Developer Circles

Все тонкости композиции и роутинга в Relay Modern

Rudenko Alexander

Senior Front End Engineer / Raiffeisen Bank
Community Lead / Facebook Developer Circle: Moscow,
PhD

facebook for developers

Any solution is usable

Any solution is usable while it works good!!!

We have some sketch…

Slider with new products

Carousel with most popular products

News List

What is SPA?

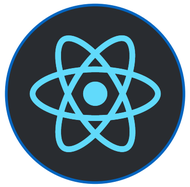
Where?

Performance

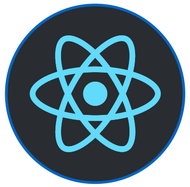
What?

How?

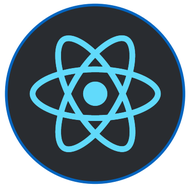
What is SPA?

	Performance	Services
Where?		
What?		⚙ Auth ⚙ i18n Locations ⚙ Routing ⚙ Chats ⚙ Monitoring
How?		 

What is SPA?

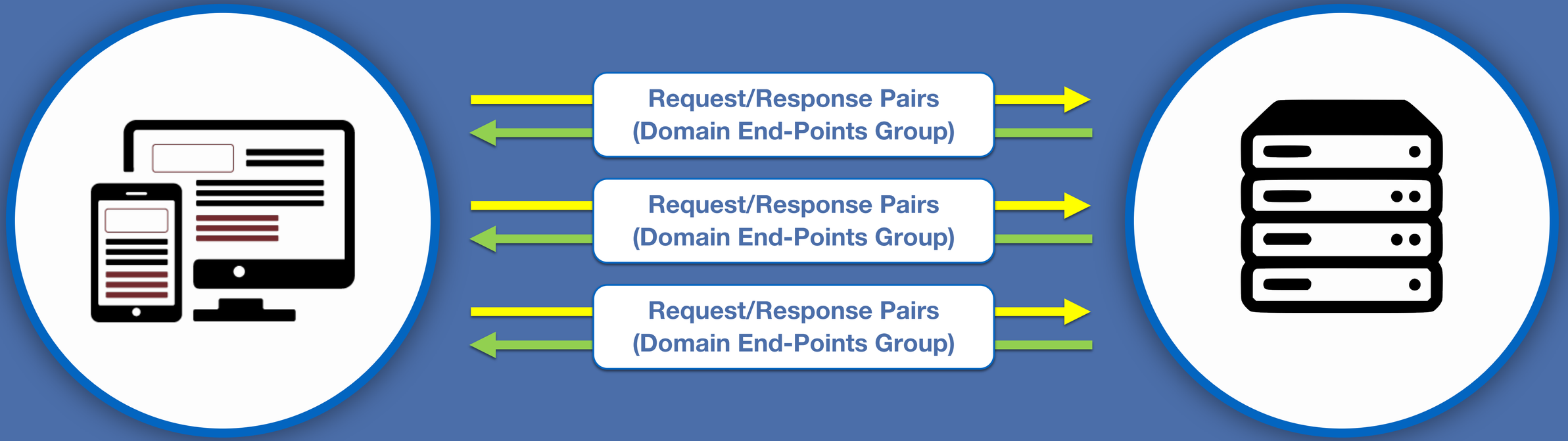
	Performance	Services	Store	Domains
Where?				
What?		⚙ Auth ⚙ i18n Locations ⚙ Routing ⚙ Chats ⚙ Monitoring	❑ Cookies ❑ Local Store ❑ Session Store ❑ IndexedDB ❑ Memory	
How?		 	   	 Data Structures

What is SPA?

	Performance	Services	Store	Domains	Network
Where?					
What?		⚙ Auth ⚙ i18n Locations ⚙ Routing ⚙ Chats ⚙ Monitoring	❑ Cookies ❑ Local Store ❑ Session Store ❑ IndexedDB ❑ Memory		➢ String ➢ File
How?		 	   	 Data Structures	➢ Ajax ➢ Fetch API ➢ Sockets

