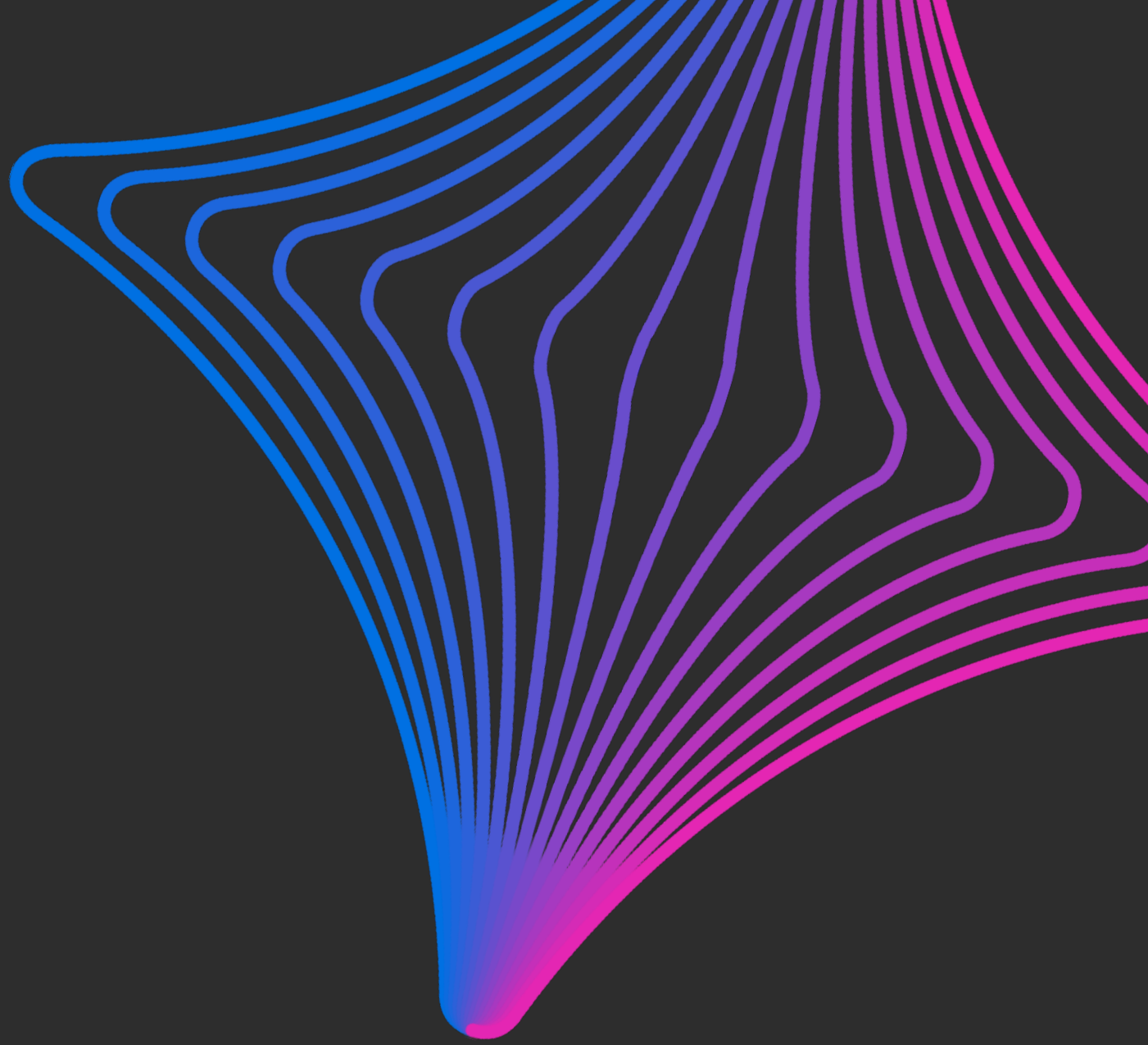
GraphQL.
Вредные советы

Алексей Хайминов
Senior Android developer

YOULABS

юла

Что такое Юла?



YOULABS

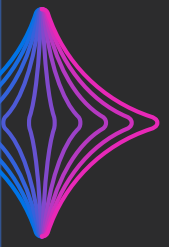
Что такое Юла?

Classified 2.0

миллиарды

документов в базе

YOULABS



Что такое Юла?

Classified 2.0

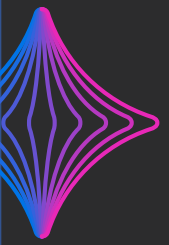
миллиарды

документов в базе

терабайты

данных

YOULABS



Что такое Юла?

Classified 2.0

миллиарды
документов в базе

>32 млн
активных объявлений

терабайты
данных

YOULABS

Что такое Юла?

Classified 2.0

миллиарды
документов в базе

терабайты
данных

>32 млн
активных
объявлений

>27 млн
активных
пользователей
в месяц

YOULABS

Что такое Юла?

Classified 2.0

Переход от стартапа к крупной стабильной компании

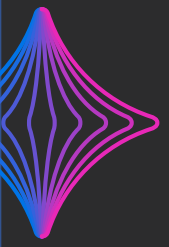
миллиарды
документов в базе

терабайты
данных

>32 млн
активных объявлений

>27 млн
активных пользователей в месяц

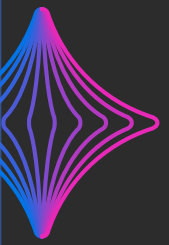
YOULABS



Что такое Юла?

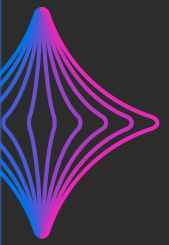
- Time to Market

YOULABS



Что такое Юла?

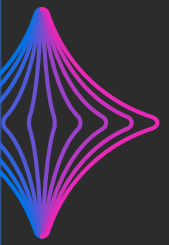
- Time to Market
- Компромиссы



Что такое Юла?

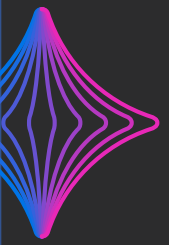
- Time to Market
- Компромиссы
- Бекэнд: PHP + MongoDb => REST

YOULABS



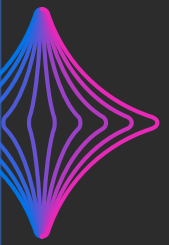
Что такое Юла?

- Time to Market
- Компромиссы
- Бекэнд: PHP + MongoDb => REST
- Android:
  - 2015 – 1-я версия = 1 разработчик + 3 месяца



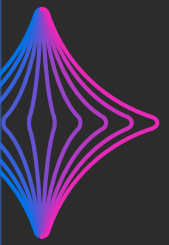
Что такое Юла?

- Time to Market
- Компромиссы
- Бекэнд: PHP + MongoDb => REST
- Android:
  - 2015 – 1-я версия = 1 разработчик + 3 месяца
  - 2017 – 2 разработчика



Что такое Юла?

- Time to Market
- Компромиссы
- Бекэнд: PHP + MongoDb => REST
- Android:
  - 2015 – 1-я версия = 1 разработчик + 3 месяца
  - 2017 – 2 разработчика
  - 2020 – 8 разработчиков



Что такое Юла?

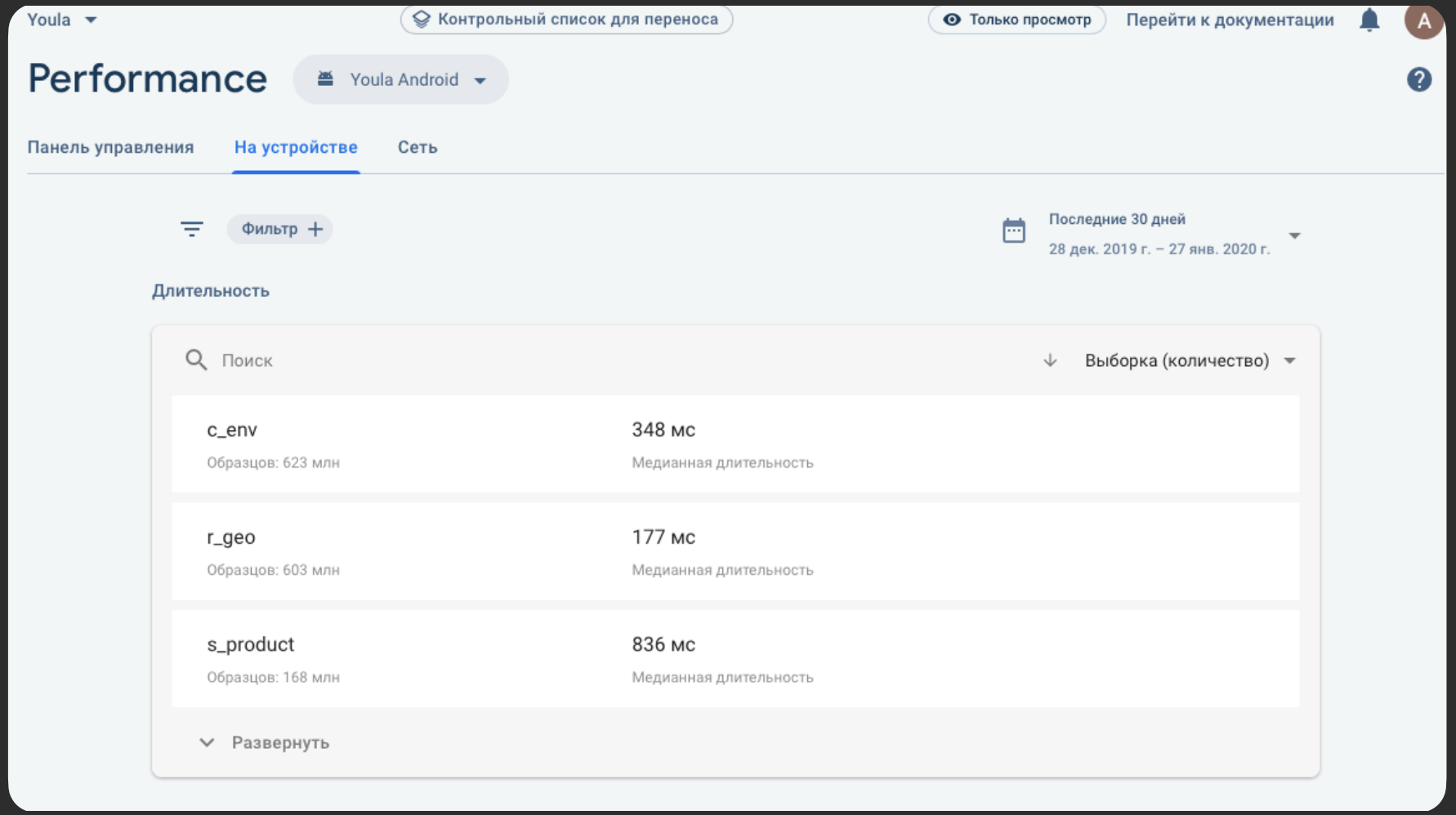
- Time to Market
- Компромиссы
- Бекэнд: PHP + MongoDb => REST
- Android:
  - 2015 – 1-я версия = 1 разработчик + 3 месяца
  - 2017 – 2 разработчика
  - 2020 – 8 разработчиков
  - 2025 - 1 ИИ ?

Вредный совет #1

Ignorance is bliss

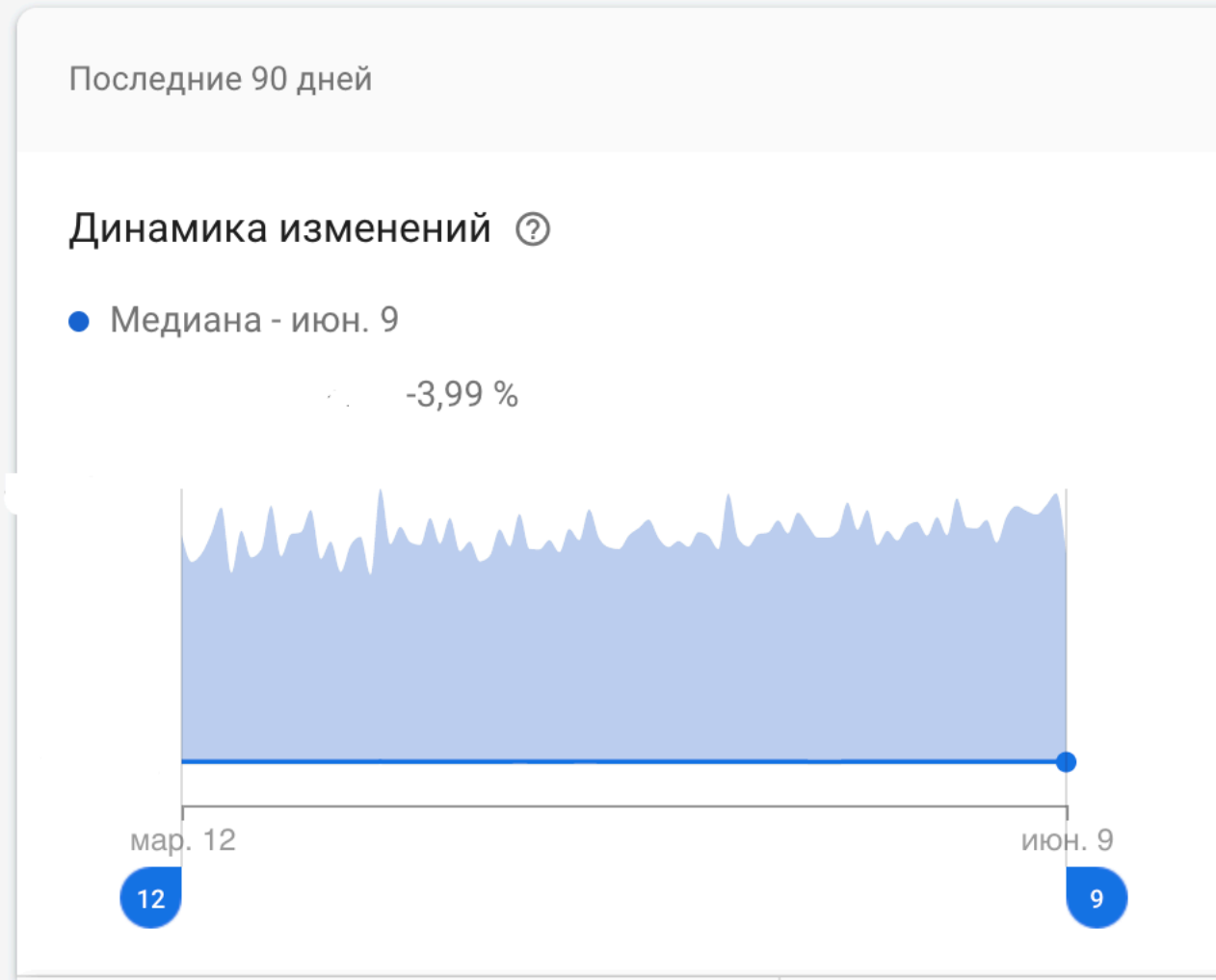
YOULABS

Performance-бизнес-метрики



YOULABS

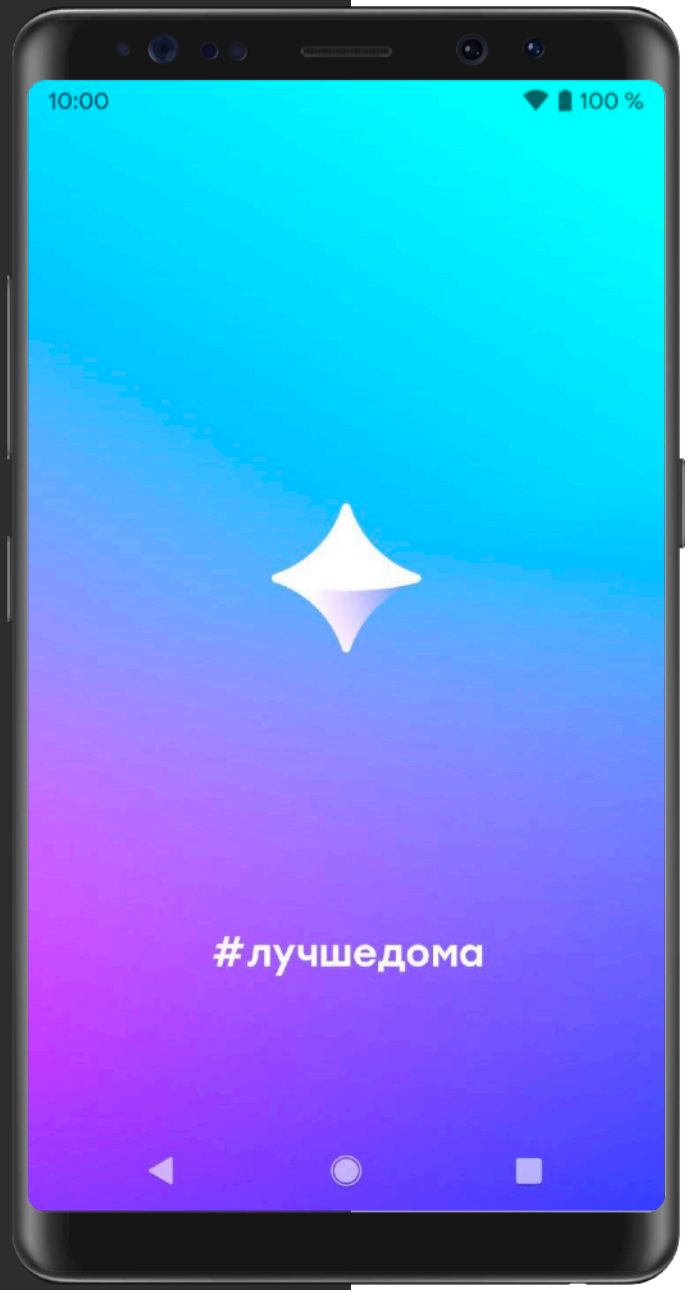
Метрика



YOULABS

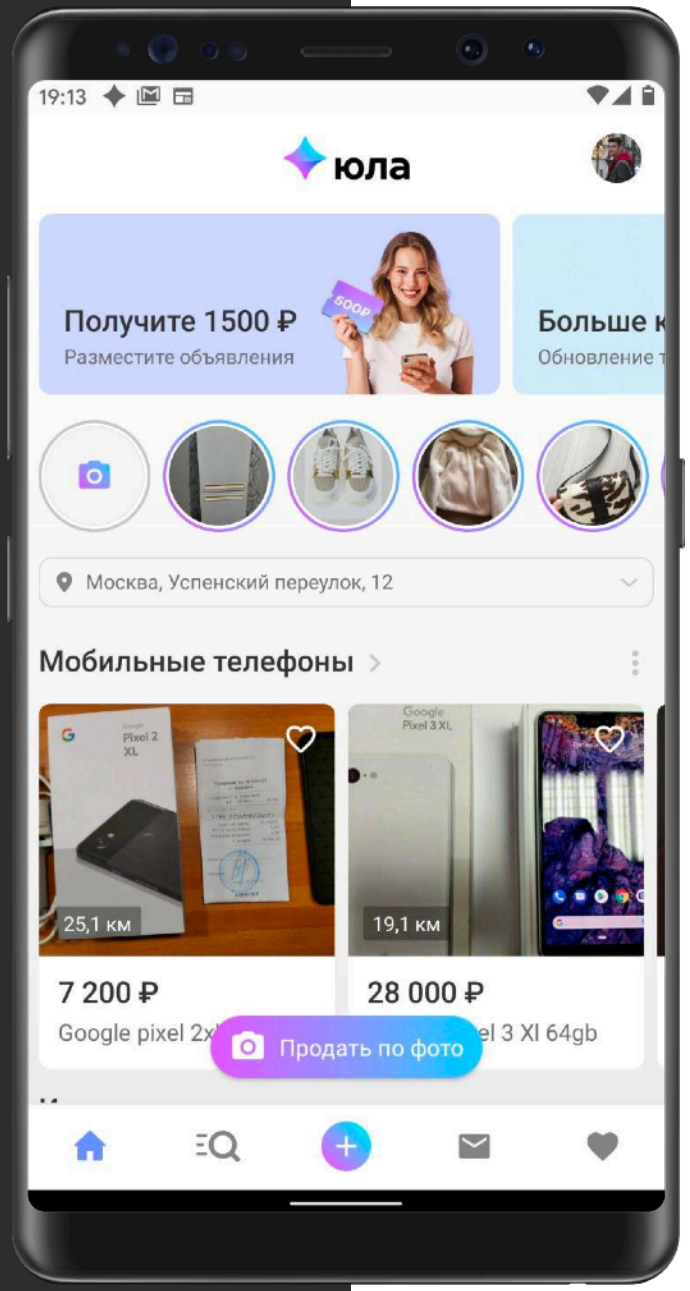
А почему так долго?

