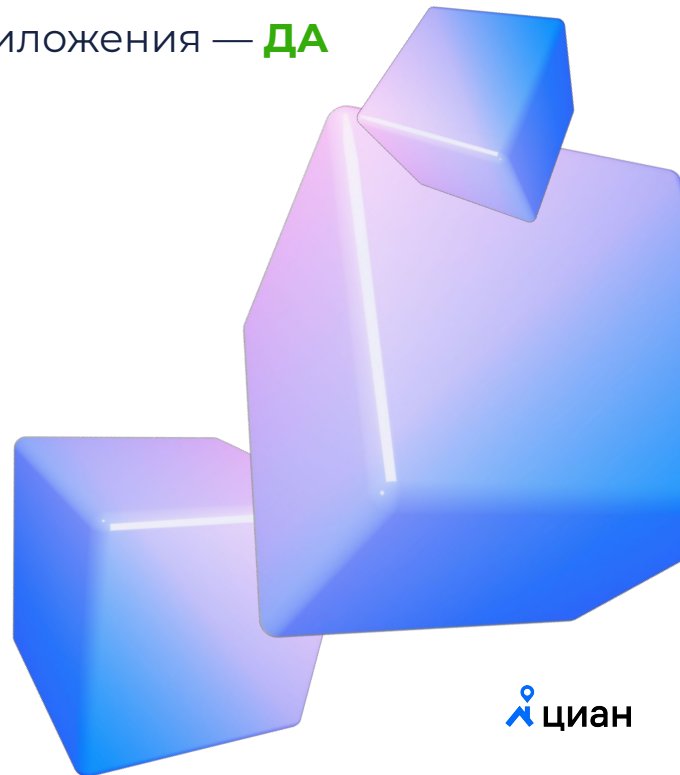


# Да или нет BDUI в фиче?

---

Продукту требуется максимально быстрый TTM — **ДА**

Функционал должен быть актуален без обновления приложения — **ДА**



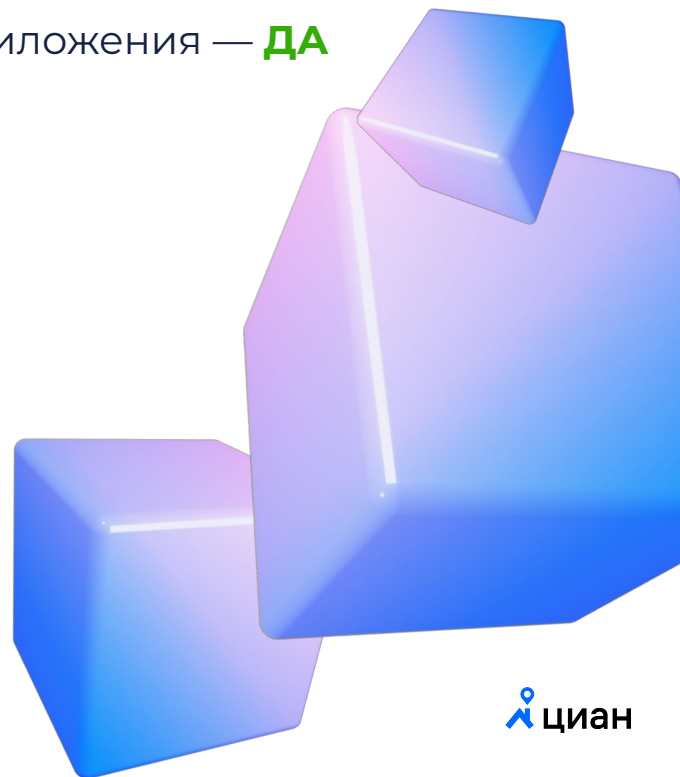
## Да или нет BDUI в фиче?