2

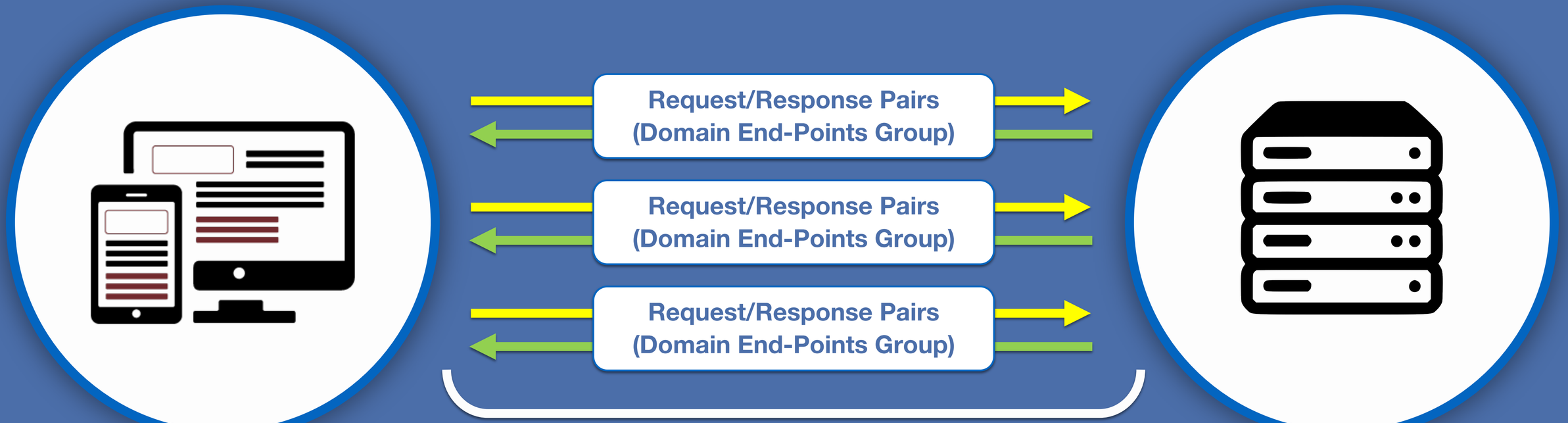
Network Process in Classic SPA



Multiple Clients

Back End Platform

Network Process in Classic SPA



Multiple Clients

{ REST API }

Back End Platform

1. Великое множество end-point’ов

1. Великое множество end-point’ов

2. Store при неумелой разработке может превратиться в помойку

1. Великое множество end-point’ов
2. Store при неумелой разработке может превратиться в помойку
3. Верификация и интроспекция данных

1. Великое множество end-point’ов
2. Store при неумелой разработке может превратиться в помойку
3. Верификация и интроспекция данных
4. Проблемы с pre-fetching и over-fetching

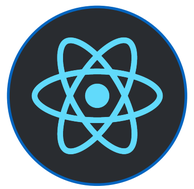
Network Process with GraphQL

Multiple Clients

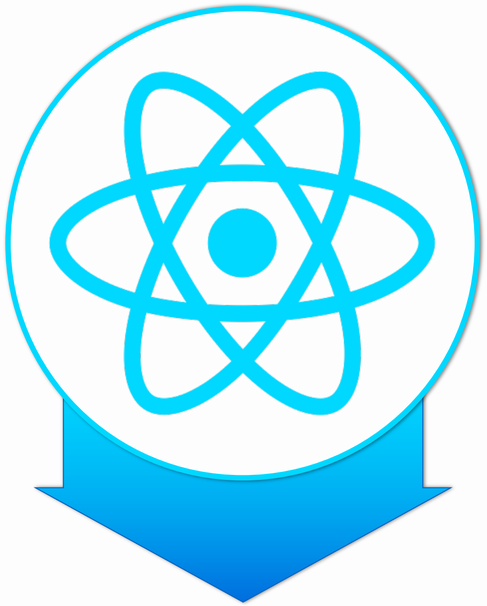
Back End Platform

Network Process with GraphQL

What has changed with GraphQL?