Стартовая страница приложения

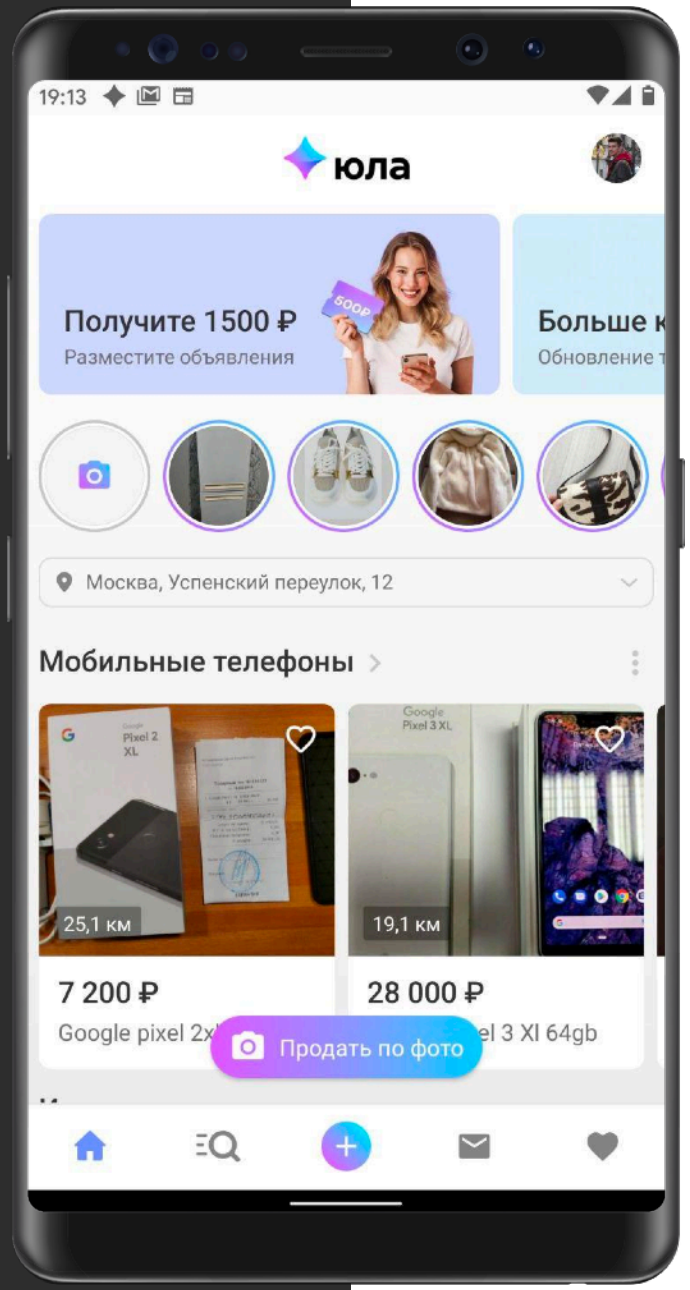


Брендированный сплэш

Стартовая страница приложения

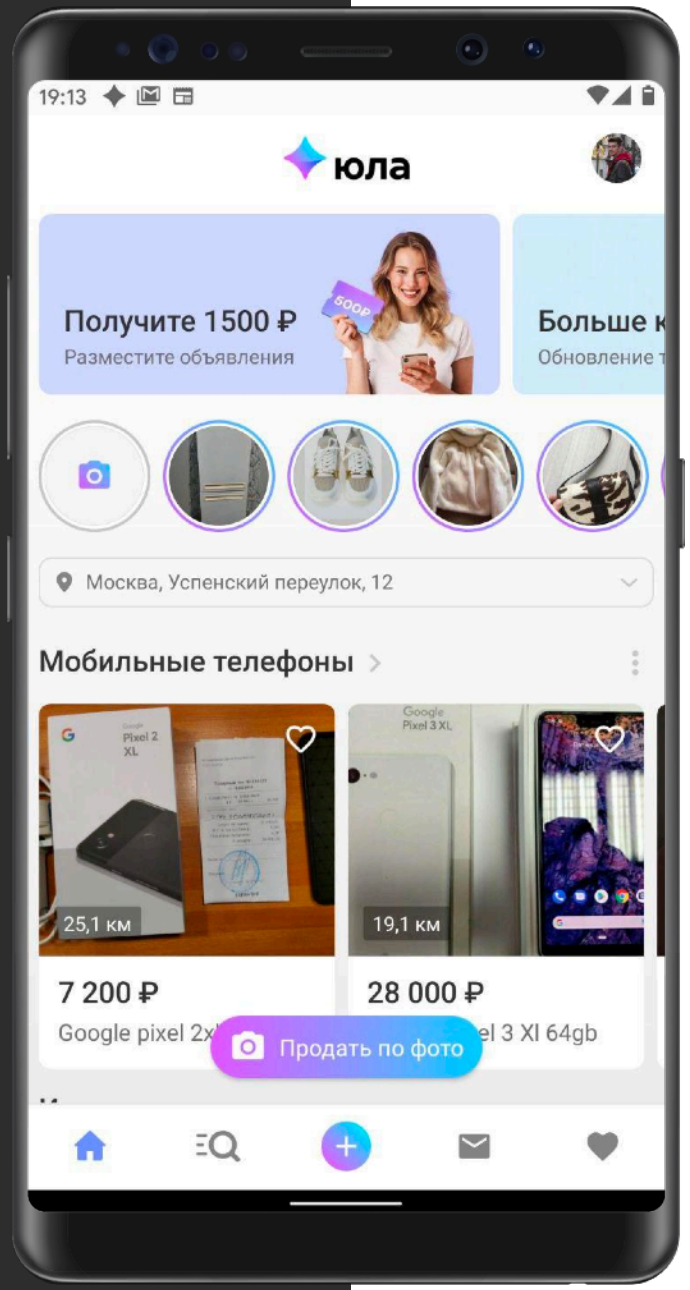


Стартовая страница приложения



Подборки товаров и баннеры

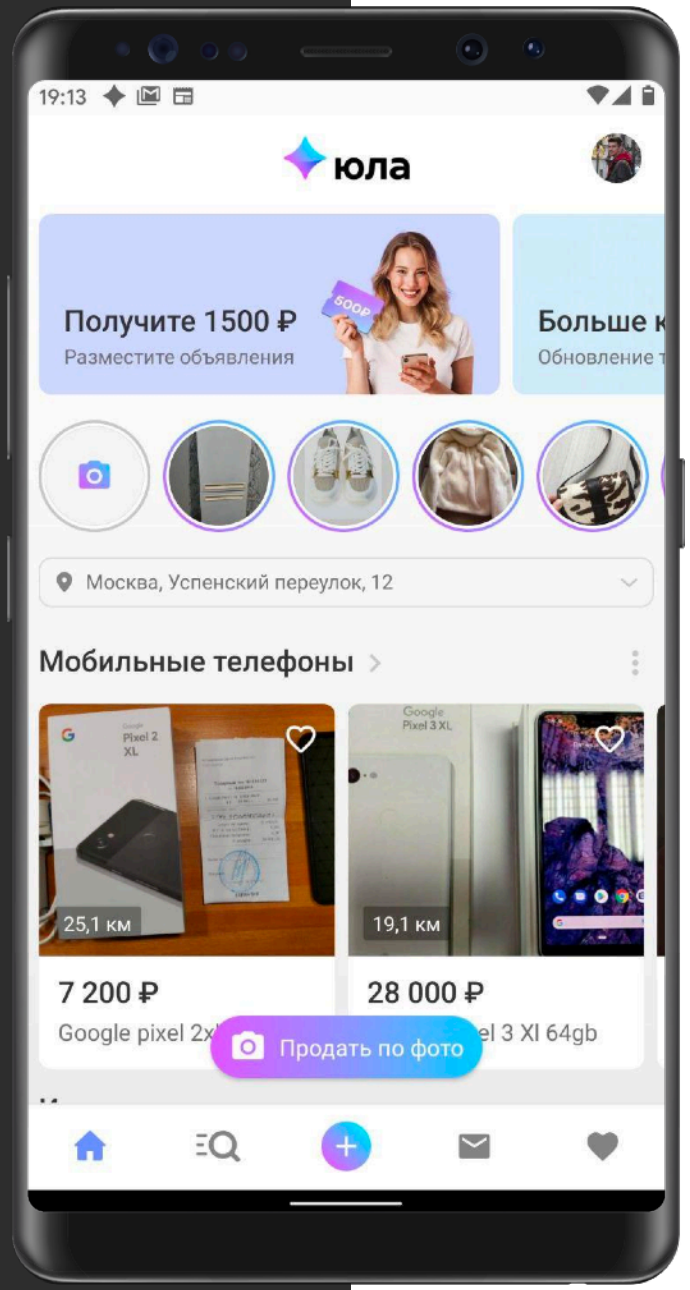
Стартовая страница приложения



Подборки товаров и баннеры

Истории

Стартовая страница приложения

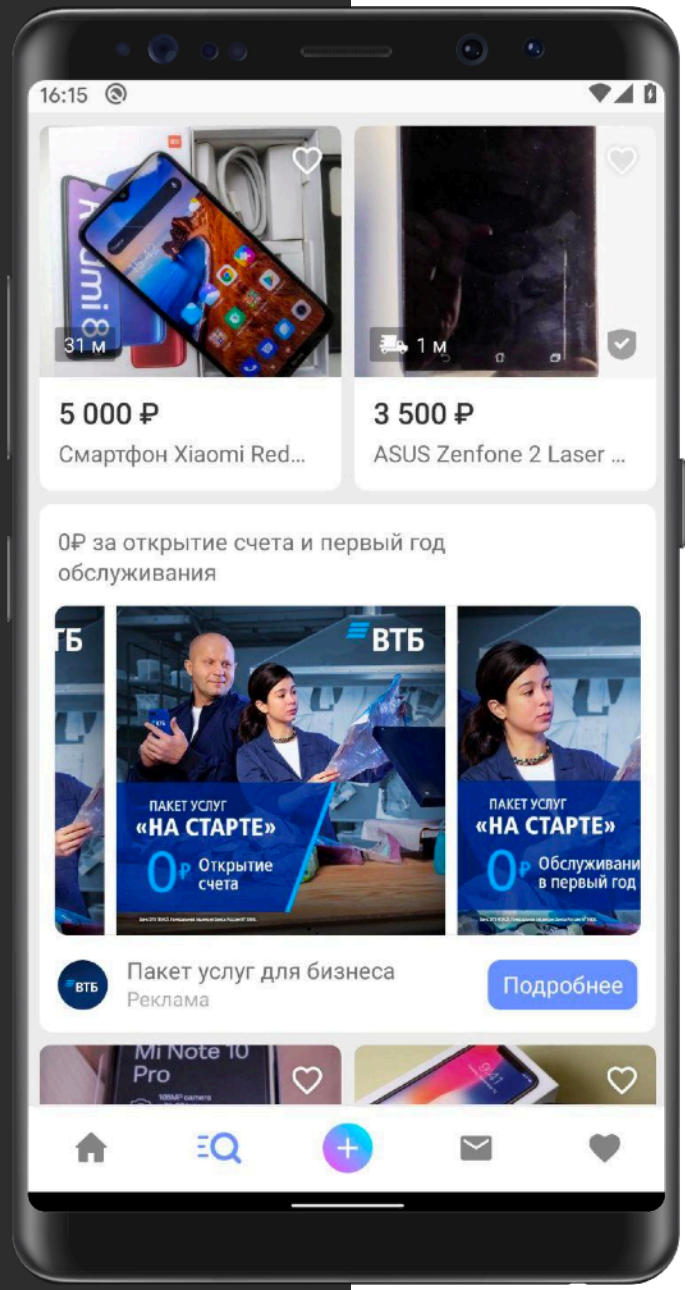


Подборки товаров и баннеры

Истории

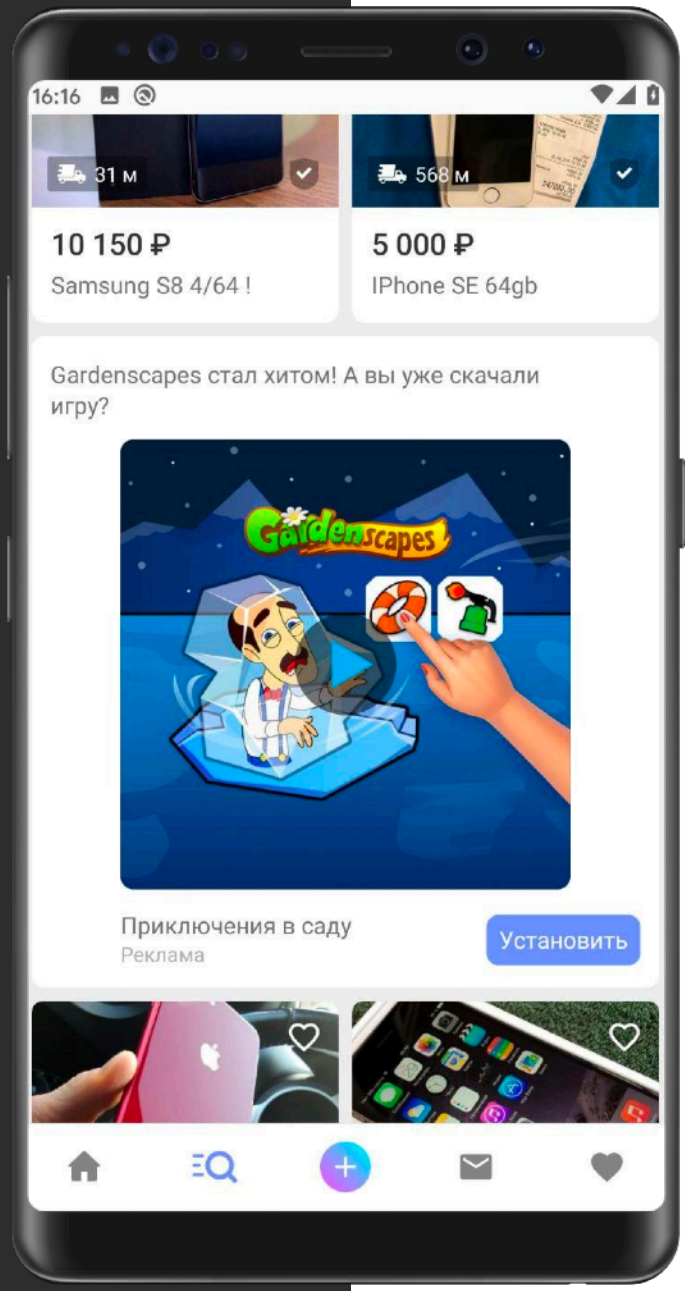
Горизонтальные карусели товаров

Стартовая страница приложения



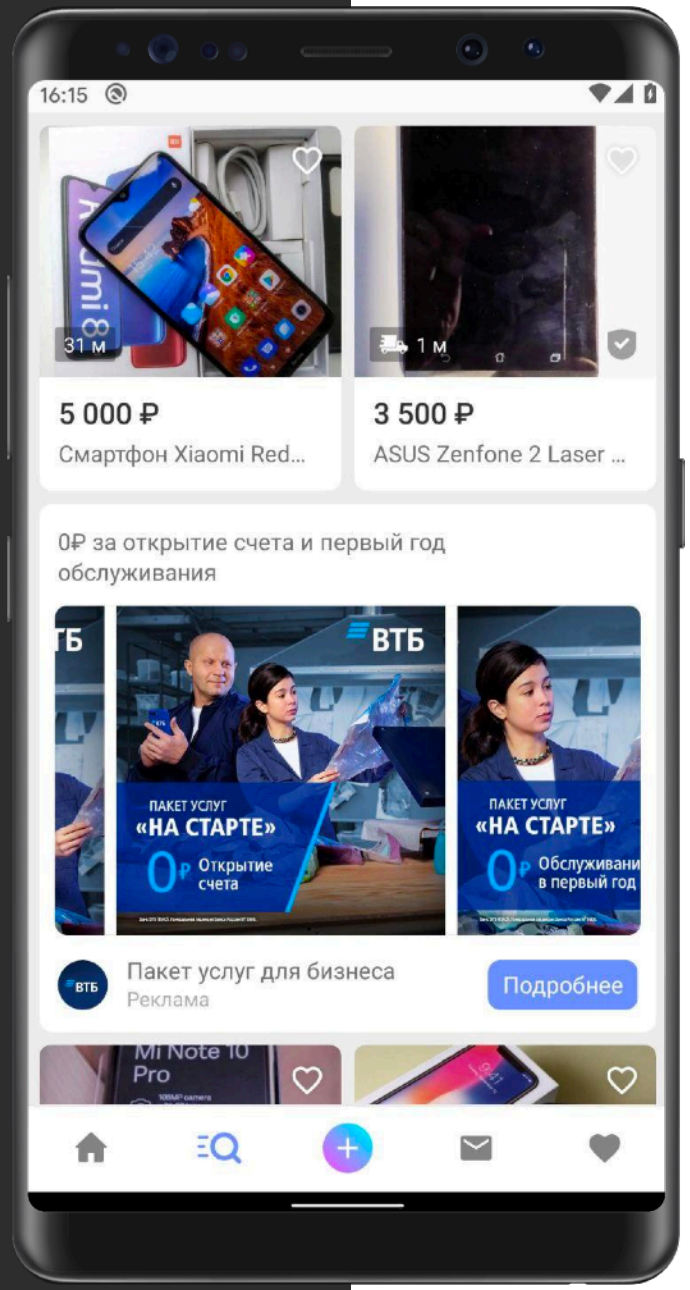
Реклама

Стартовая страница приложения

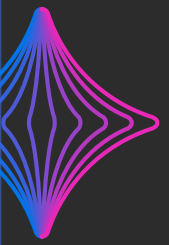


Видеореклама

Стартовая страница приложения



Товары

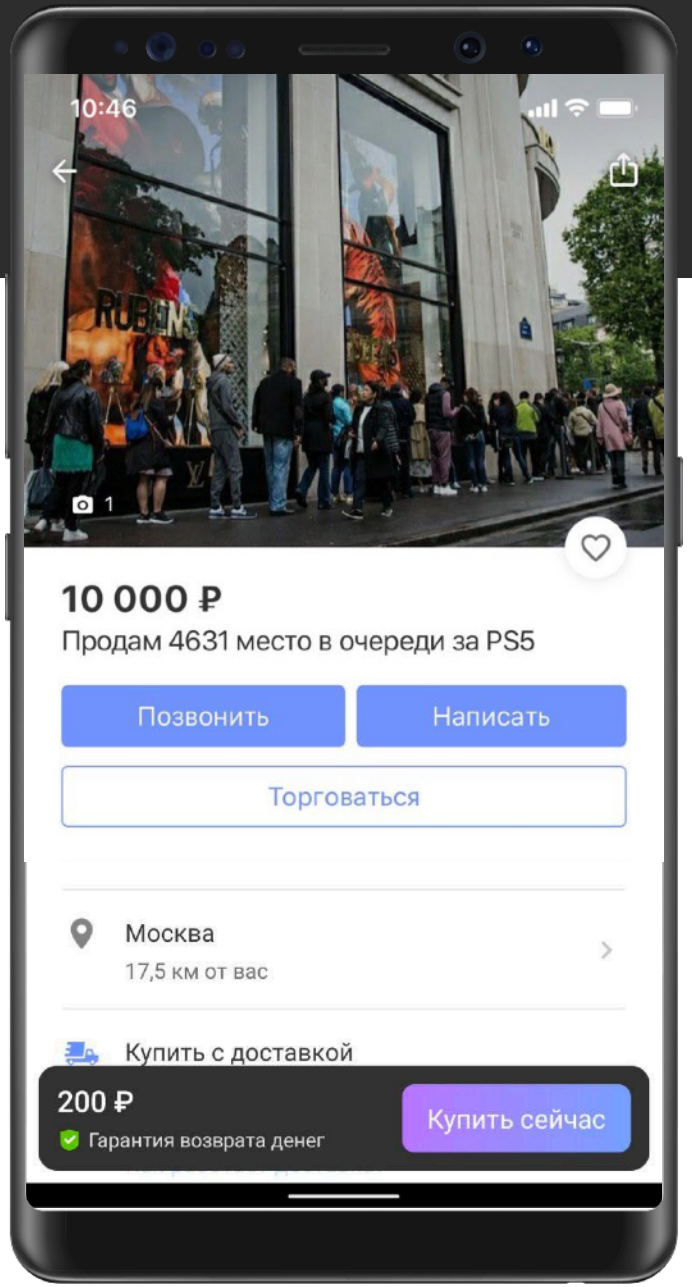


Что по запросам?