---

Продукту требуется максимально быстрый TTM — **ДА**

Функционал должен быть актуален без обновления приложения — **ДА**

Планируется много экспериментов и изменений — **ДА**



# Да или нет BUI в фиче?

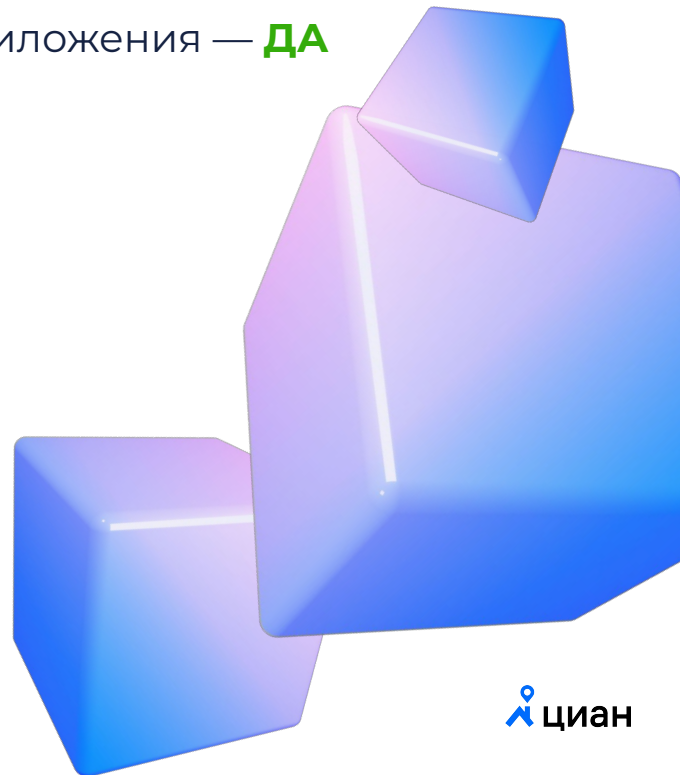
---

Продукту требуется максимально быстрый TTM — **ДА**

Функционал должен быть актуален без обновления приложения — **ДА**

Планируется много экспериментов и изменений — **ДА**

UI по дизайн-системе — **ДА**



# Да или нет BUI в фиче?

---

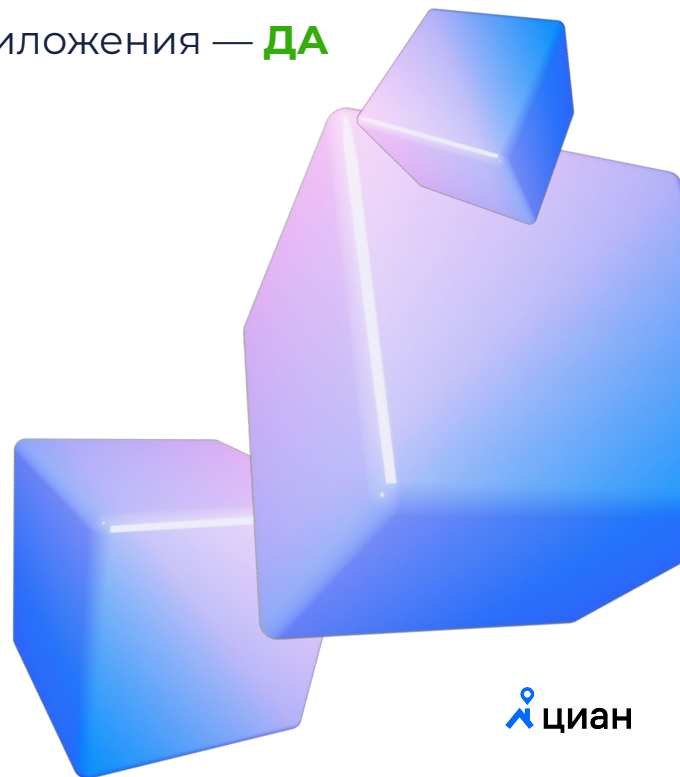
Продукту требуется максимально быстрый TTM — **ДА**

Функционал должен быть актуален без обновления приложения — **ДА**

Планируется много экспериментов и изменений — **ДА**

UI по дизайн-системе — **ДА**

Есть сложные жесты или анимации — **НЕТ**



# Да или нет BUI в фиче?

---

Продукту требуется максимально быстрый TTM — **ДА**

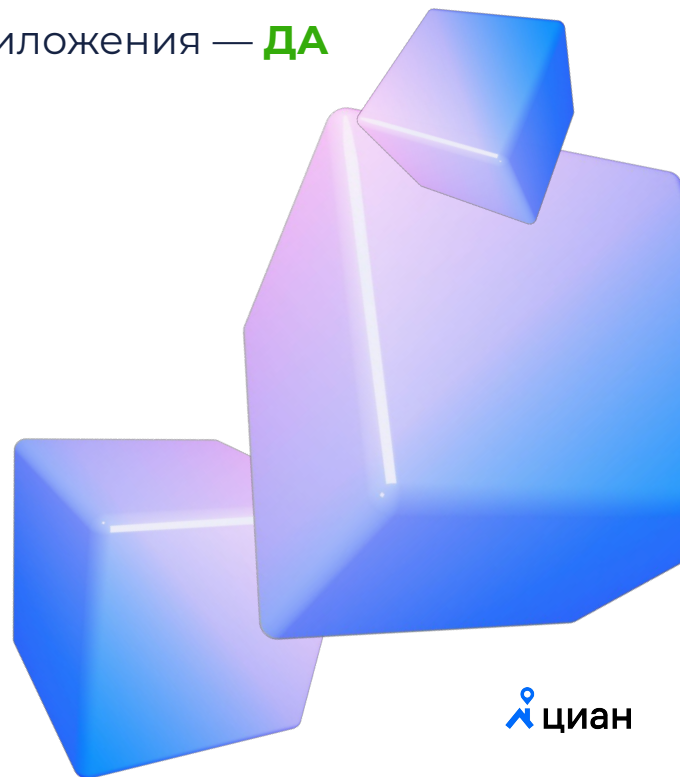
Функционал должен быть актуален без обновления приложения — **ДА**