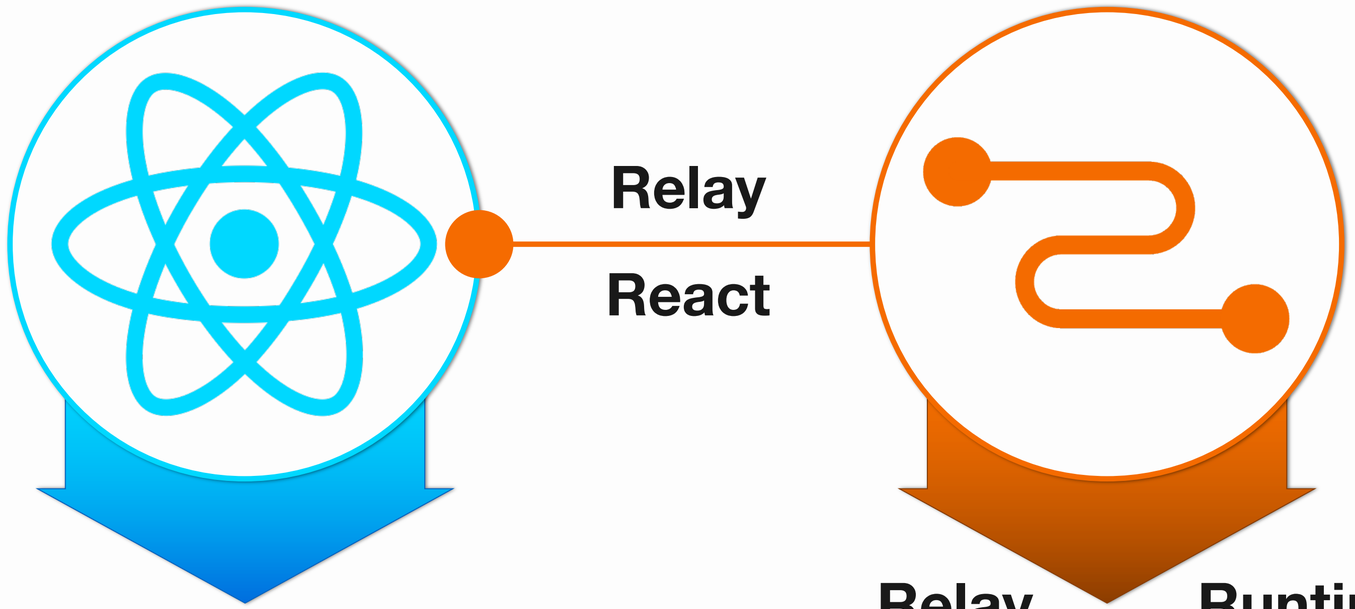
Where?	Performance	Services	Store	Domains	Network
What?		⚙ Normalization ⚙ Subscriptions ⚙ Garbage collector	❑ Cookies ❑ Local Store ❑ Session Store ❑ IndexedDB ❑ Memory	❑ Collections ❑ RecordSource ❑ Record	JSON in String
How?				Schema.graphql 	Through Environment 

What is Relay Modern?



Class
&
Functional
Components

What is Relay Modern?



Class
&
Functional
Components

QueryRenderer
&
HOCs

What is Relay Modern?

Class & Functional Components

QueryRenderer & HOCs

Schema + Fragments + directives

Cool!!!

Finally

it happened!!!