- api/v1/abtests
- api/v1/user
- api/v1/product_list
- api/v1/bundles
- api/v1/carousels
- api/v1/ stories
- api/v1/category_config
- api/v1/infoblock
- api/v1/brand_splash
- 2-3 служебных запроса для других частей приложения

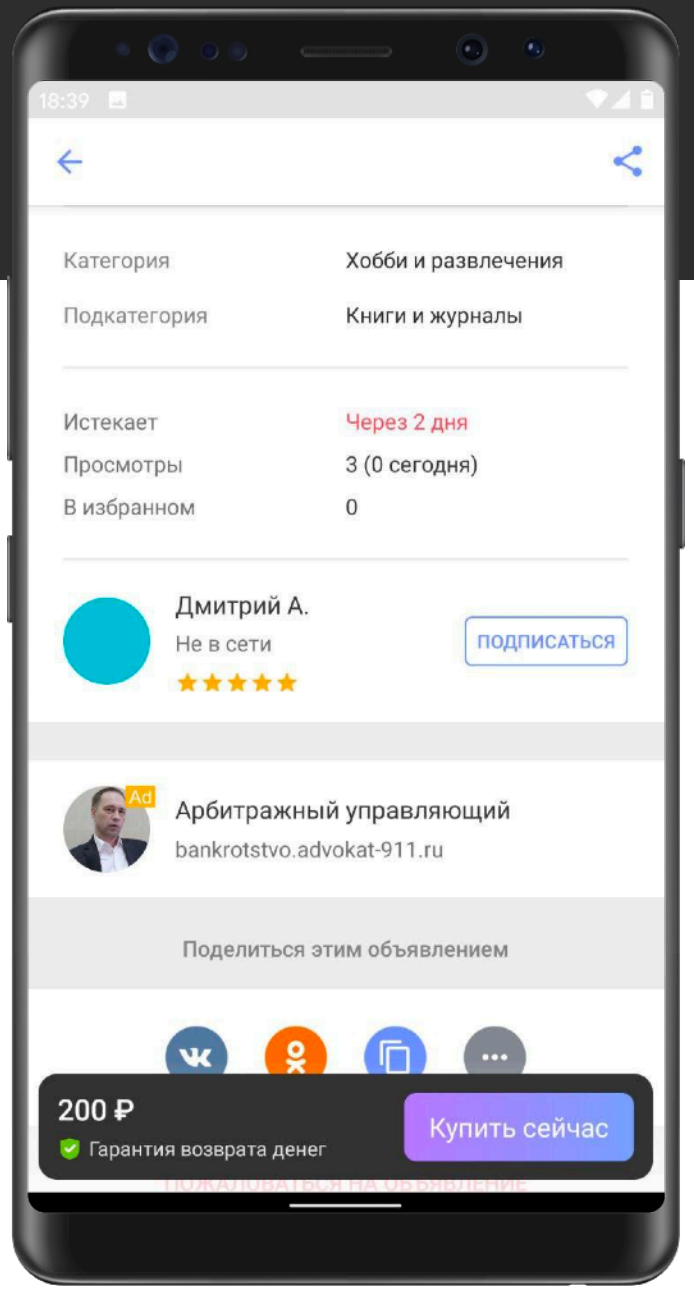
Страница товара


```
"id": "5c500d5485e9d23237014b8f",
"type": "product",
"name": "PHP и MySQL Разработка веб-приложения",
"slug": "php-i-mysql-razrabotka-viebprilozhieniia",
"description": "Отличная книга для начинающих и не только, желающих научиться самому популярному языку программирования PHP и базой дан
"price": 85000,
"price_description": "-30% при покупке онлайн",
"discount": 30,
"discounted_price": 59500,
"date_published": 1571919056,
"is_promoted": false,
"images": [
  {
      "id": "5c500d51cf689a729f038d63",
      "num": 1,
      "url": "https://cache3.youla.io/files/images/orig/5c/50/5c500d51cf689a729f038d63.jpg",
      "width": 3024,
      "height": 4032
  },
  { … }, // 5 items
  { … } // 5 items
],
"location": {
  "latitude": 55.7,
  "longitude": 37.5,
  "description": "Moskva",
  "city": "576d0612d53f3d80945f8b5d"
},
"category": 10,
"subcategory": 1005,
"is_favorite": false,
"views": 14,
"favorite_counter": 1,
"group_id": 1,
"url": "/moskva/hobbi-razvlecheniya/knigi-zhurnaly/php-i-mysql-razrabotka-viebprilozhieniia-5c500d5485e9d23237014b8f",
"share_url": "https://trk.mail.ru/c/phjpd1?id=5c500d5485e9d23237014b8f",
"share_text": "PHP и MySQL Разработка веб-приложения",
```

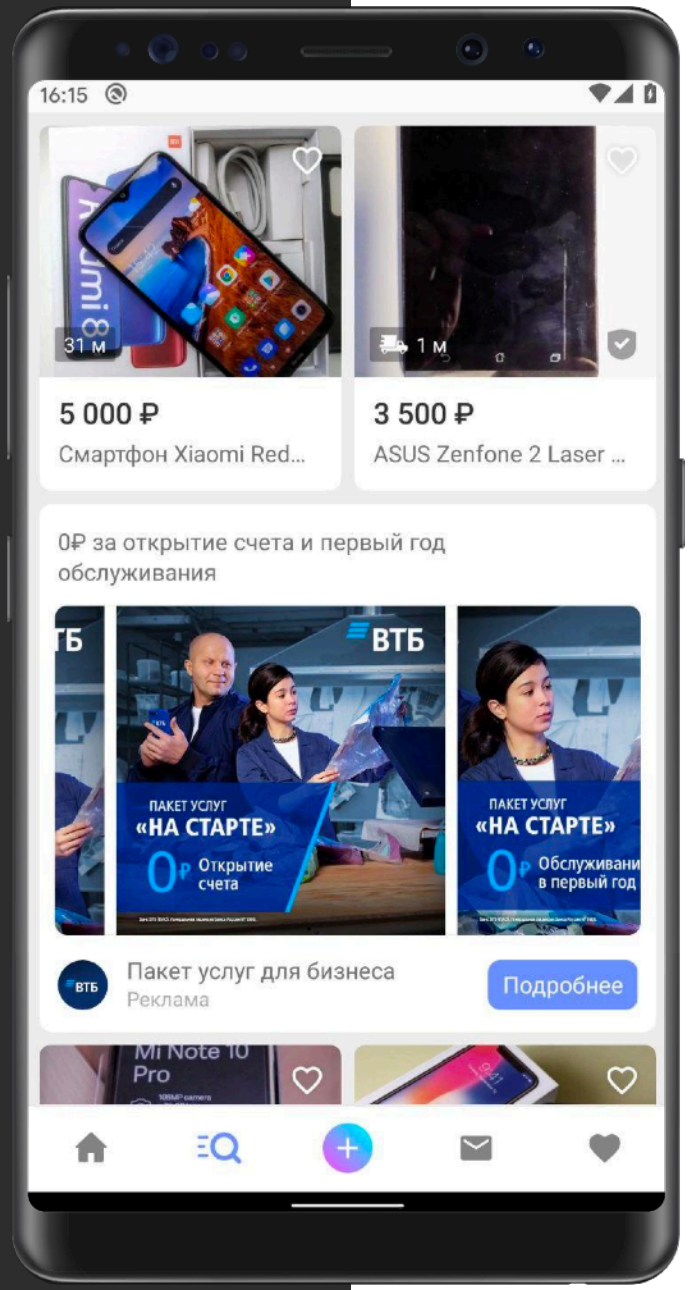



Страница товара

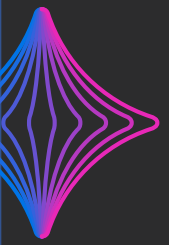

```
"delivery_visible": true,
"delivery_enabled": true,
"owner": {
    "id": "59a28c7ecf689a3615156903",
    "name": "Михаил Ч.",
    "type": "person",
    "is_verified": false,
    "image": {
        "id": "59a28c9fc6ab9e903e17fcf4",
        "num": 1,
        "url": "https://cache3.youla.io/files/images/orig/59/a2/59a28c9fc6ab9e903e17fcf4.jpg"
    },
    "date_registered": 1503825022,
    "settings": {
        "display_phone": true,
        "display_chat": true,
        "location": {
            "description": "Москва"
        }
    },
    "display_phone_num": "7926******1",
    "isOnline": false,
    "onlineText": "Не в сети",
    "is_blocked": false,
    "rating_mark": 5,
    "rating_mark_cnt": 32,
    "prods_active_cnt": 4,
    "prods_sold_cnt": 35,
    "orders_cnt": 2,
    "orders_seller_cnt": 2,
    "orders_buyer_cnt": 0,
},
```

Стартовая страница приложения

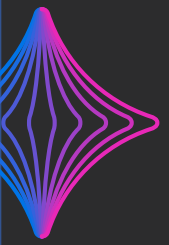


Карточка товара



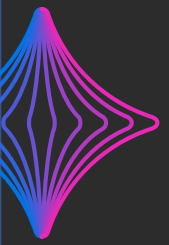
Проблемы

1. Объём передаваемых данных
2. Сложная последовательность rest-запросов
3. Версионирование
4. Девиация времени запуска
    а. Задержки «обязательных» сетевых ручек
    б. «Мигание» ленты при «необязательных» сетевых ручках



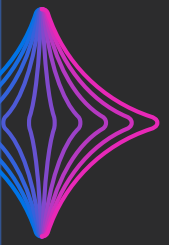
Проблемы

1. Объём передаваемых данных
2. Сложная последовательность rest-запросов
3. Версионирование
4. Девиация времени запуска
    а. Задержки «обязательных» сетевых ручек
    б. «Мигание» ленты при «необязательных» сетевых ручках



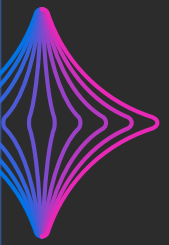
Проблемы

1. Объём передаваемых данных
2. Сложная последовательность rest-запросов
3. Версионирование
4. Девиация времени запуска
    a. Задержки «обязательных» сетевых ручек
    б. «Мигание» ленты при «необязательных» сетевых ручках



Проблемы

1. Объём передаваемых данных
2. Сложная последовательность rest-запросов
3. Версионирование
4. Девиация времени запуска
    a. Задержки «обязательных» сетевых ручек
    б. «Мигание» ленты при «необязательных» сетевых ручках

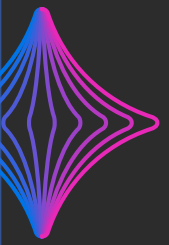


Проблемы

1. Объём передаваемых данных
2. Сложная последовательность rest-запросов
3. Версионирование
4. Девиация времени запуска
    a. Задержки «обязательных» сетевых ручек
    б. «Мигание» ленты при «необязательных» сетевых ручках

Результат!

YOULABS



Результат

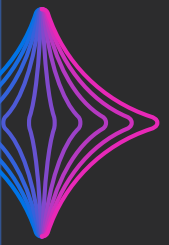
-5%

текстовый поиск

-20%

старт приложения

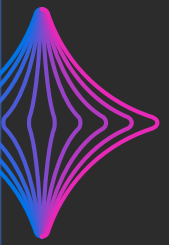
YOULABS



Результат

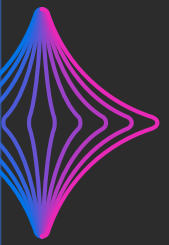
- Консистентность – предсказуемые результаты метрики от изменений на клиенте

YOULABS



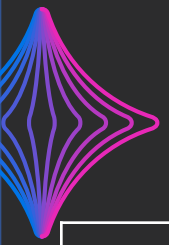
Результат

- Консистентность – предсказуемые результаты метрики от изменений на клиенте
- Доработан мониторинг на бэкенде, т.к. Gateway GQL – клиент api



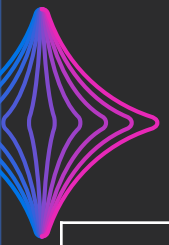
Результат

- Консистентность – предсказуемые результаты метрики от изменений на клиенте
- Доработан мониторинг на бэкенде, т.к. Gateway GQL – клиент api
- Микросервисы на Go, формат ответа для клиента не меняется



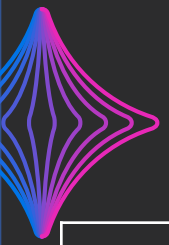
Как можно было бы решать задачу

Решение	Произвольный json. Потоковые парсеры	gRPC + protobuf	Falcor	oData	GraphQL
Плюсы	* Гибкость * Быстродействие	* Гибкость * Малый объём данных	JSON + только необходимые данные	Гибкая выборка данных	* Объём данных * Внедрение * N-ресурсов в 1 запросе * Success у коллег
Минусы	Время внедрения	Время внедрения	Нет библиотек для IOS/Android	Ограничения Rest	* Сложнее в освоении



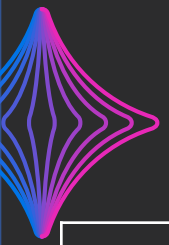
Как можно было бы решать задачу

Решение	Произвольный json. Потоковые парсеры	gRPC + protobuf	Falcor	oData	GraphQL
Плюсы	* Гибкость * Быстродействие	* Гибкость * Малый объём данных	JSON + только необходимые данные	Гибкая выборка данных	* Объём данных * Внедрение * N-ресурсов в 1 запросе * Success у коллег
Минусы	Время внедрения	Время внедрения	Нет библиотек для IOS/Android	Ограничения Rest	* Сложнее в освоении



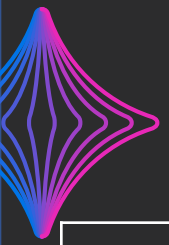
Как можно было бы решать задачу

Решение	Произвольный json. Потоковые парсеры	gRPC + protobuf	Falcor	oData	GraphQL
Плюсы	* Гибкость * Быстродействие	* Гибкость * Малый объём данных	JSON + только необходимые данные	Гибкая выборка данных	* Объём данных * Внедрение * N-ресурсов в 1 запросе * Success у коллег
Минусы	Время внедрения	Время внедрения	Нет библиотек для IOS/Android	Ограничения Rest	* Сложнее в освоении



Как можно было бы решать задачу

Решение	Произвольный json. Потоковые парсеры	gRPC + protobuf	Falcor	oData	GraphQL
Плюсы	* Гибкость * Быстродействие	* Гибкость * Малый объём данных	JSON + только необходимые данные	Гибкая выборка данных	* Объём данных * Внедрение * N-ресурсов в 1 запросе * Success у коллег
Минусы	Время внедрения	Время внедрения	Нет библиотек для IOS/Android	Ограничения Rest	* Сложнее в освоении



Как можно было бы решать задачу

Решение	Произвольный json. Потоковые парсеры	gRPC + protobuf	Falcor	oData	GraphQL
Плюсы	* Гибкость * Быстродействие	* Гибкость * Малый объём данных	JSON + только необходимые данные	Гибкая выборка данных	* Объём данных * Внедрение * N-ресурсов в 1 запросе * Success у коллег
Минусы	Время внедрения	Время внедрения	Нет библиотек для IOS/Android	Ограничения Rest	* Сложнее в освоении

GraphQL -> Graph Query Language

Facebook since 2012 (Relay)

YOULABS

GraphQL -> Graph Query Language Facebook since 2012 (Relay)

- Ask for what you need, get exactly that

GraphQL -> Graph Query Language Facebook since 2012 (Relay)

- Ask for what you need, get exactly that
- Get many resources in a single request

GraphQL -> Graph Query Language Facebook since 2012 (Relay)

- Ask for what you need, get exactly that
- Get many resources in a single request
- Describe what’s possible with a type system

GraphQL -> Graph Query Language Facebook since 2012 (Relay)

- Ask for what you need, get exactly that
- Get many resources in a single request
- Describe what’s possible with a type system
- Evolve your API without versions

GraphQL

Query

Запрос данных,
эквивалент GET
в REST

YOULABS

GraphQL

Query

Запрос данных,
эквивалент GET
в REST

Mutation

Создание,
обновление и
удаление данных,
эквивалент PUT,
POST, DELETE
в REST

GraphQL

Query

Запрос данных, эквивалент GET в REST

Mutation

Создание, обновление и удаление данных, эквивалент PUT, POST, DELETE в REST

Subscription

Подписки на события, отправляемые с сервера клиенту, нет эквивалента в REST

Graph QL – what is

Type - тип с полями

```
type Image {
        id: String!
        url: String!
    }
```

Graph QL – what is

Schema - контракт, набор типов

```
schema {
  query: Query
}
```

Graph QL –what is

Fragment - set полей для определённого типа

```
fragment FeedProduct
on Product {
  name: String!
  price: Price!
}
```

Graph QL – what is

Union - объединение типов

```
union FeedItem =
AdvertItem |
BundleItem |
CarouselItem |
ProductItem |
StoryItem | …
```

Graph QL – what is

Interface - можно вынести общие поля

```
interface Node {
  id: ID!
}

type Product
implements Node {
  id: ID!
  ...
}
```

GraphQL – what is

- Инструменты на этапе проектирования
- «Интроспекция»

YOULABS

Пример SWApi

https://graphql.org/swapi-graphql

Persons

Пример SWApi

```
query {
  allPeople(first : 2) {
    people {
      name,
      birthYear,
      eyeColor
    }
  }
}
```

```
{
  "data": {
    "allPeople": {
      "people": [{
                  "name": "Luke Skywalker",
                  "birthYear": "19BBY",
                  "eyeColor": "blue"
                },
                {
                  "name": "Darth Vader",
                  "birthYear": "41.9BBY",
                  "eyeColor": "yellow"
                }]
    }
    }
}
```

Было – стало

- api/v1/abtests
- api/v1/user
- api/v1/product_list
- api/v1/bundles
- api/v1/carousels
- api/v1/stories
- api/v1/category_config
- api/v1/infoblock
- api/v1/brand_splash
- 2-3 служебных запроса для других частей приложения

```
type Query {
   feed(input: Filter!): Feed!
      ...
 }
```