Планируется много экспериментов и изменений — **ДА**

UI по дизайн-системе — **ДА**

Есть сложные жесты или анимации — **НЕТ**

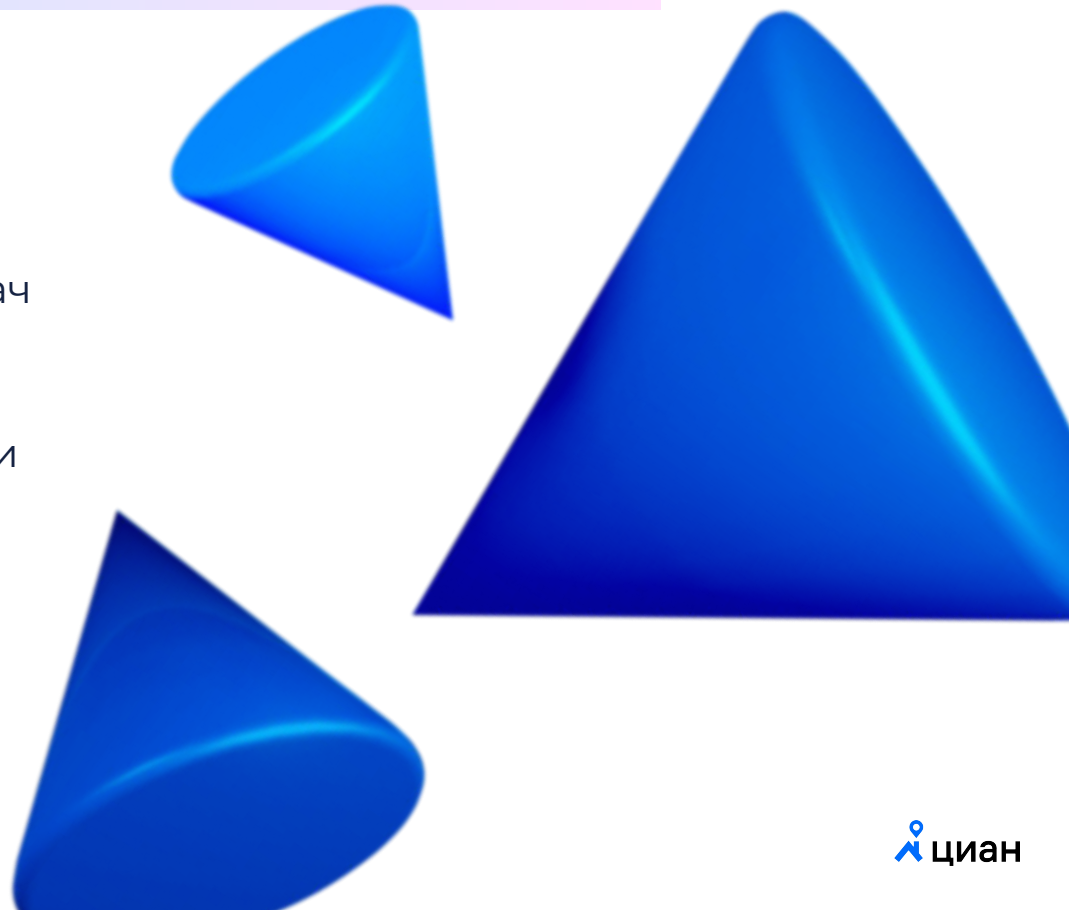
Ресурс бэкенда — ...



# Нюансы BDUI

---

- Практически всю логику нужно перенести на бэкенд
- Вынос логики развязывает руки
- Сильное перераспределение задач в сторону бэкенда
- Мы не рассматриваем BDUI как средство удешевления разработки
- Подходит не для всех фичей



# Выводы

---

- Глобальные новинки архитектуры требуют тщательной проработки на старте
- Переводить всё приложение на BDUI не планируем
- Архитектуру нужно выбирать под продукт
- Каждый инструмент хорош в своей области применимости

The End





# Вопросы?

Mail: [d.zherdev@cian.ru](mailto:d.zherdev@cian.ru)

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