Компоненты relay-react

Fragment Container

Refetch Container

Pagination Container

QueryRenderer

We have some sketch…

Slider with new products

We have some sketch…

Slider with new products

Carousel with most popular products

We have some sketch…

Slider with new products

Carousel with most popular products

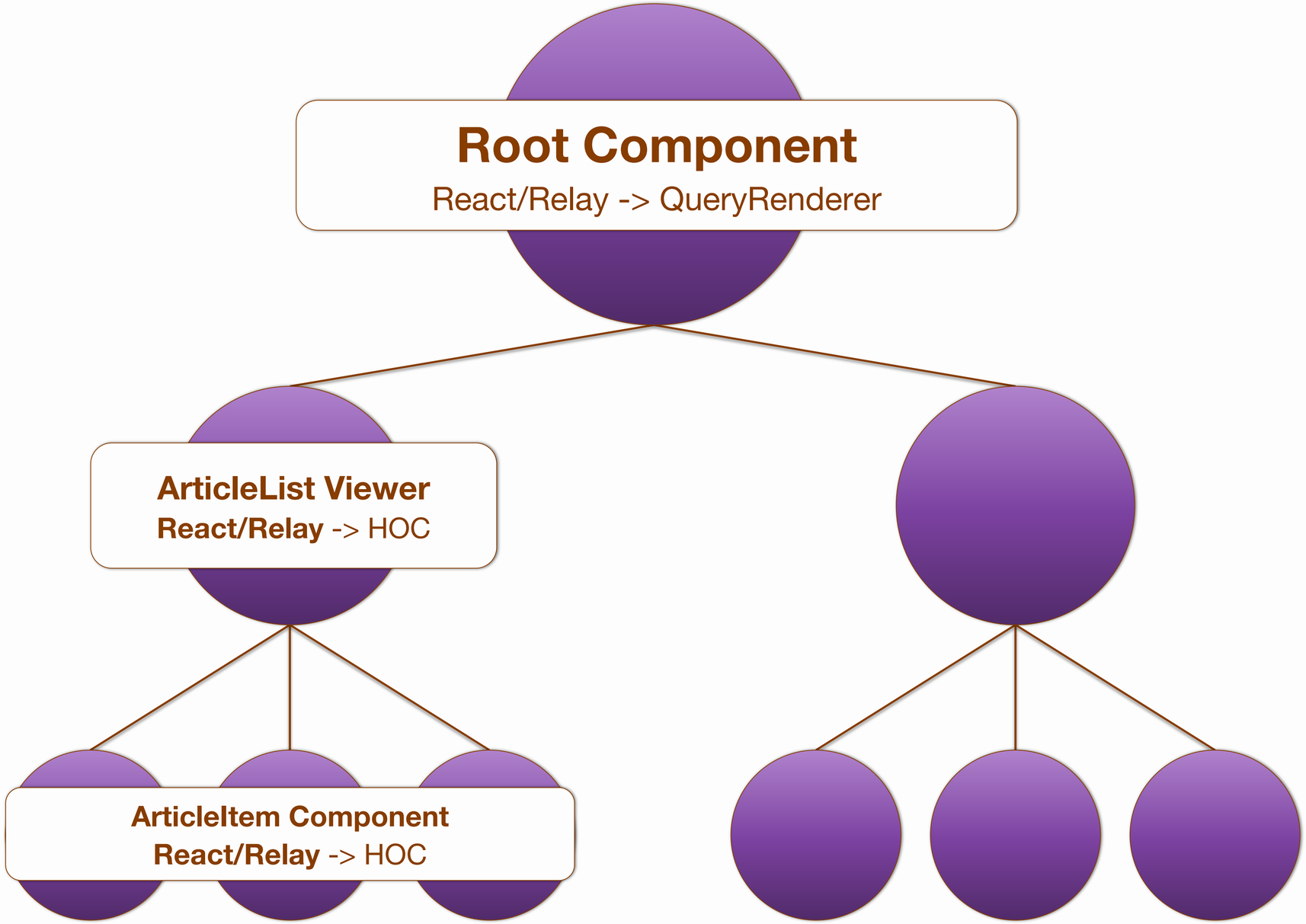
News List

We have some sketch…

Slider with new products

Carousel with most popular products

News List


Root Component
React/Relay -> QueryRenderer
ArticleList Viewer
React/Relay -> HOC
ArticleItem Component
React/Relay -> HOC

Пишем фрагмент…

```
const articleFragment = graphql`
fragment Article_article on Article {
          id,
          ...
```

Пишем компонент...

```
class Article extends React.Component<IProps> {
  public render() {
    return (<Wrapper>
      <Title>
        {this.props.article.title}
      </Title>
          ...
```

Экспортим HOC с компонентом и фрагментом…

```
export default
createFragmentContainer(
    Article, {
    item: articleFragment,
});
```

Пишем фрагмент списка…

```
const articleListFragment = graphql`
fragment ArticleList_list on ArticleList {
  items @connection(
    key: ”ArticleList_list”
  ) {
    edges{
      node {
        id,
        ... Article_article
        ...
```

Пишем компонент и экспортим HOC…

```
import {createFragmentContainer, graphql} from 'react-relay';
class ArticleList extends React.Component { ... }
const articleListFragment = ...;
export default
  createFragmentContainer(
    ArticleList,
    articleListFragment,
);
```

Пишем Query-фрагмент…

```
const newsQuery = graphql`
   query newsQuery {
      viewer {
         ...ArticleList_viewer,
         ...
}}`;
```

Пишем Query-фрагмент…

```
const renderNewsQuery = (
  {error, props, retry}
) => {
  if (error) {
    return ...;
  } else if (props) {
    return <News refetchData={retry} ... />;
  }
    return <div>Loading</div>;
}
```

Оборачиваем в QueryRenderer...

```
<QueryRenderer
  environment={environment}
  query={newsQuery}
  variables={{}}
  render={renderNewsQuery}
/>
```