Было – стало

- api/v1/abtests
- api/v1/user
- api/v1/product_list
- api/v1/bundles
- api/v1/carousels
- api/v1/stories
- api/v1/category_config
- api/v1/infoblock
- api/v1/brand_splash
- 2-3 служебных запроса для других частей приложения

```
type Query {
   feed(input: Filter!): Feed!
      ...
 }

type Feed {
    items: [FeedItem!]
    pageInfo: PageInfo
}
```

Было – стало

- api/v1/abtests
- api/v1/user
- api/v1/product_list
- api/v1/bundles
- api/v1/carousels
- api/v1/stories
- api/v1/category_config
- api/v1/infoblock
- api/v1/brand_splash
- 2-3 служебных запроса для других частей приложения

```
type Query {
   feed(input: Filter!): Feed!
      ...
 }

type Feed {
    items: [FeedItem!]
    pageInfo: PageInfo
}

union FeedItem = AdvertItem |
BundleItem | CarouselItem |
ProductItem | StoryItem | …
```

Было – стало


```
"id": "5e4a98032217296194324e61",
 "owner": {
     "id": "5bfeb1350eae8d3337269ed2"
 },
 "linked_id": null,
 "type": "product",
 "name": "Лабутены",
 "slug": "labutieny",
 "description": "Нах!",
 "price": 10050,
 "discount": 0,
 "discounted_price": 10050,
 "date_created": 1581946883,
 "date_updated": 1581946883,
 "date_published": 1581946883,
 "date_sold": null,
 "date_blocked": null,
 "date_deleted": null,
 "date_archivation": 1584538883,
 "is_published": true,
 "is_sold": false,
 "sold_mode": 0,
 "is_deleted": false,
 "is_blocked": false,
 "is_archived": false,
 "archive_mode": 0,
 "is_expiring": false,
 "is_verified": false,
 "block_mode": 0,
 "block_type": 0,
 "block_types": [],
 "block_type_text": "",
 "images": [
     {
        "id": "5e4a97f88bfa8b07e803a33a",
         "num": 1,
         "url": "http://img.youla.portal-omproxy.devmail.ru/files/images/
orig/5e/4a/5e4a97f88bfa8b07e803a33a.jpg",
         "width": 1920,
         "height": 1200

     }
 ],
 "location": {
     "latitude": 10,
     "longitude": 20,
     "description": "Moscow"
 },
 "category": 9,
 "subcategory": 902,
 "is_favorite": false,
 "date_favorited": null,
 "views": 1074,
 "favorite_counter": 10,
```

Было – стало

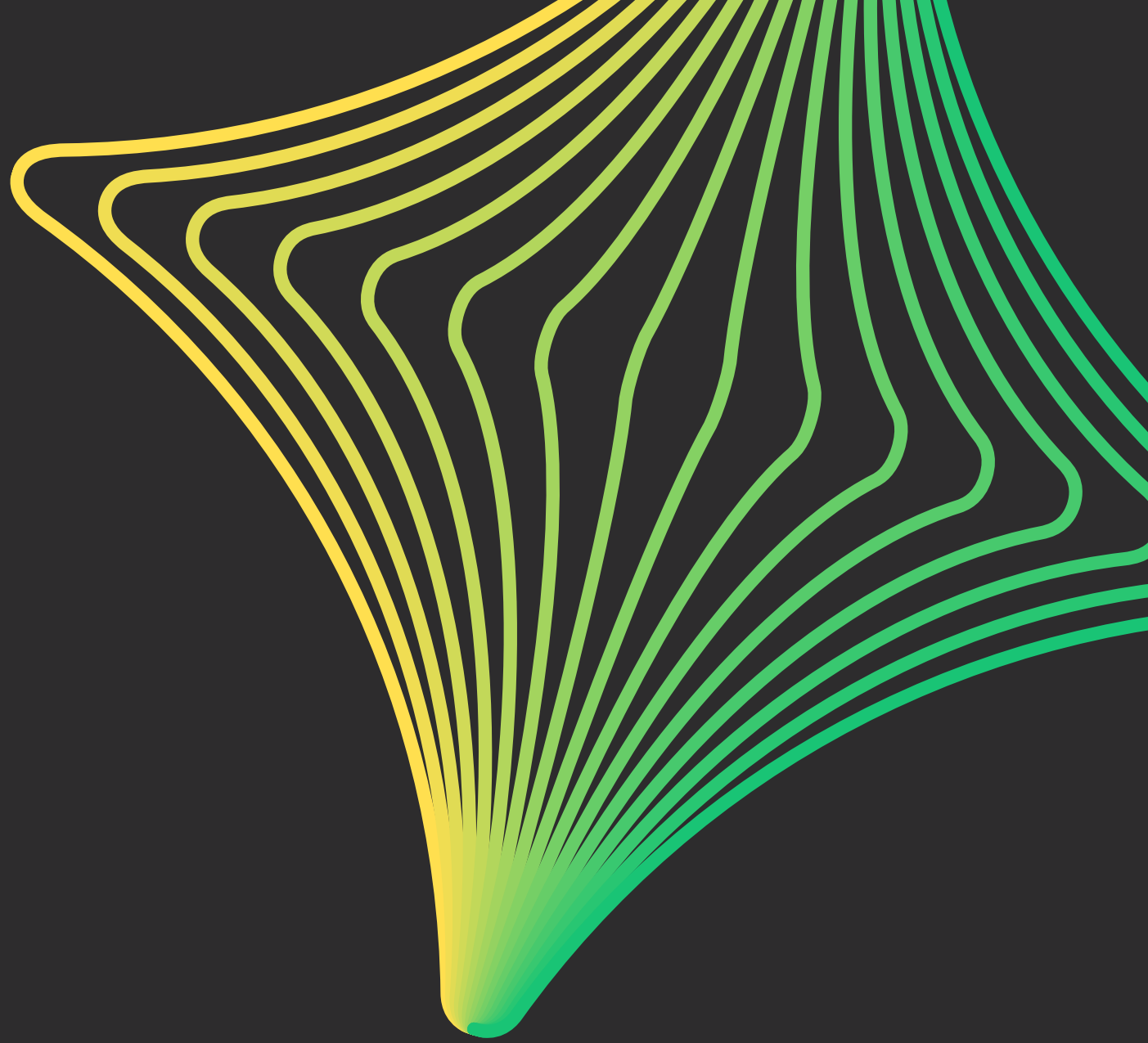

```
"id": "5e4a98032217296194324e61",
"owner": {
    "id": "5bfeb1350eae8d3337269ed2"
},
"linked_id": null,
"type": "product",
"name": "Лабутены",
"slug": "labutieny",
"description": "Нах!",
"price": 10050,
"discount": 0,
"discounted_price": 10050,
"date_created": 1581946883,
"date_updated": 1581946883,
"date_published": 1581946883,
"date_sold": null,
"date_blocked": null,
"date_deleted": null,
"date_archivation": 1584538883,
"is_published": true,
"is_sold": false,
"sold_mode": 0,
"is_deleted": false,
"is_blocked": false,
"is_archived": false,
"archive_mode": 0,
"is_expiring": false,
"is_verified": false,
"block_mode": 0,
"block_type": 0,
"block_types": [],
"block_type_text": "",
"images": [
    {
        "id": "5e4a97f88bfa8b07e803a33a",
        "num": 1,
        "url": "http://img.youla.portal-omproxy.devmail.ru/files/images/
orig/5e/4a/5e4a97f88bfa8b07e803a33a.jpg",
        "width": 1920,
        "height": 1200

    }
],
"location": {
    "latitude": 10,
    "longitude": 20,
    "description": "Moscow"
},
"category": 9,
"subcategory": 902,
"is_favorite": false,
"date_favorited": null,
"views": 1074,
"favorite_counter": 10,
```


```
query($input: Filter!) {
    feed(input: $input) {
        … on ProductItem {
            id,
            images(limit: 1) {
              url
            },
            name,
            isSafePay,
            price
        }
        ...
    }
}
```

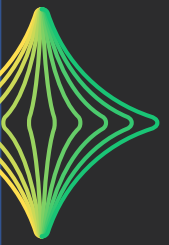
Версионирование

```
type Feed {
    items: [FeedItem!]
    @Deprecated
    pageInfo: PageInfo
}
```

Apollo



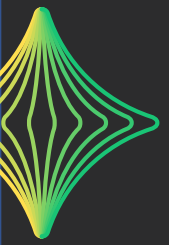
YOULABS



Apollo (Mike Nakhimovich)

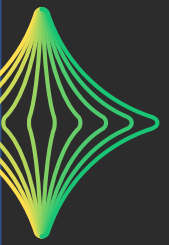
Apollo-Android

Apollo-Android is designed primarily with Android in mind but you can use it in any Java/Kotlin app. The android-only parts are in apollo-android-support and are only needed to use SQLite as a cache or the android main thread for callbacks.



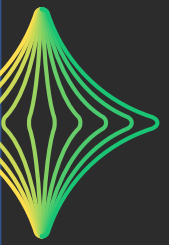
Apollo

- Automatic generation of typesafe models



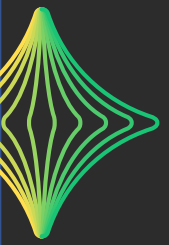
Apollo

- Automatic generation of typesafe models
- Support for Java and Kotlin code generation



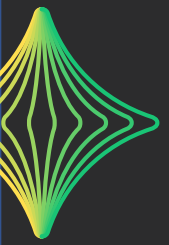
Apollo

- Automatic generation of typesafe models
- Support for Java and Kotlin code generation
- Queries, Mutations and Subscriptions



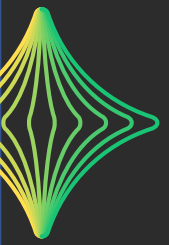
Apollo

- Automatic generation of typesafe models
- Support for Java and Kotlin code generation
- Queries, Mutations and Subscriptions
- Reflection-free parsing of responses



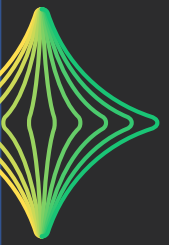
Apollo

- Automatic generation of typesafe models
- Support for Java and Kotlin code generation
- Queries, Mutations and Subscriptions
- Reflection-free parsing of responses
- HTTP cache. Normalized cache



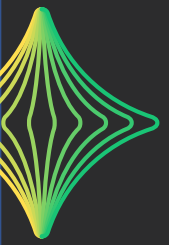
Apollo

- Automatic generation of typesafe models
- Support for Java and Kotlin code generation
- Queries, Mutations and Subscriptions
- Reflection-free parsing of responses
- HTTP cache. Normalized cache
- File Upload



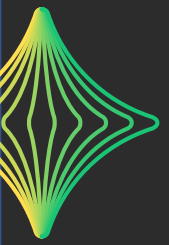
Apollo

- Automatic generation of typesafe models
- Support for Java and Kotlin code generation
- Queries, Mutations and Subscriptions
- Reflection-free parsing of responses
- HTTP cache. Normalized cache
- File Upload
- Custom scalar types



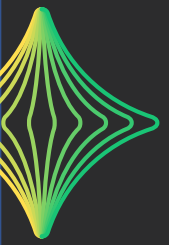
Apollo

- Automatic generation of typesafe models
- Support for Java and Kotlin code generation
- Queries, Mutations and Subscriptions
- Reflection-free parsing of responses
- HTTP cache. Normalized cache
- File Upload
- Custom scalar types
- Support for RxJava2 and Coroutines



Apollo

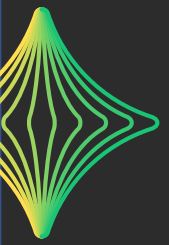
- Automatic generation of typesafe models
- Support for Java and Kotlin code generation
- Queries, Mutations and Subscriptions
- Reflection-free parsing of responses
- HTTP cache. Normalized cache
- File Upload
- Custom scalar types
- Support for RxJava2 and Coroutines
- Persisted Queries



Apollo

- Espresso Idling Resource

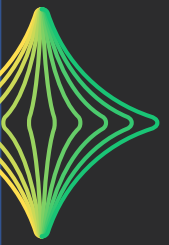
YOULABS



Apollo

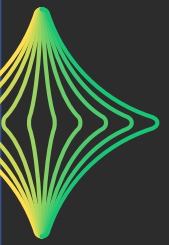
- Espresso Idling Resource
- Open-source, поддержка сообщества

YOULABS



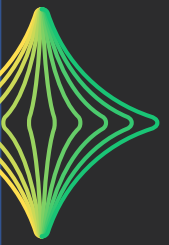
Apollo

- Espresso Idling Resource
- Open-source, поддержка сообщества
- Свой executor (можно заменить)



Apollo

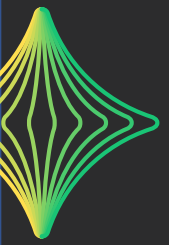
- Espresso Idling Resource
- Open-source, поддержка сообщества
- Свой executor (можно заменить)
- Subscription over websocket



Apollo

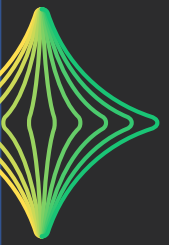
- Espresso Idling Resource
- Open-source, поддержка сообщества
- Свой executor (можно заменить)
- Subscription over websocket
- __typename для кодогенерации

YOULABS



Apollo - Android - Setup

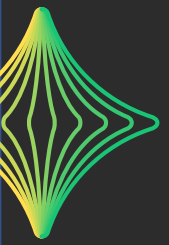
- Gradle: зависимость + плагин



Apollo - Android - Setup

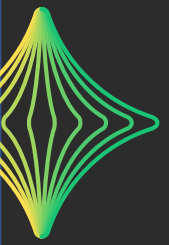
- Gradle: зависимость + плагин
- Файл с query

YOULABS



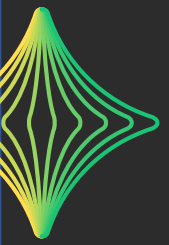
Apollo - Android - Setup

- Gradle: зависимость + плагин
- Файл с query
- Kotlin-code с помощью apollo-cli



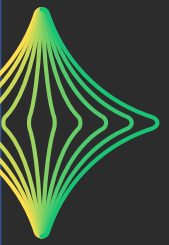
Apollo - Android - Setup

- Gradle: зависимость + плагин
- Файл с query
- Kotlin-code с помощью apollo-cli
- Async-call (Apollo-Call, RxJava Single, Coroutine)



Apollo - Android - Setup

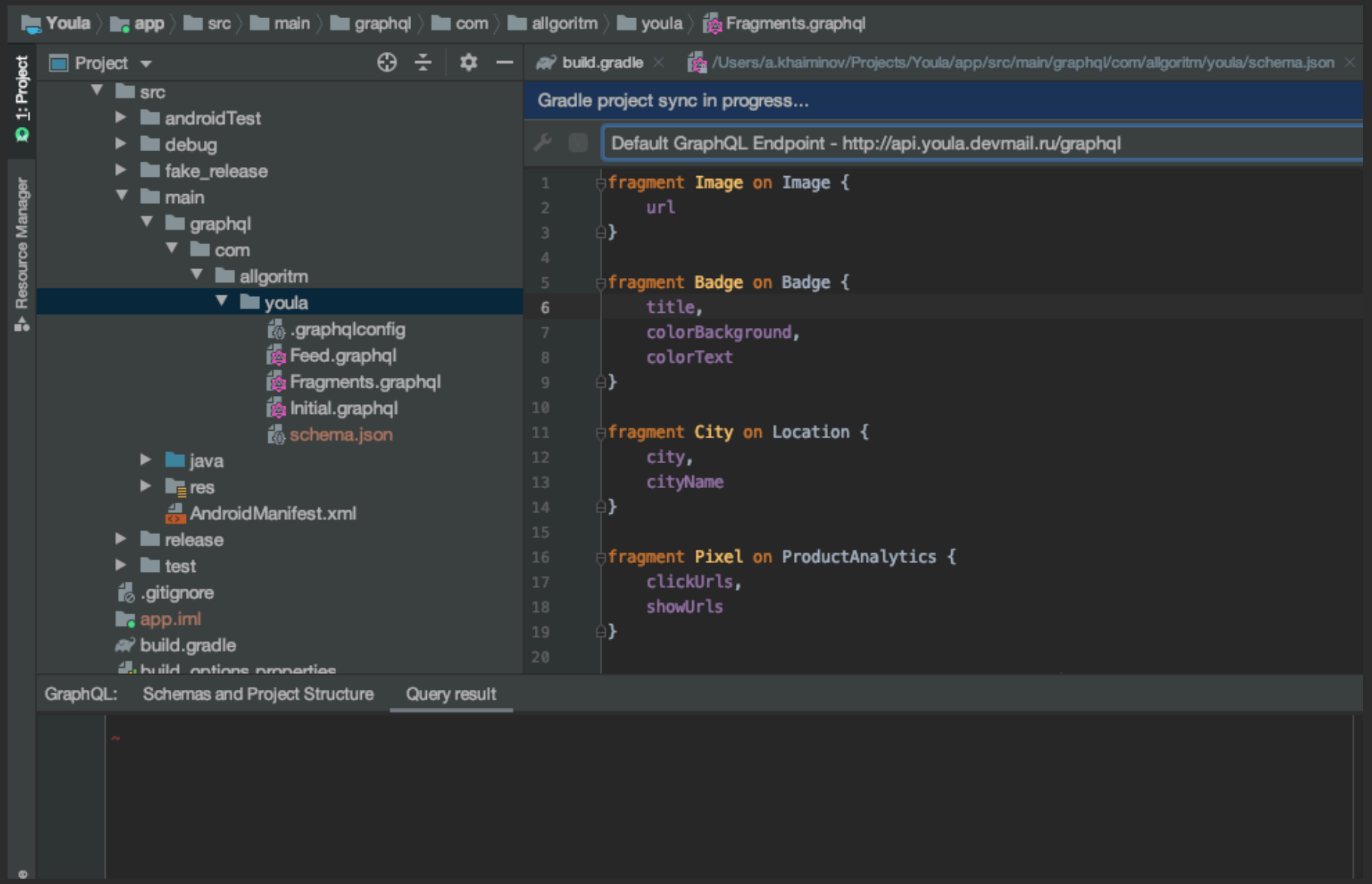
- Gradle: зависимость + плагин
- Файл с query
- Kotlin-code с помощью apollo-cli
- Async-call (Apollo-Call, RxJava Single, Coroutine)
- Profit!