Оборачиваем в QueryRenderer…

```
<QueryRenderer
  environment={environment}
  query={newsQuery}
  variables={{}}
  render={renderNewsQuery}
/>
```

Оборачиваем в QueryRenderer…

```
<QueryRenderer
  environment={environment}
  query={newsQuery}
  variables={{}}
  render={renderNewsQuery}
/>
```

Оборачиваем в QueryRenderer...

```
<QueryRenderer
  environment={environment}
  query={newsQuery}
  variables={{}}
  render={renderNewsQuery}
/>
```

Relay Modern Antipattern

```
const Article = ({ article }) => (
  <View>
     <Text>{article.title}</Text>
     <Text>{article.text}</Text>
  </View>
);

const  ArticleList = ({ viewer }) => (
  <View>
    {viewer
     .articles.edges.map(({node}) =>
     <Article key={node.id} article={node}
     />))}
  </View>
);
```

Relay Modern Antipattern

```
const Article = ({ article }) => (
  <View>
     <Text>{article.title}</Text>
     <Text>{article.text}</Text>
  </View>
);

const  ArticleList = ({ viewer }) => (
  <View>
    {viewer
     .articles.edges.map(({node}) =>
     <Article key={node.id} article={node}
     />))}
  </View>
);
```

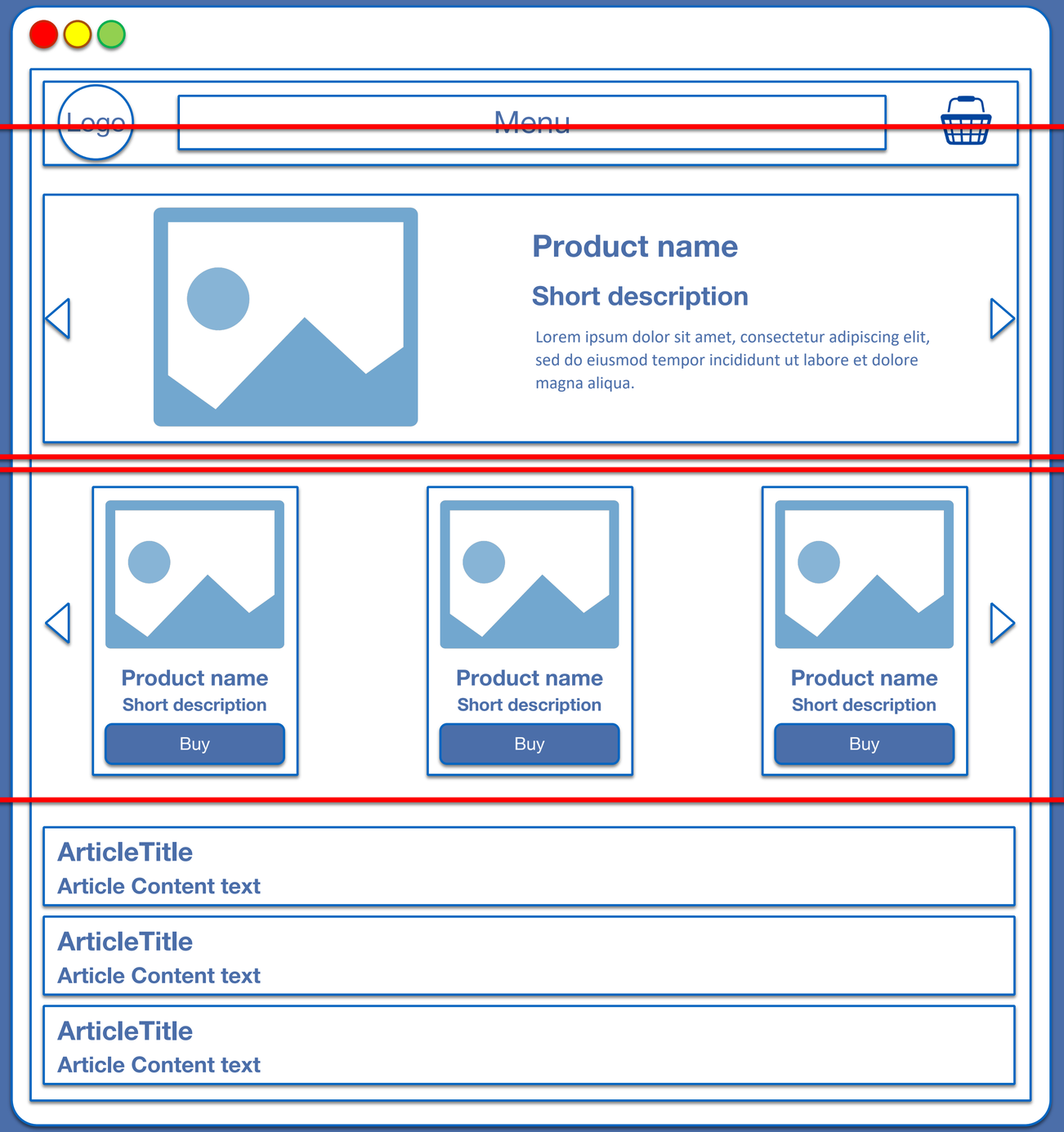
```
const ArticleListFragmentContainer =
   createFragmentContainer(
      ArticleList, {
         viewer: graphql`

   fragment ArticleList_viewer on Viewer {
            articles(first: 10) {
               edges {
                     node {
                        id
                        title
                        text
            }
         }
      }}`})
```

We have some sketch…

Slider with new products

Carousel with most popular products

News List



Relay Modern directives

@argumentDefinitions

@arguments

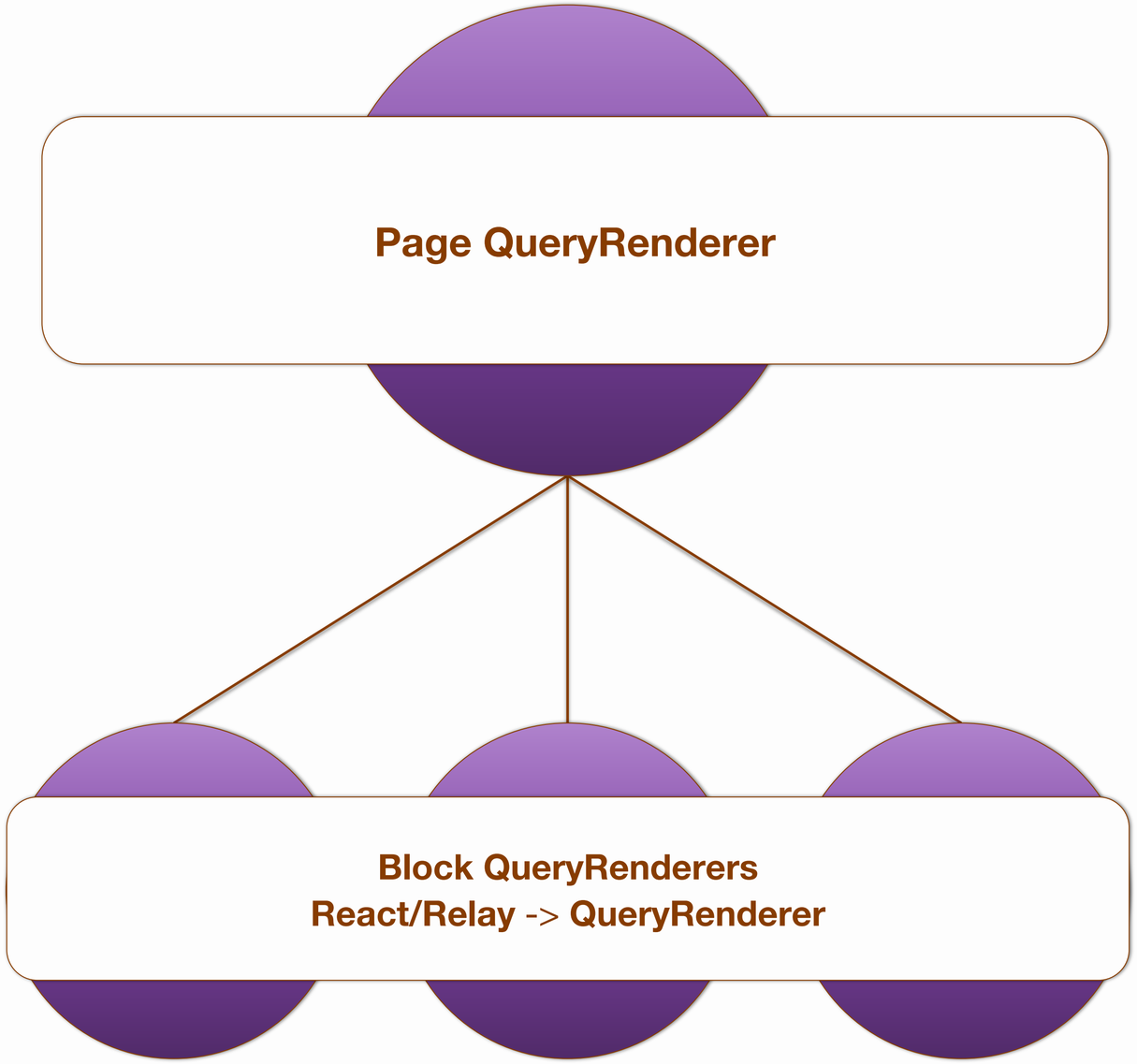
@connection

@argumentDefinitions

```
fragment TopProductList_list on TopProductList
@argumentDefinitions(
  count: {type: "Int", defaultValue: 10},
  isNew: {type: "Boolean"},
) {
  title
  todoItems(isNew: $isNew, first: $count) {
    ...Product_product
  }
}
```

@arguments

```
query TopProductsListQuery($count: Int, $isNew: Boolean) {
    ...TopProductsList_list @arguments(
      count: $count,
      isNew: $isNew
    )
}
```


Page QueryRenderer
Block QueryRenderers
React/Relay -> QueryRenderer

What about Routes!

Found Relay - config

```
import queryMiddleware from 'farce/queryMiddleware';
import Route from 'found/Route';
import makeRouteConfig from 'found/makeRouteConfig';

export const routeConfig = makeRouteConfig(
   <Route path="/” Component={App} query={graphql`query router_App_Query {
         viewer {
            ...App_viewer
         }
      }
   `}>
      <Route Component={ProductsList} query={ProductsListQuery}
         prepareVariables={...} />
   </Route>,
);
```

Found Relay - Client

```
import BrowserProtocol from 'farce/BrowserProtocol';
import { Resolver } from 'found-relay';
import createInitialFarceRouter from 'found/createInitialFarceRouter';
import React from 'react';
import ReactDOM from 'react-dom';
import RelayClientSSR from 'react-relay-network-modern-ssr/lib/client';
```

Found Relay - Client

```
(async () => {
   const resolver = new Resolver(
      createRelayEnvironment(
      // eslint-disable-next-line no-underscore-dangle
      new RelayClientSSR(window.__RELAY_PAYLOADS__), '/graphql',),
   );

   const Router = await createInitialFarceRouter({
      historyProtocol: new BrowserProtocol(),
      historyMiddlewares,
      routeConfig,
      resolver,
   });

   ReactDOM.hydrate(
      <Router resolver={resolver} />,
      document.getElementById('root'),
   );
})();
```

Кто уже использует Relay Modern?

Facebook Social Network

Used on facebook.com, and in the React Native mobile app.

1stdibs

Used on 1stdibs.com e-commerce project

Oculus

Used on oculus.com, Oculus Home in VR, and the React Native Oculus companion app.