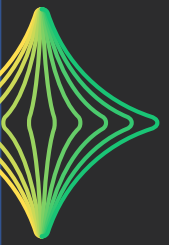
Apollo - Android - Setup

- Gradle: зависимость + плагин
- Файл с query
- Kotlin-code с помощью apollo-cli
- Async-call (Apollo-Call, RxJava Single, Coroutine)
- Profit!
- Бонус: JS-Plugin

Setup Apollo, JS Plugin

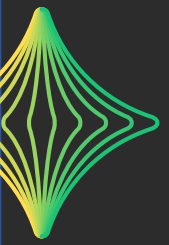


YOULABS



Интроспекция (schema.json)

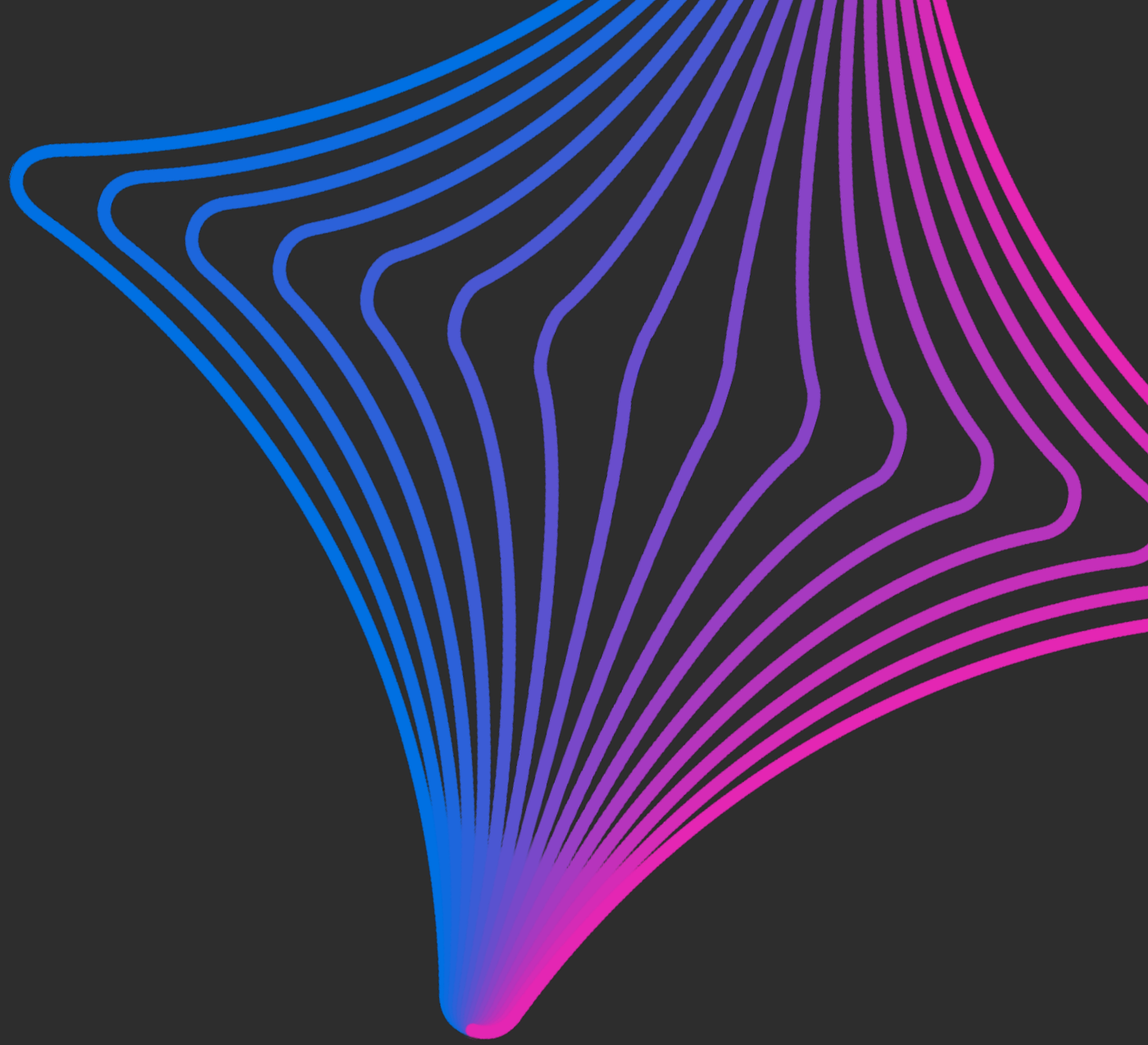

```
{
  "kind": "ENUM",
  "name": "__TypeKind",
  "description": "An enum describing what kind of type a given `__Type` is.",
  "fields": null,
  "inputFields": null,
  "interfaces": null,
  "enumValues": [
    {
      "name": "SCALAR",
      "description": "Indicates this type is a scalar.",
      "isDeprecated": false,
      "deprecationReason": null
    },
    {
      "name": "OBJECT",
      "description": "Indicates this type is an object. `fields` and `interfaces` are valid fields.",
      "isDeprecated": false,
      "deprecationReason": null
    },
    {
      "name": "INTERFACE",
      "description": "Indicates this type is an interface. `fields` and `possibleTypes` are valid fields.",
      "isDeprecated": false,
      "deprecationReason": null
    },
```



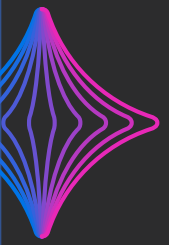
Интроспекция (swapi)

SWApi - typename

Проблемы

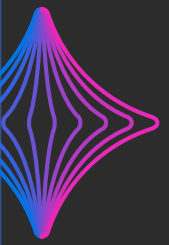


YOULABS



Проблема №1 – генерация классов

```
query {
   items {
      … ProductItem {
         name
        }
        ... CarouselItem {
         product { //ProductItem
            name
            }
        }
    }
}
```

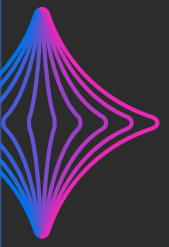


Решение №1 – фрагменты

```
query {
   items {
      … on ProductItem {
         name
        }
        … on CarouselItem {
         product { //ProductItem
            name
            }
        }
        ...
    }
}
```

```
fragment ProductFragment on ProductItem {
    name
}
```


YOULABS



Решение №1 – фрагменты

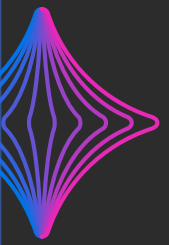
```
query {
   items {
      … on ProductItem {
         ...ProductFragment
        }
        … on CarouselItem {
         product { //ProductItem
            ...ProductFragment
           }
        }
        ...
    }
}
```

```
fragment ProductFragment on ProductItem {
    name
}
```

Вредный совет #2

Работает — не обновляй!

YOULABS

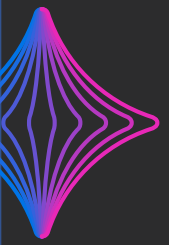


Проблемы

- В Union вернули незапрошенный элемент.

union FeedItem = ProductItem | CarouselItem - было на версии 1.0

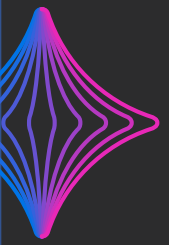
YOULABS



Проблемы

- В Union вернули незапрошенный элемент.

union FeedItem = ProductItem | CarouselItem - было на версии 1.0
union FeedItem = ProductItem | CarouselItem | StoryItem - на версии 1.1

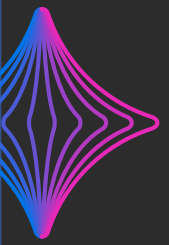


Проблемы

- В Union вернули незапрошенный элемент.

union FeedItem = ProductItem | CarouselItem - было на версии 1.0
union FeedItem = ProductItem | CarouselItem | StoryItem - на версии 1.1

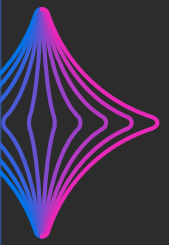
Problem: StoryItem пришёл на версию 1.0



Проблемы

- В Union вернули незапрошенный элемент.
- Union из одного элемента -> генерировал один элемент

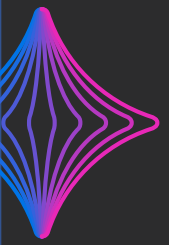
Отсутствовал switch!



Проблемы

- В Union вернули незапрошенный элемент.
- Union из одного элемента -> генерировал один элемент
- Проблемы представления

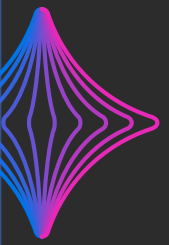
GraphQL как база данных – нет проблем!



Проблемы

- В Union вернули незапрошенный элемент.
- Union из одного элемента -> генерировал один элемент
- Проблемы представления

GraphQL как база данных – нет проблем!
Специфичные UI данные – неудобства для бэкенда.)

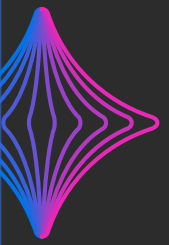


Проблемы

- В Union вернули незапрошенный элемент.
- Union из одного элемента -> генерировал один элемент
- Проблемы представления
- Автоматическое обновление схемы

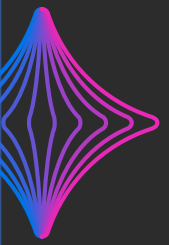
“+” – актуальное консистентное состояние

“-” – тратим время, когда проект не компилируется



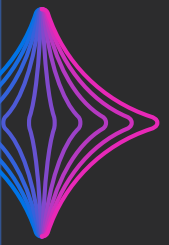
Автоматическая подгрузка схемы (раньше)

```
ext.updateGQLSchema = {
    def base_name = "schema.json"
    def current_dir='pwd'.execute().text.trim()
    def file_location="${current_dir}/your/package/name/${base_name}"
    def schema_file = new File(file_location)
    schema_file.delete()

    def result = "apollo service:download --skipSSLValidation
--endpoint=${host()}".execute().text.trim()

    def src = new File("${current_dir}/${base_name}")

    try {
        schema_file.write(src.text)
    } catch (Exception e) {
        e.printStackTrace()
    }
}
```



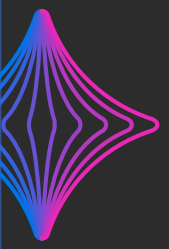
Подгрузка схемы (сейчас)

```
./gradlew downloadApolloSchema \
  -Pcom.apollographql.apollo.endpoint=https://your.graphql.endpoint \
  -Pcom.apollographql.apollo.schema=src/main/graphql/com/example/schema.json
```



Проблемы

- В Union вернули незапрошенный элемент.
- Union из одного элемента -> генерировал один элемент
- Проблемы представления
- Автоматическое обновление схемы
- «Эволюционное» обновление схемы

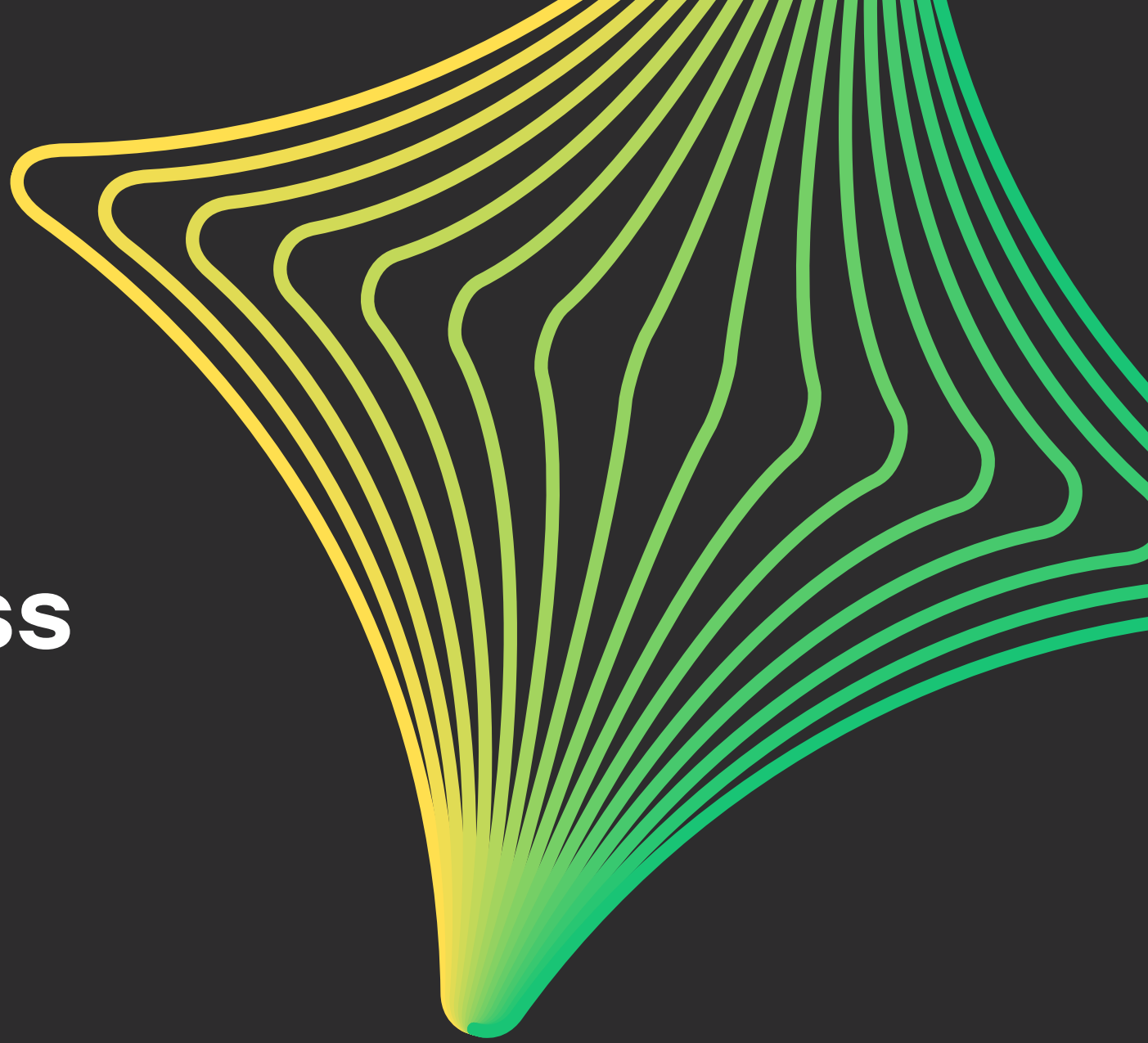


Проблемы

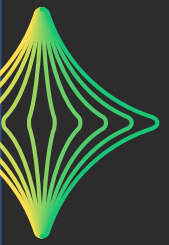
- Рекурсивные структуры данных

```
fragment FieldsRecursive on AllFields {
    fields {
        ...Field
        ... on NestedField {
            subfields {
                ...Field
                ... on NestedField {
                    subfields {
                        ...Field
                    }
                }
            }
        }
        ... on FieldGroup {
            subfields {
                ...Field
            }
        }
    }
}
```

Decision process

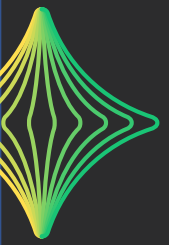


YOULABS



Проблемы

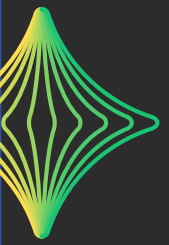
1. Объём передаваемых данных.
2. Сложная последовательность rest-запросов
3. Версионирование
4. Девиация времени запуска
    a. Задержки «обязательных» сетевых ручек
    б. «Мигание» ленты при «необязательных» сетевых ручках



Принятие решения

Влияют ли сетевые задержки?

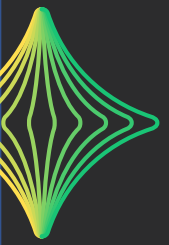
YOULABS



Принятие решения

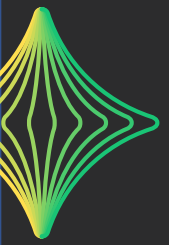
Влияют ли сетевые задержки?

Да, влияют.
Как оценили: interceptor +
ожидание сетевых задержек



Debug OkHttpInterceptor

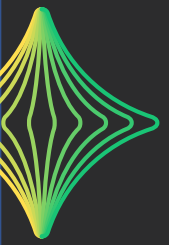
```
private val pathToFileName = mapOf("api/v1/products" to "products.json")

override fun intercept(chain: Interceptor.Chain): Response {
    val path = request.url().pathSegments().joinToString(separator = "/")
    val filename = pathToFileName[path]
    if (filename == null) return chain.proceed()

    return Response.Builder().body(ResponseBody.create(
      contentType, context.getStringFromAssets(filename)))
}
```



Принятие решения

Прототип

- Описали первую схему (типы перенесли)
- Статический шаблон на бэкенде
- Apollo на клиенте
- Парсинг на клиенте



Принятие решения

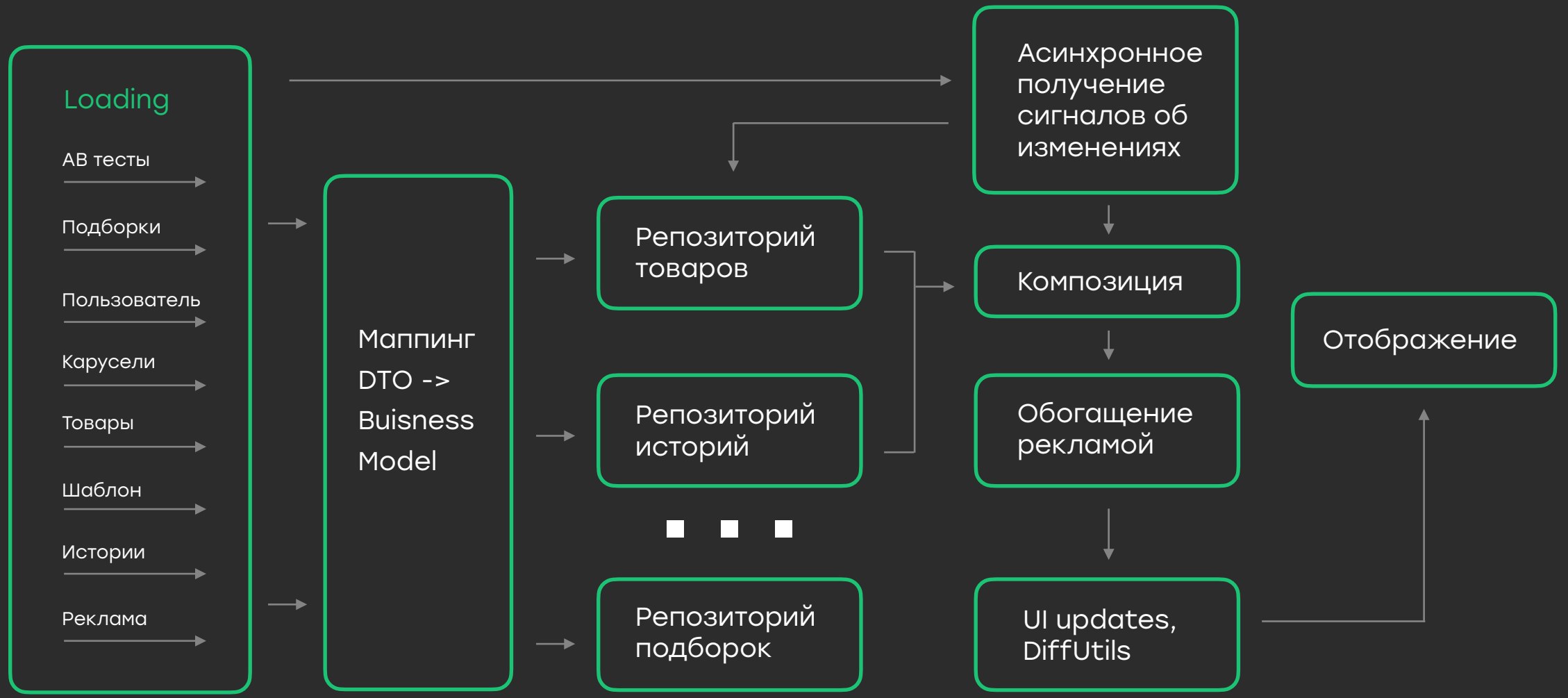
Прототип

- Описали первую схему (типы перенесли)
- Статический шаблон на бэкенде
- Apollo на клиенте
- Парсинг на клиенте

Итого: 4 дня


С самого начала у нас была
какая-то архитектура
WINDSOR MOSCOW
LES JOURS DE
2015
2015

Принятие решения



YOULABS

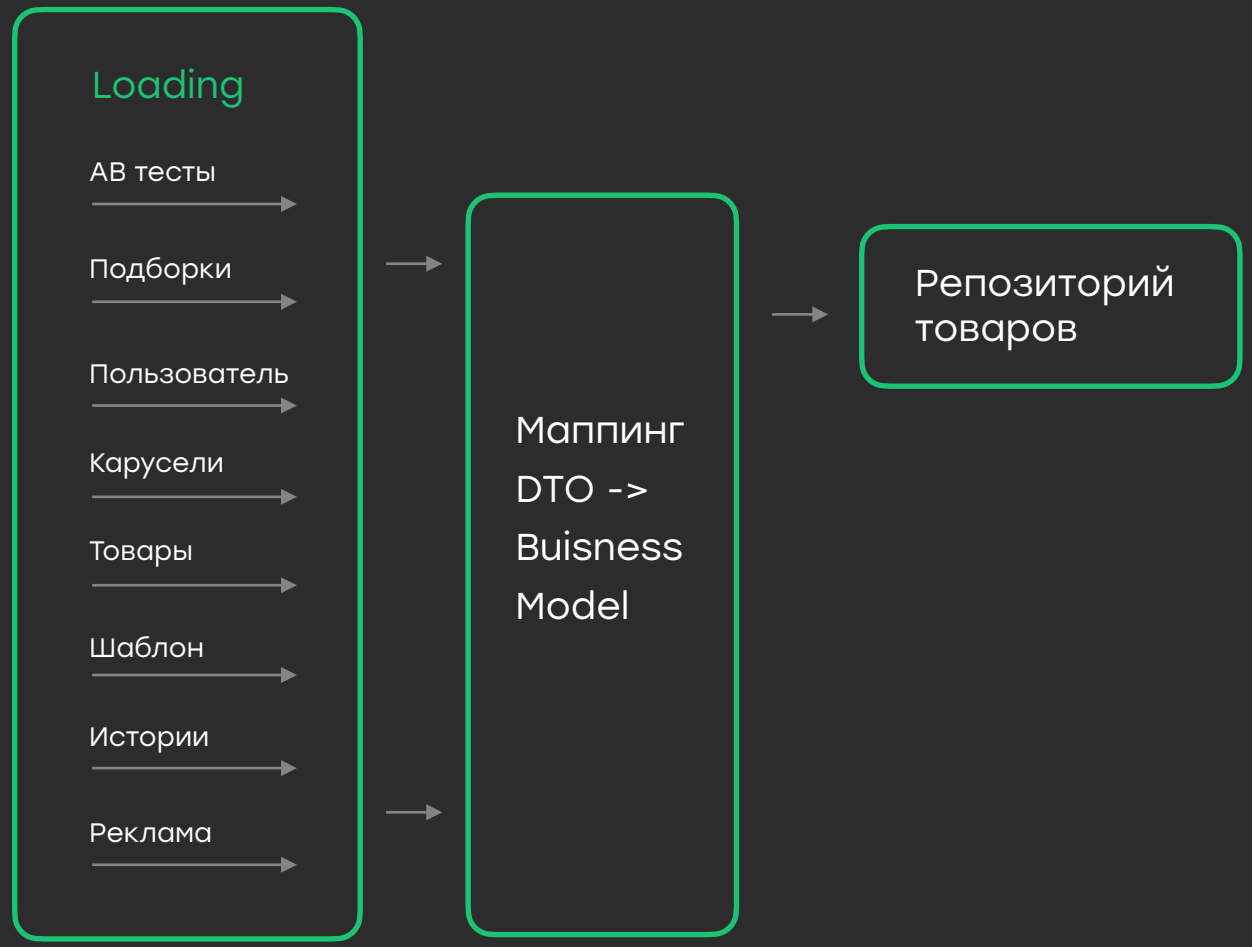
Принятие решения

YOULABS

Принятие решения

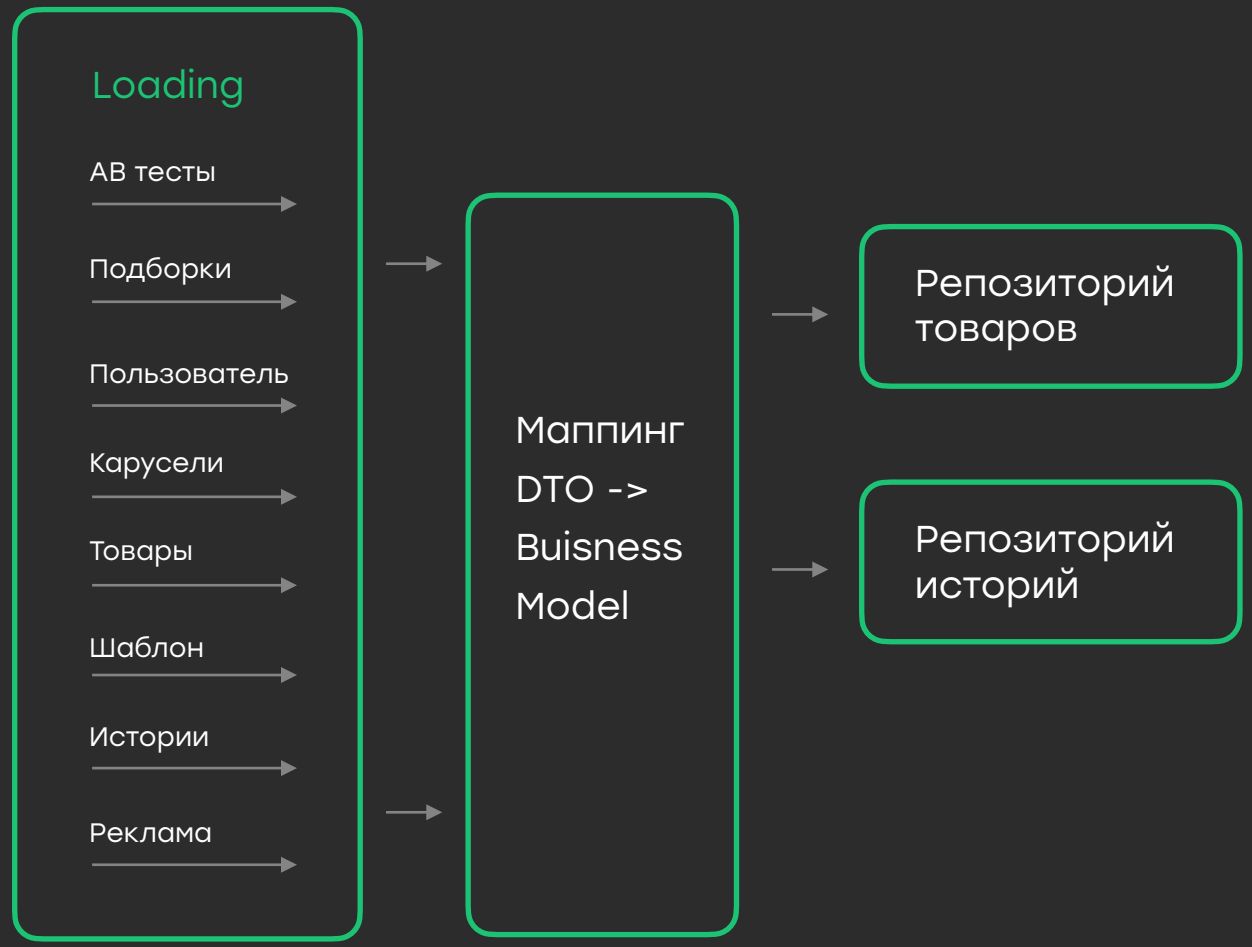
YOULABS

Принятие решения



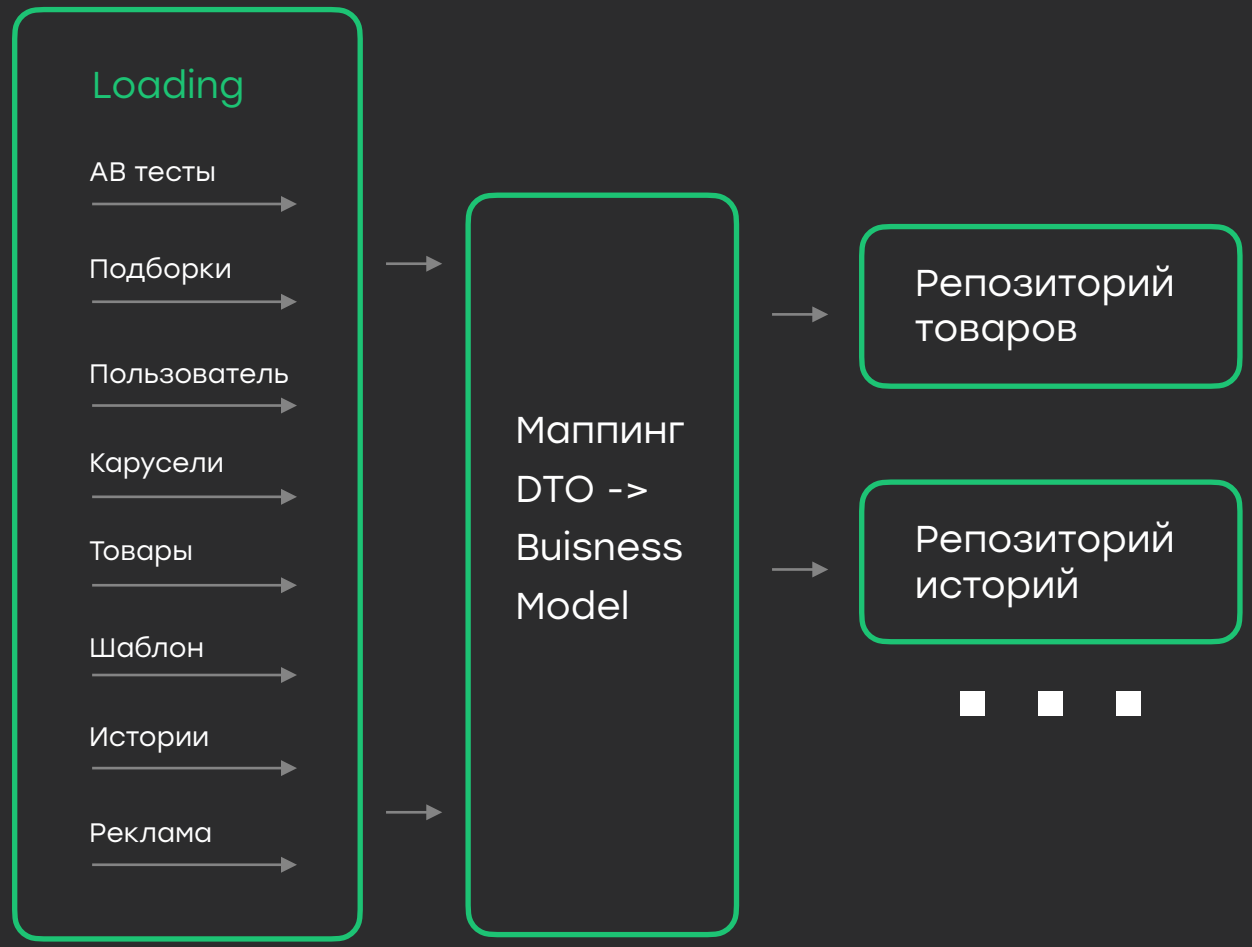
YOULABS

Принятие решения



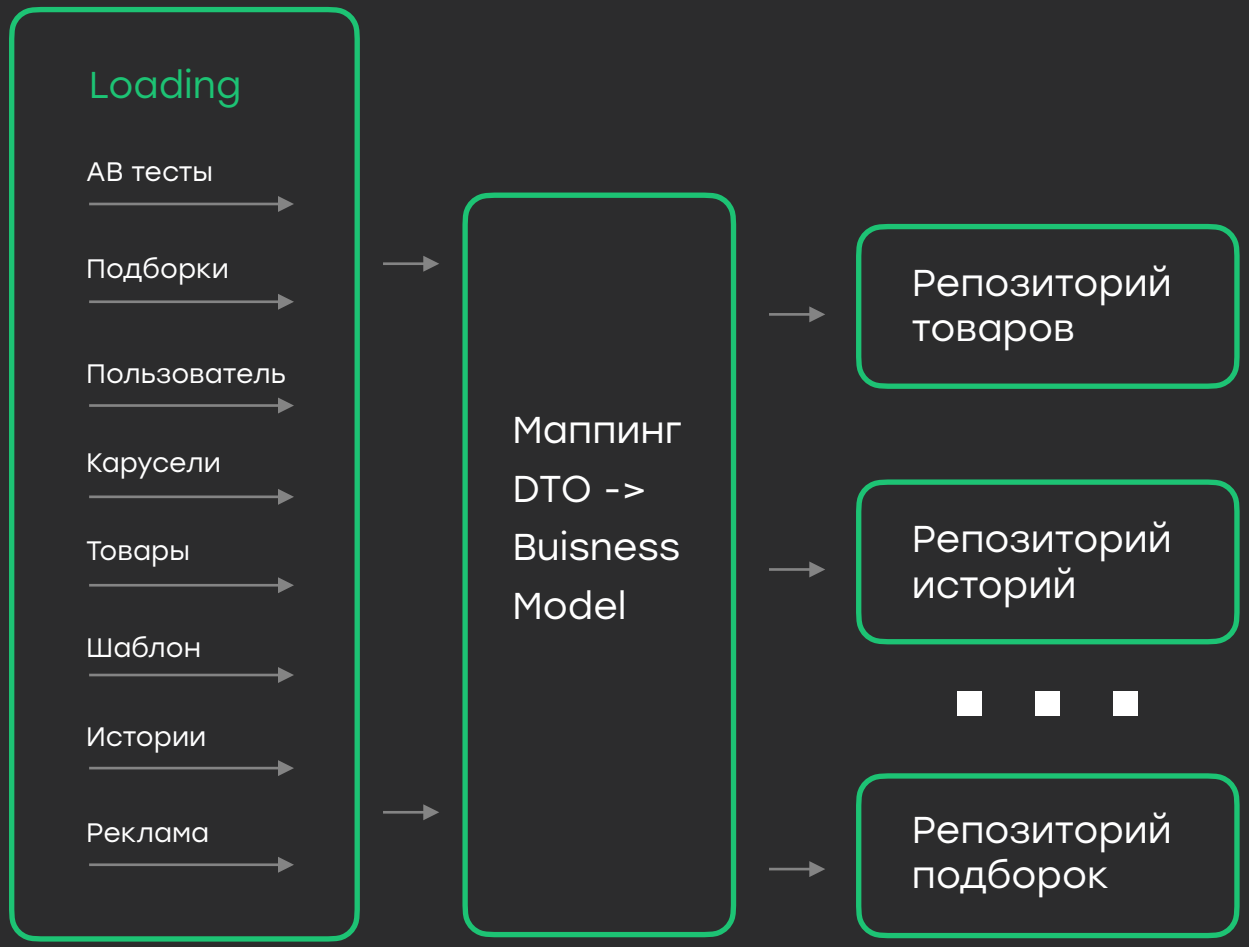
YOULABS

Принятие решения



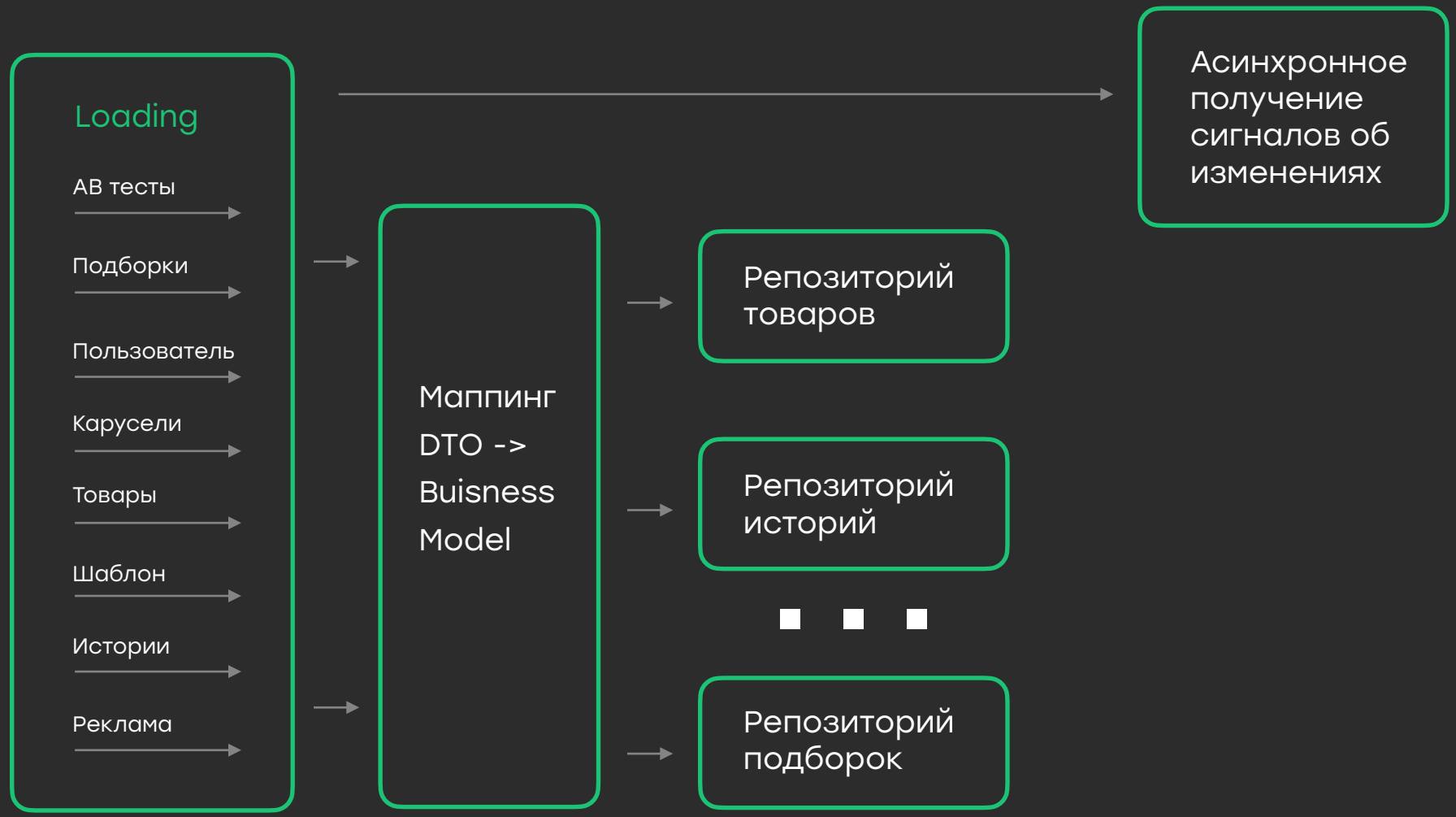
YOULABS

Принятие решения



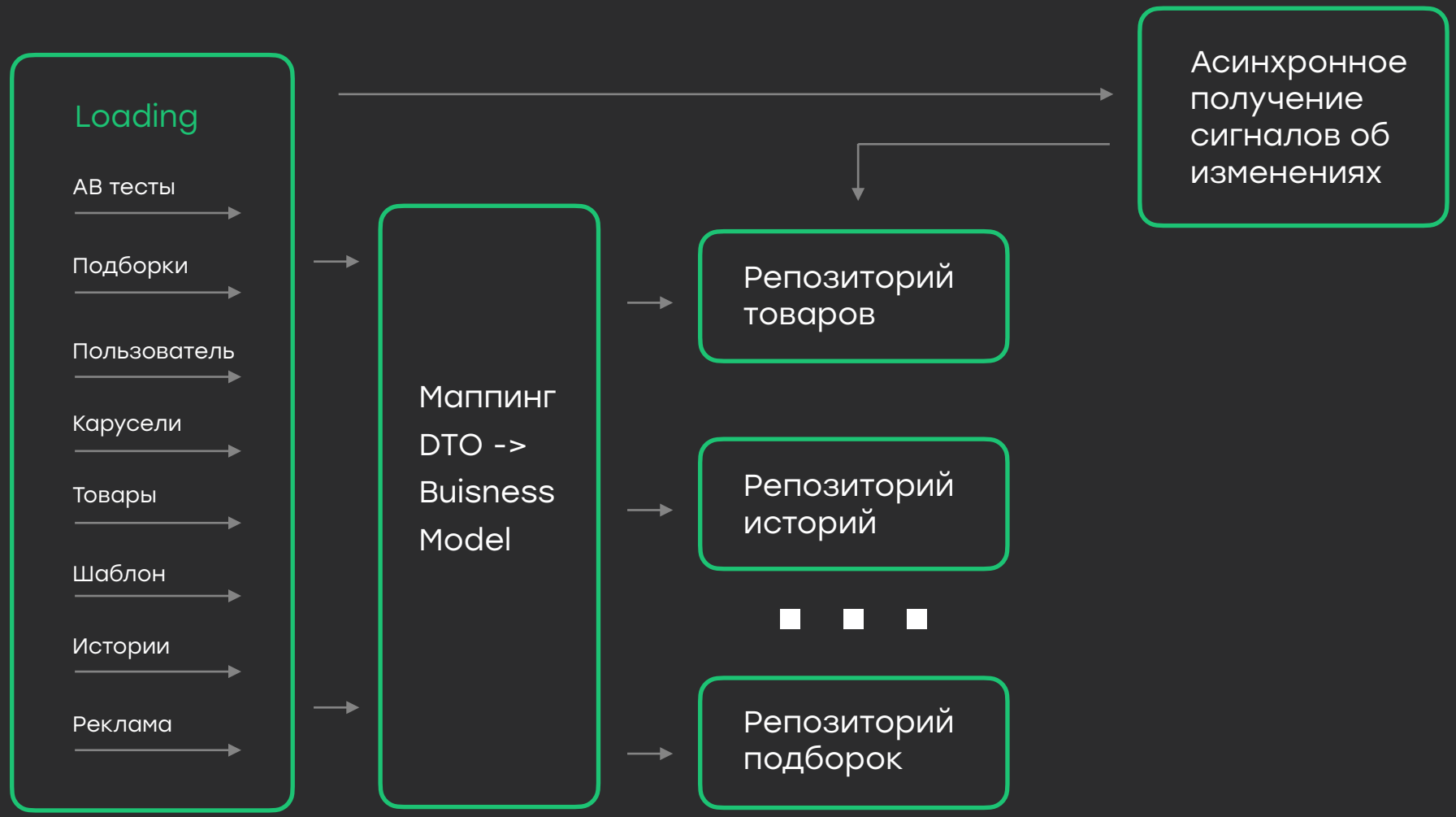
YOULABS

Принятие решения



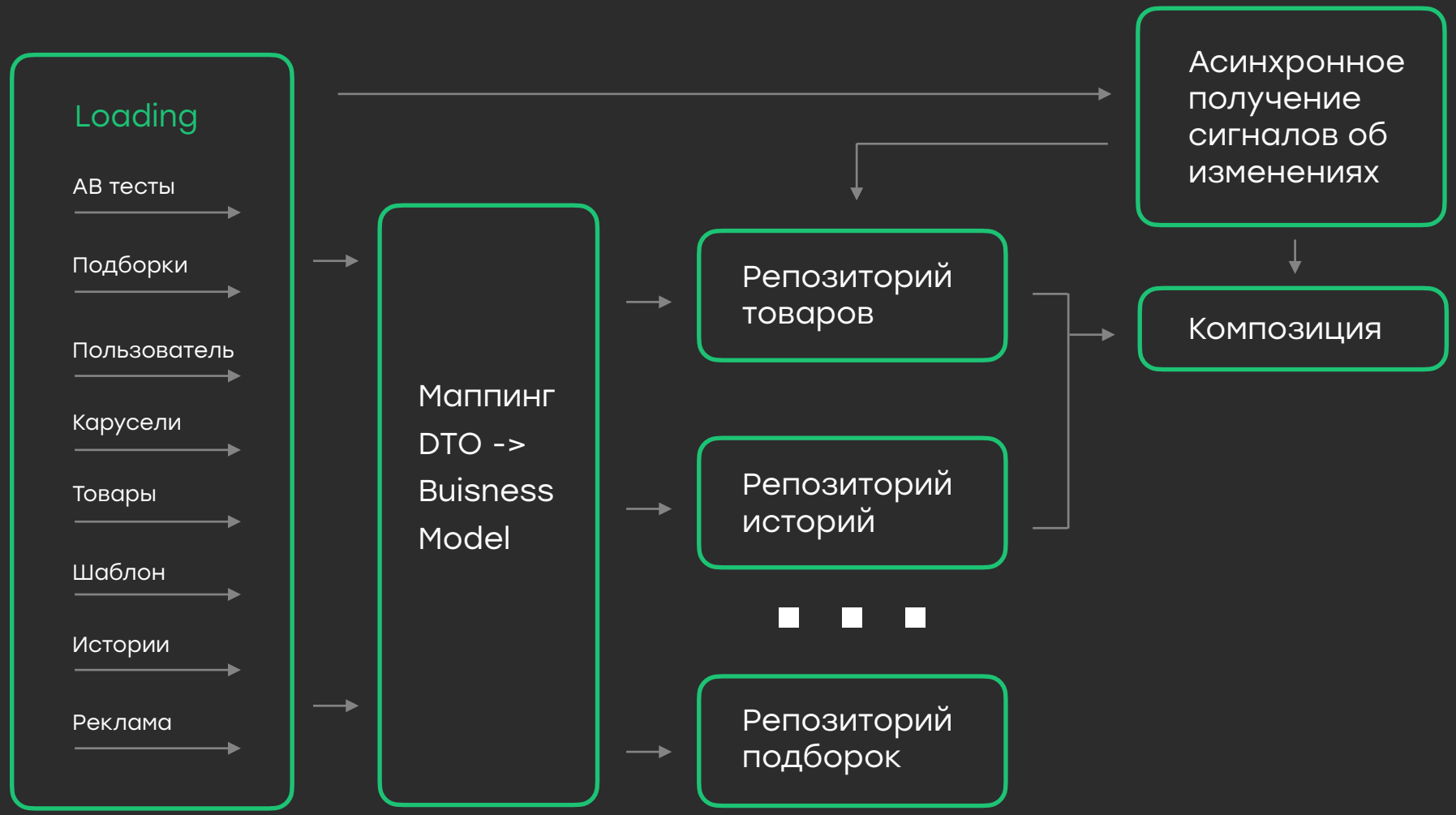
YOULABS

Принятие решения



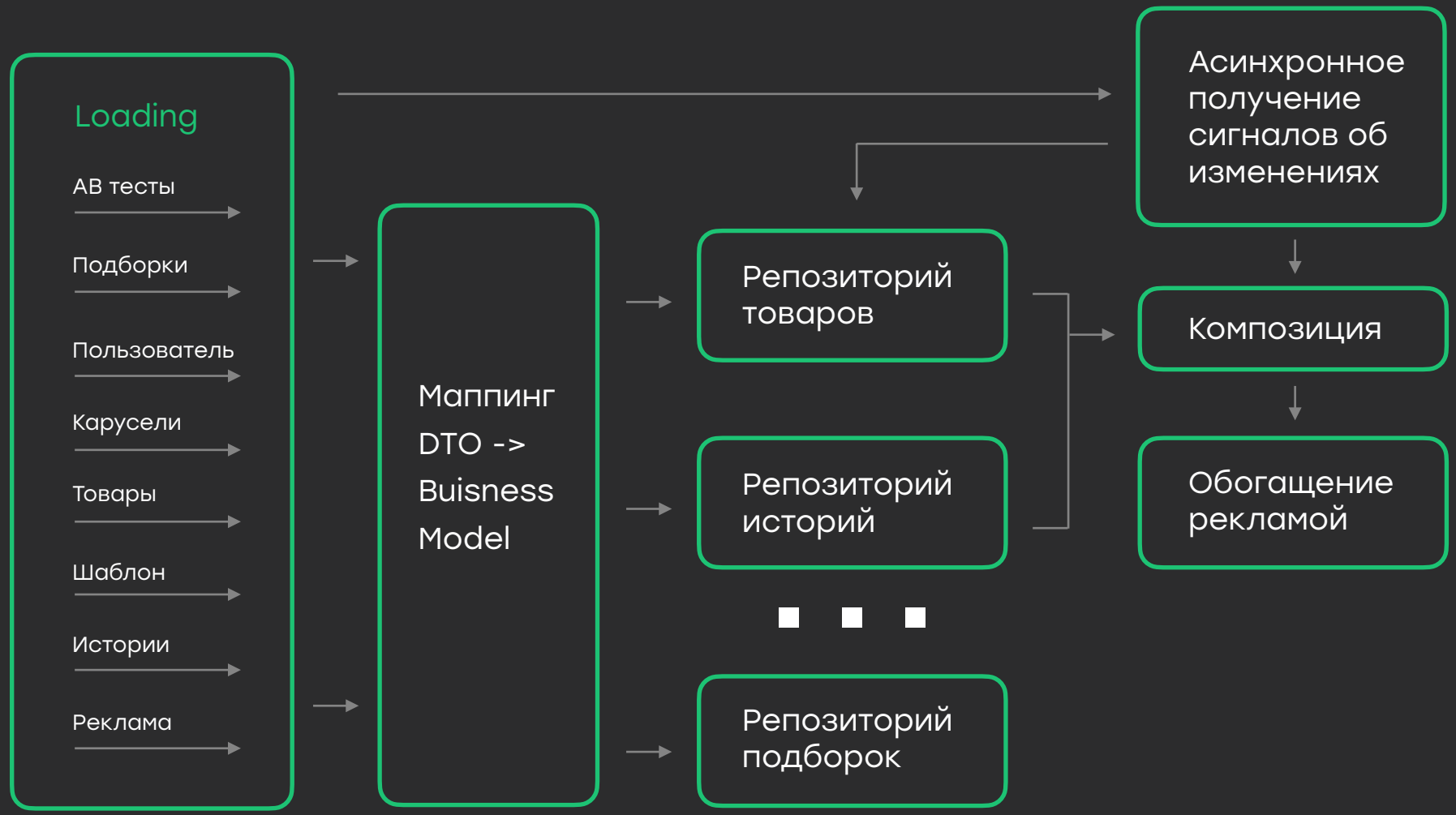
YOULABS

Принятие решения



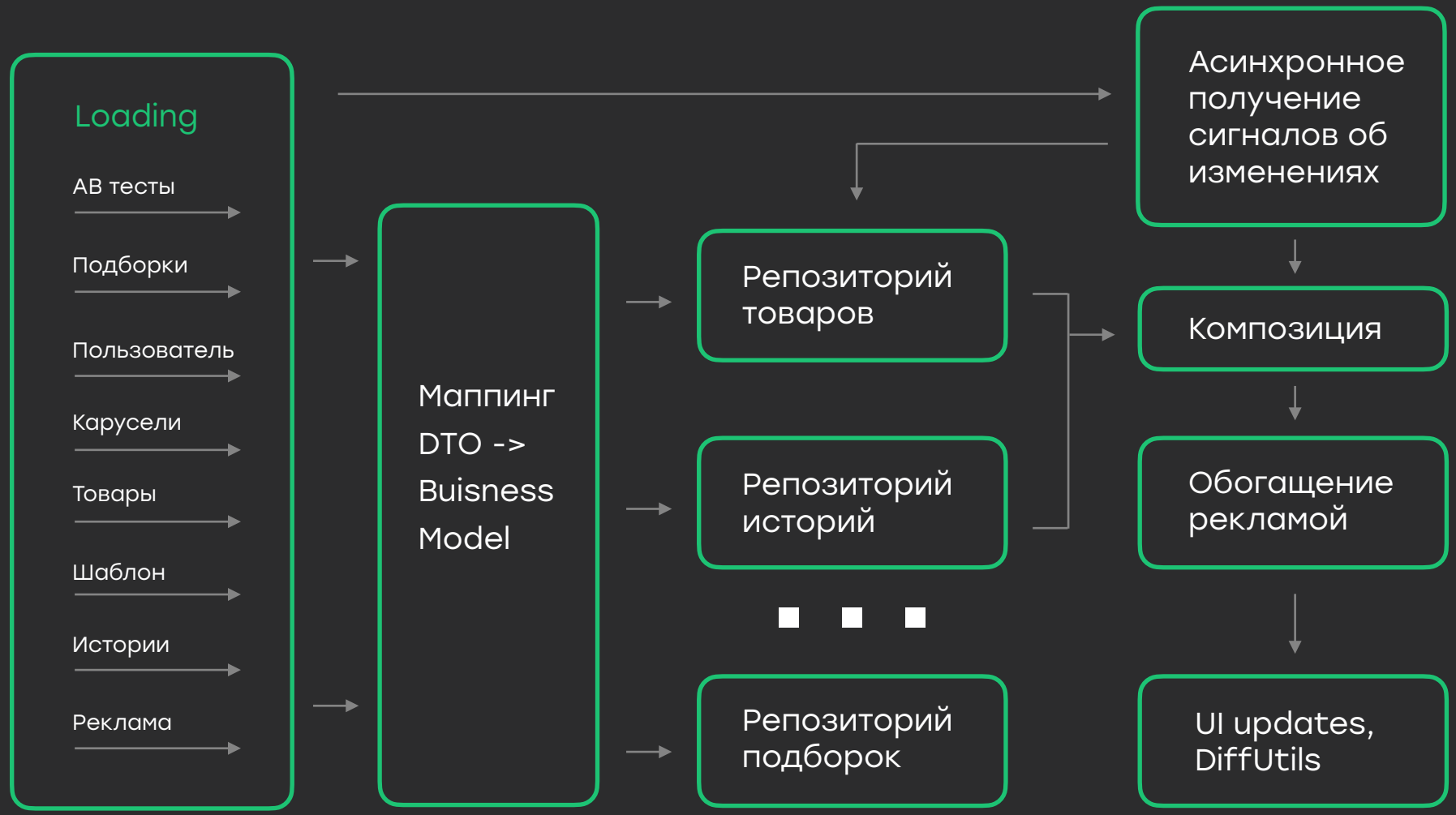
YOULABS

Принятие решения



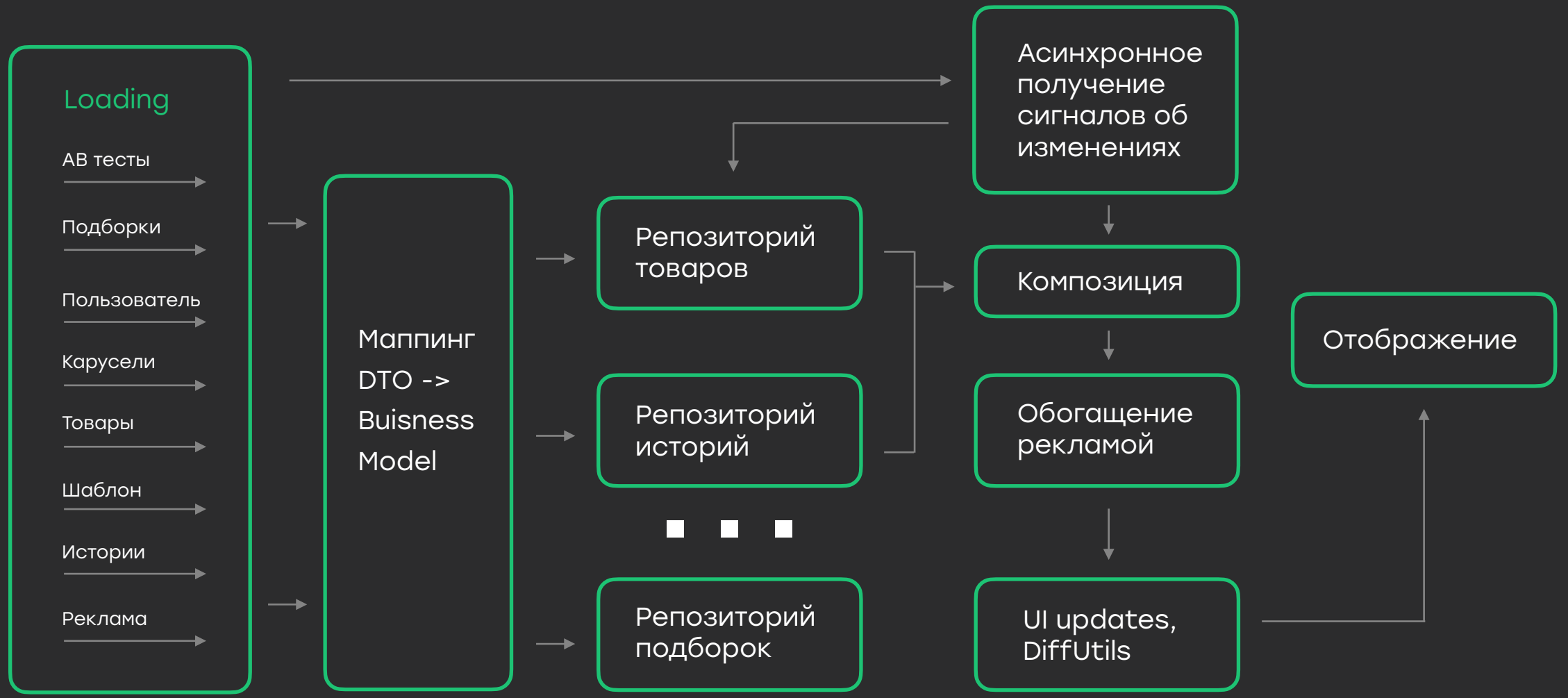
YOULABS

Принятие решения



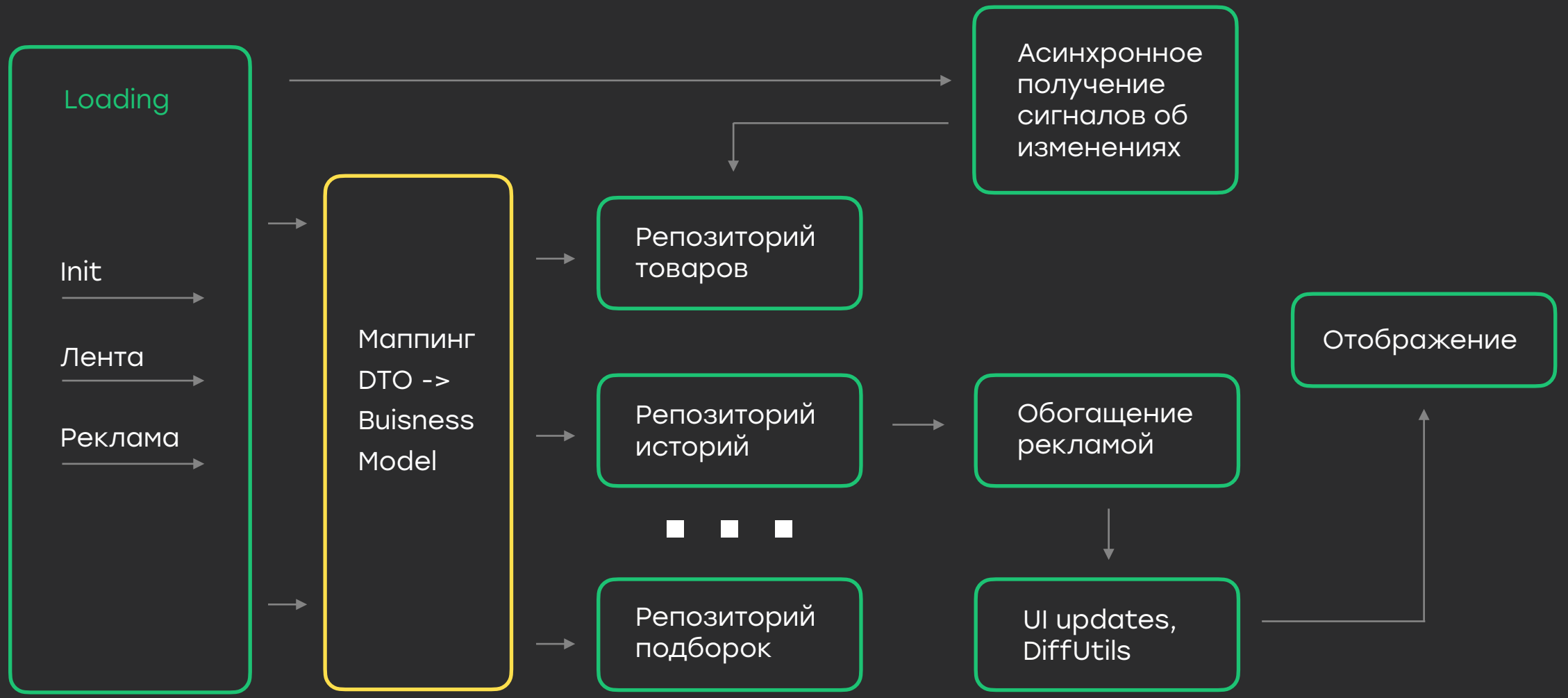
YOULABS

Принятие решения



YOULABS

Принятие решения

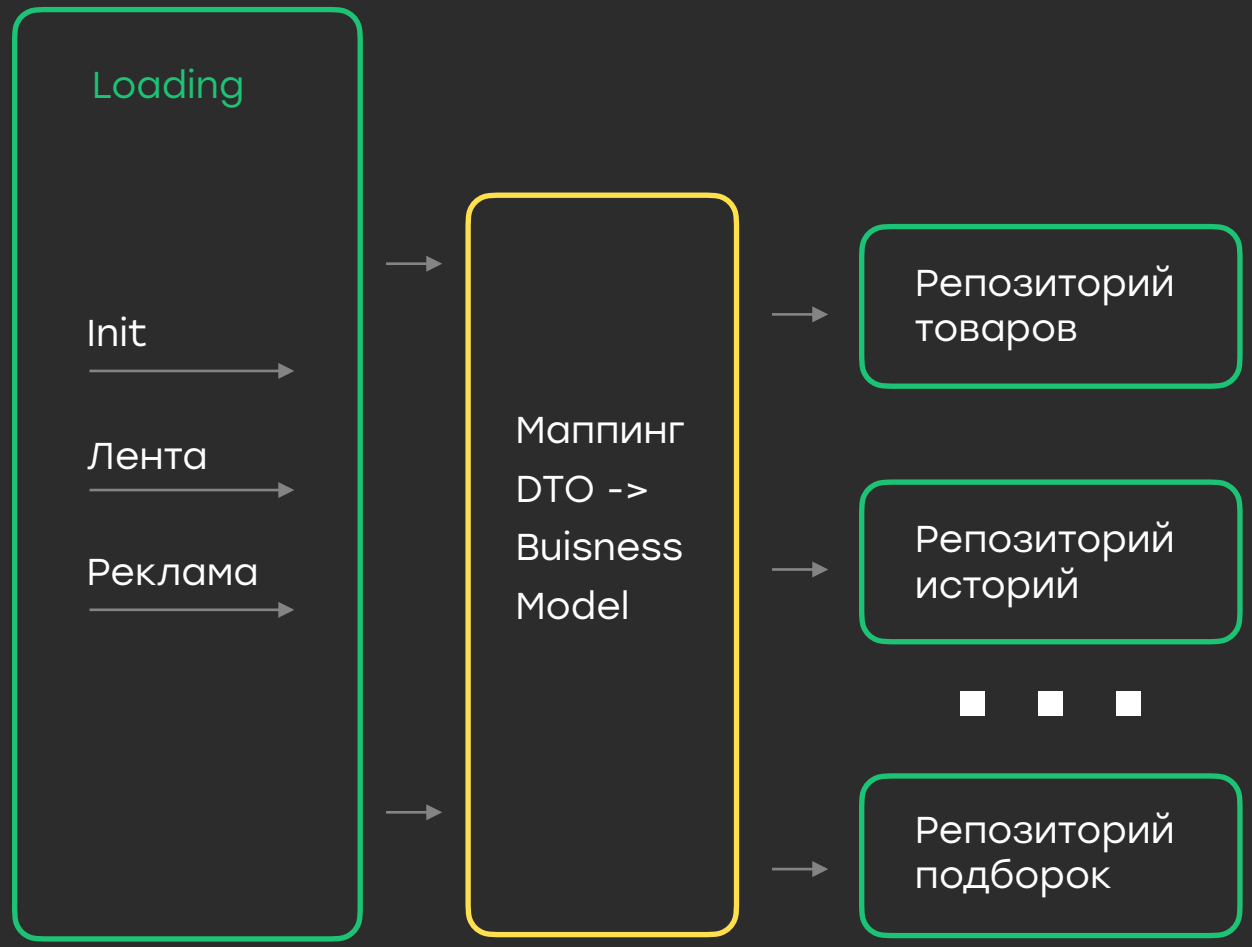


YOULABS

Принятие решения

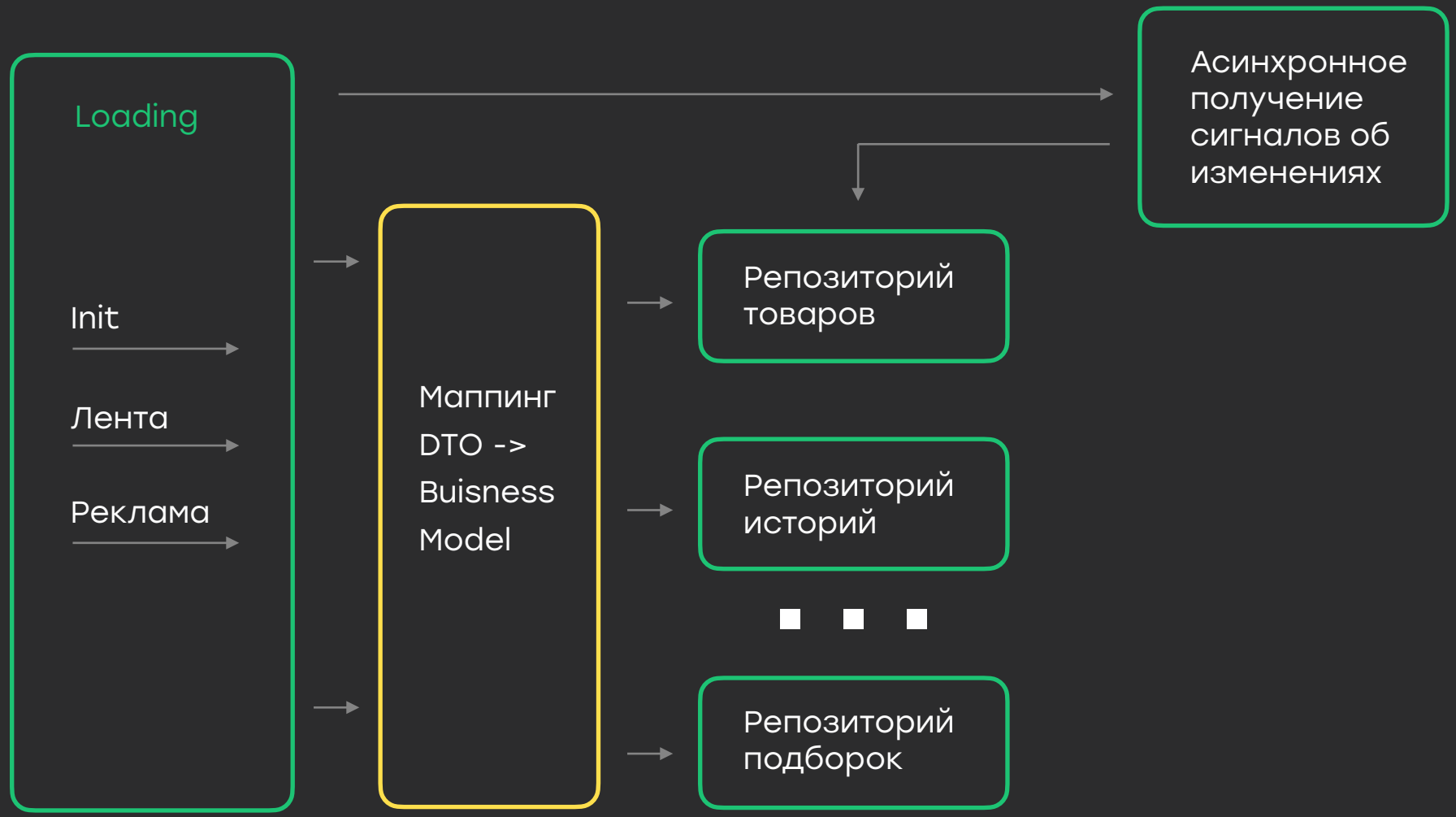
YOULABS

Принятие решения



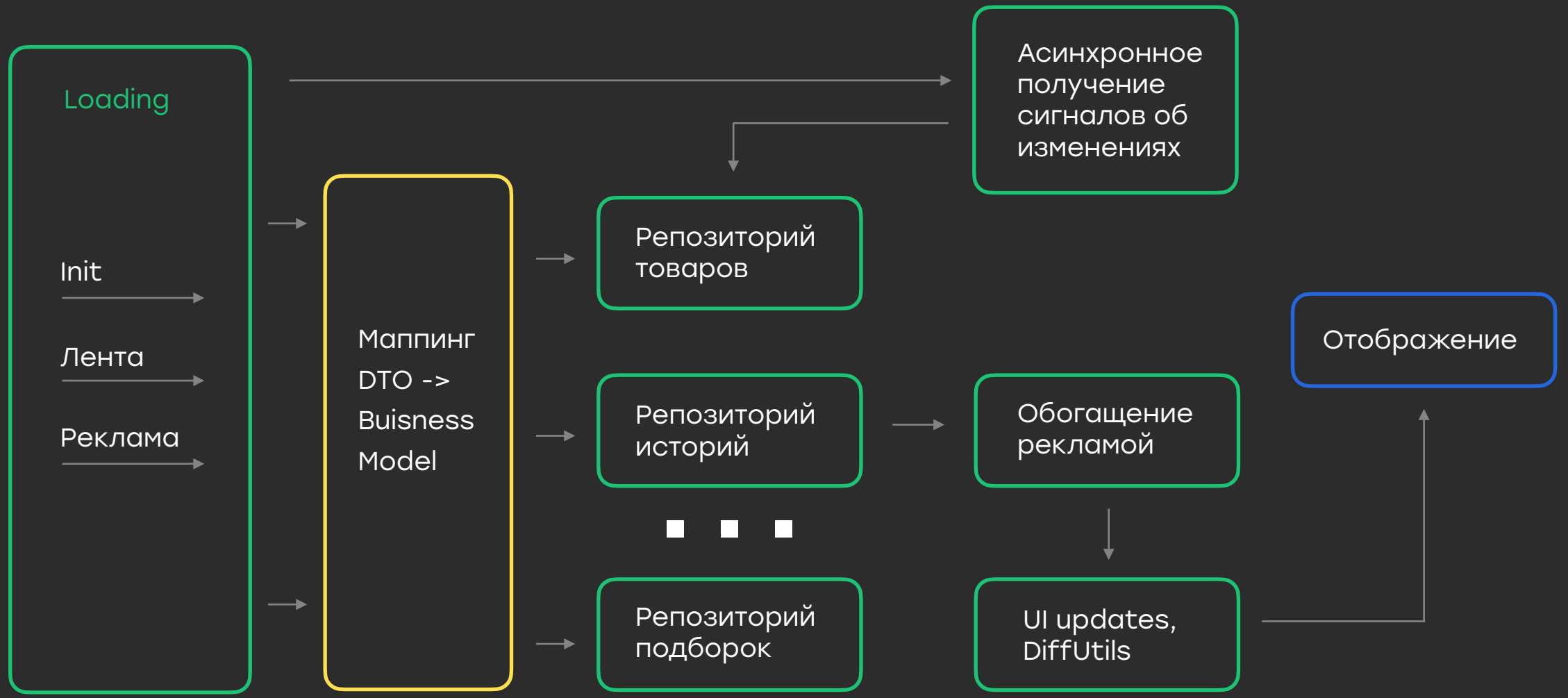
YOULABS

Принятие решения



YOULABS

Принятие решения

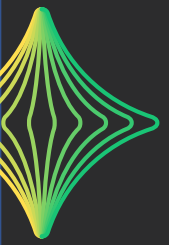


YOULABS

Вредный совет #3

Не меняйте формат

YOULABS

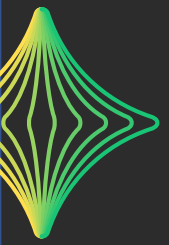


Принятие решения

- Прототип с замерами – 4 дня

Для продакшена:

- Сформировали новую схему

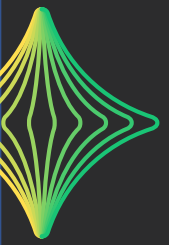


Принятие решения

- Прототип с замерами – 4 дня

Для продакшена:

- Сформировали новую схему
- Реализовали на бэкенде

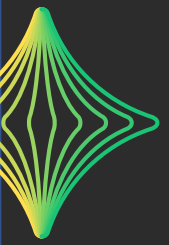


Принятие решения

- Прототип с замерами – 4 дня

Для продакшена:

- Сформировали новую схему
- Реализовали на бэкенде
- Осуществили оптимизации



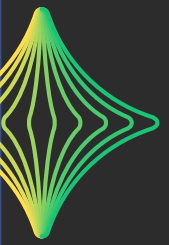
Принятие решения

- Прототип с замерами – 4 дня

Для продакшена:

- Сформировали новую схему
- Реализовали на бэкенде
- Осуществили оптимизации

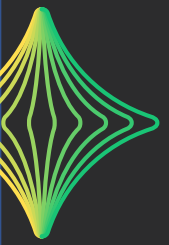
Итого: 1 месяц



Запуск

Рассчитывали на -5-10%
от времени основной метрики

YOULABS

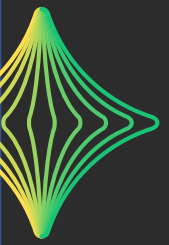


Запуск

Рассчитывали на -5-10%
от времени основной метрики

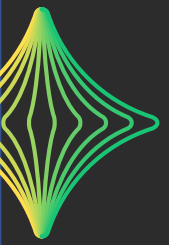
FAIL

YOULABS



Причины

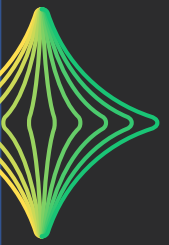
- Человеческий фактор: в релиз попали задачи, ухудшившие запуск приложения, не связанные с GQL



Причины

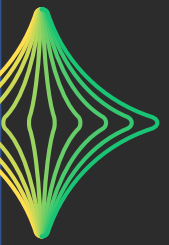
- Человеческий фактор: в релиз попали задачи, ухудшившие запуск приложения, не связанные с GQL

- Плохое основание для сравнения. Предыдущие релизы были в GooglePlay короткое время (самые плохие девайсы не успели «подтянуться»)



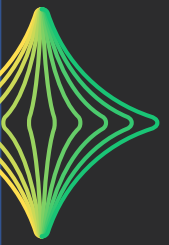