Artsy

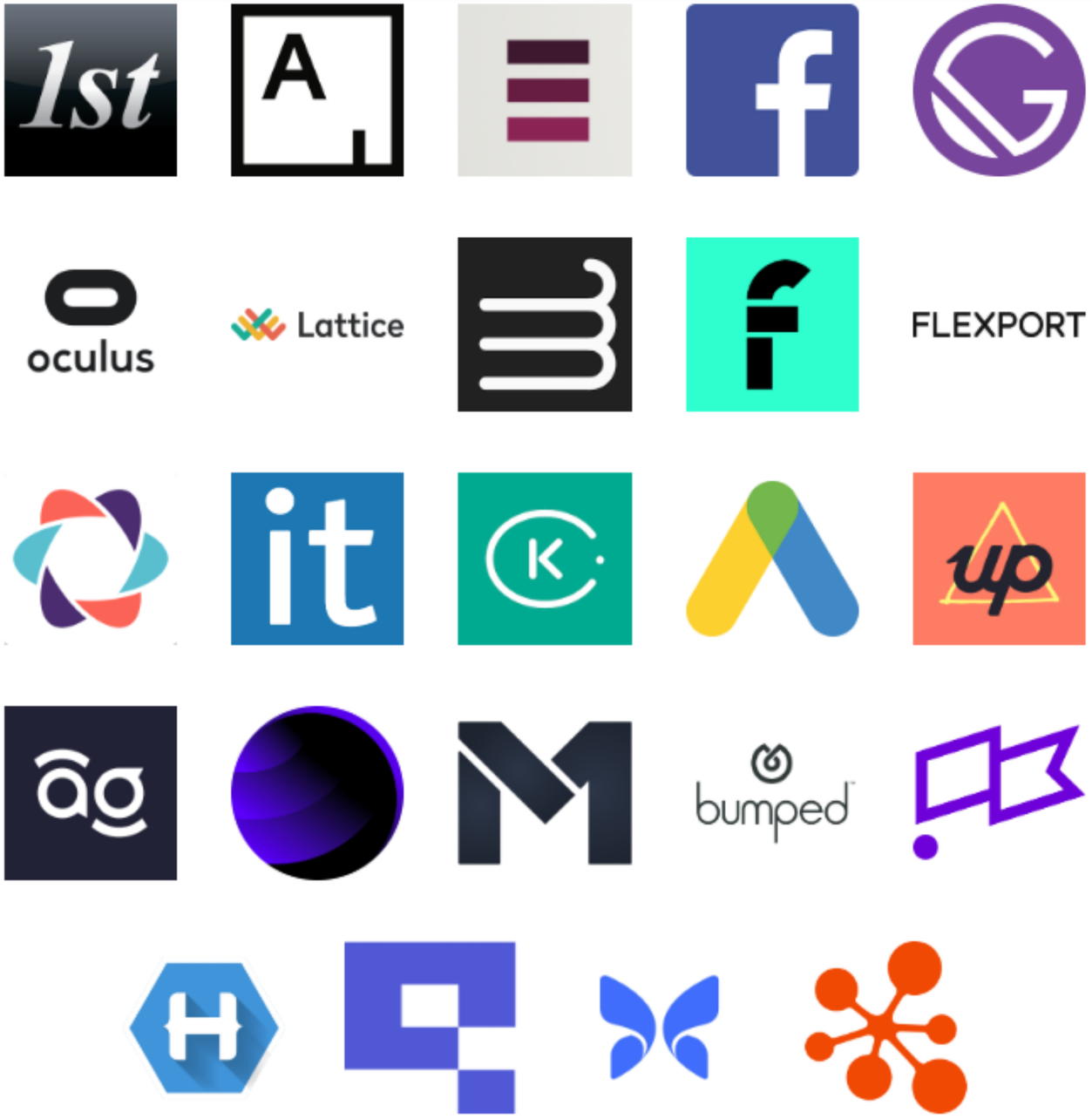
Used on artsy.net, and the React Native iOS app, Eigen.

Gatsby

Underpins the static site generation tool.

Entria

Powers feedback.house.


1st
A
oculus
Lattice
FLEXPORT
it
K
up
bumped

Итого:

1. Relay Modern – современный фреймворк с более чем достаточным количеством возможностей

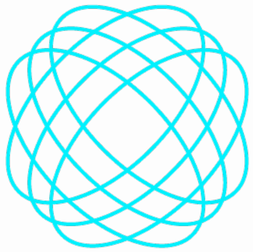
Итого:

1. Relay Modern – современный фреймворк с более чем достаточным количеством возможностей

2. В Relay Modern все строится на композиции и есть удобные механизмы работы с данными

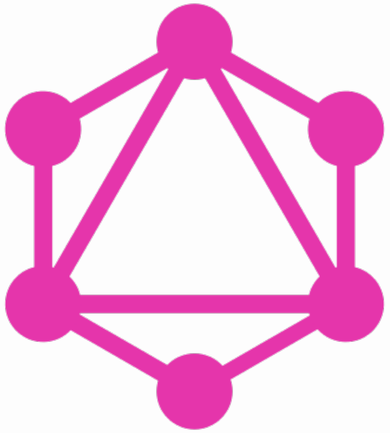
Итого:

1. Relay Modern – современный фреймворк с более чем достаточным количеством возможностей
2. В Relay Modern все строится на композиции и есть удобные механизмы работы с данными
3. При использовании Relay Modern всегда есть несколько возможностей просто и легко работать с данными



Developer Circles

from facebook



GraphQL