Причины

- Человеческий фактор: в релиз попали задачи, ухудшившие запуск приложения, не связанные с GQL
- Плохое основание для сравнения. Предыдущие релизы были в GooglePlay короткое время (самые плохие девайсы не успели «подтянуться»)
- Эксперименты с отключением функциональности на главной странице



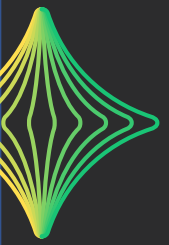
Причины

- Человеческий фактор: в релиз попали задачи, ухудшившие запуск приложения, не связанные с GQL
- Плохое основание для сравнения. Предыдущие релизы были в GooglePlay короткое время (самые плохие девайсы не успели «подтянуться»)
- Эксперименты с отключением функциональности на главной странице
- Не учли формирование параметров для аналитики



Итоги

- На клиенте стабильная метрика от версии к версии
- Сошлись по трудозатратам с командой бэкенда
- Есть куда расти в плане быстродействия, как бэкенду, так и клиентам, не теряя в функциональности



Ссылки

https://graphql.org

https://www.apollographql.com

https://www.apollographql.com/docs/devtools/cli/

https://github.com/apollographql/apollo-android

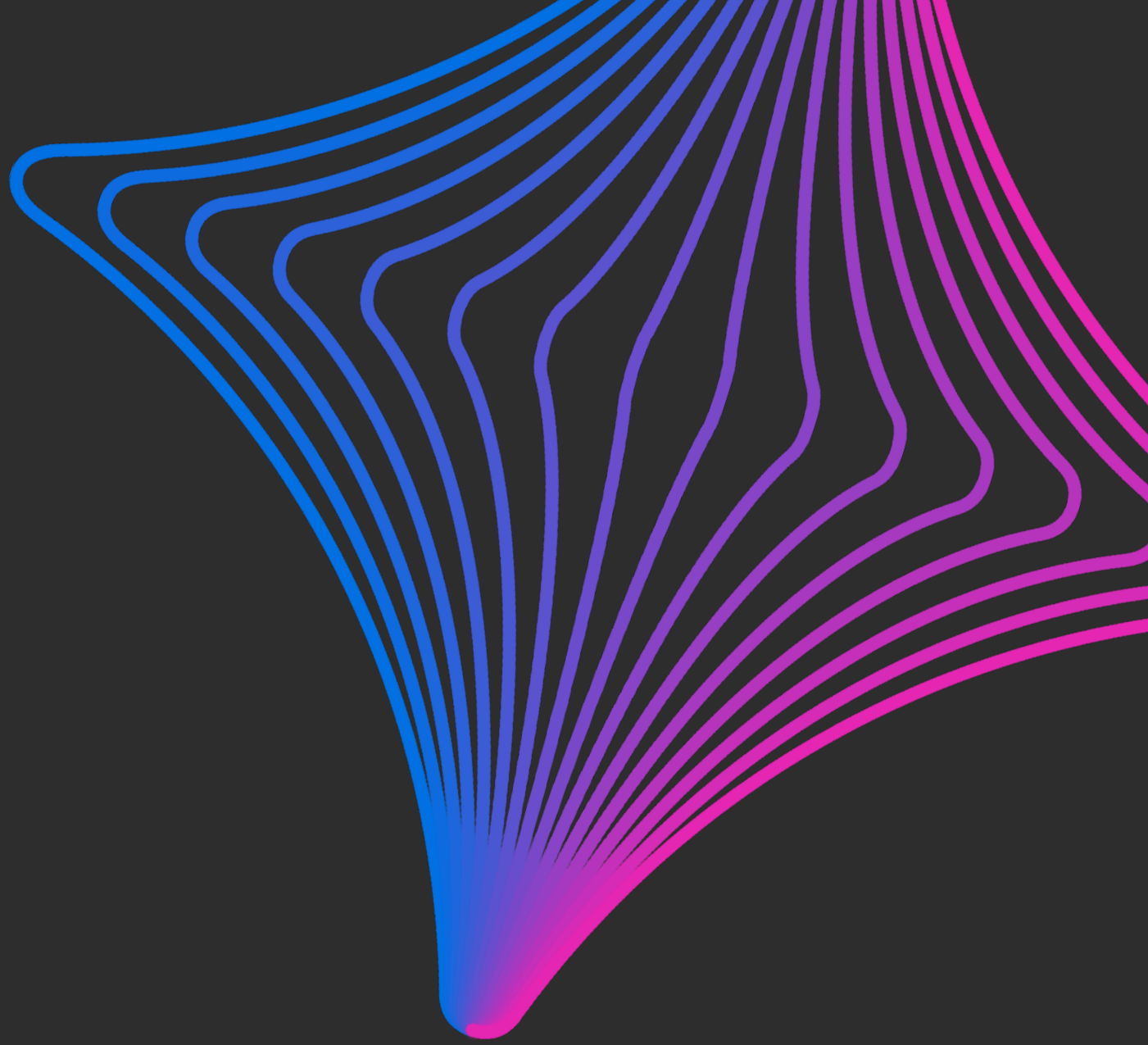
https://graphql.org/swapi-graphql/

https://relay.dev

https://github.com/excitement-engineer/ktor-graphql

https://github.com/xvar/mobius-graphql-demo

Спасибо
за внимание



YOULABS