

Создание и переиспользование модулей приложения при прототипировании

Используя *Xamarin.Forms*

Продан Станислав 19.11.2020

Что такое AMMA Pregnancy Tracker

Достижения

Количество установок

Россия	Мир
3 млн	10 млн

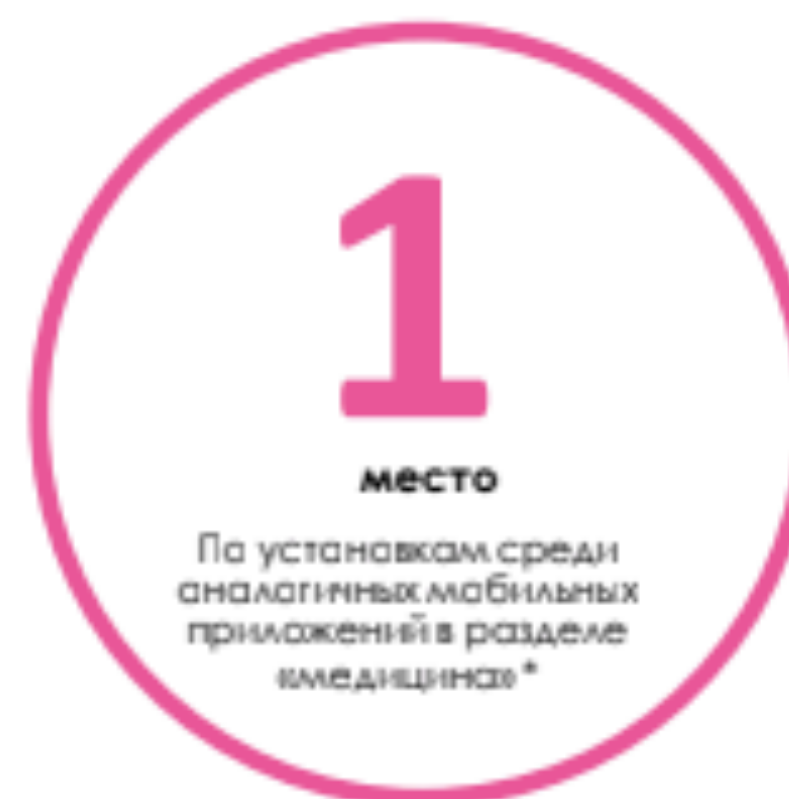
Активных пользователей в

	Россия	Мир
День	80 000	250 000
Месяц	300 000	1,5 млн

Новых пользователей в день

Россия	Мир
3 000	10 000

Google Play	65%
AppStore	35%



Рейтинг в AppStore и Google Play

ТОП 3

Мексика
Аргентина
Россия
Чили
Перу

ТОП 5

СНГ
Малайзия
Уругвай
Греция
Эквадор

ТОП 10

Испания
Латвия
Финляндия
Норвегия
Новая Зеландия

ТОП 15

США
Канада
Австралия
Великобритания
Турция
ЮАР

* По данным ресурса App Annie на дек. 2019 г.

Обо мне

- До 2016 года - разработчик .NET приложений
- До 2020 года- Директор по продукту в AMMA Pregnancy Tracker

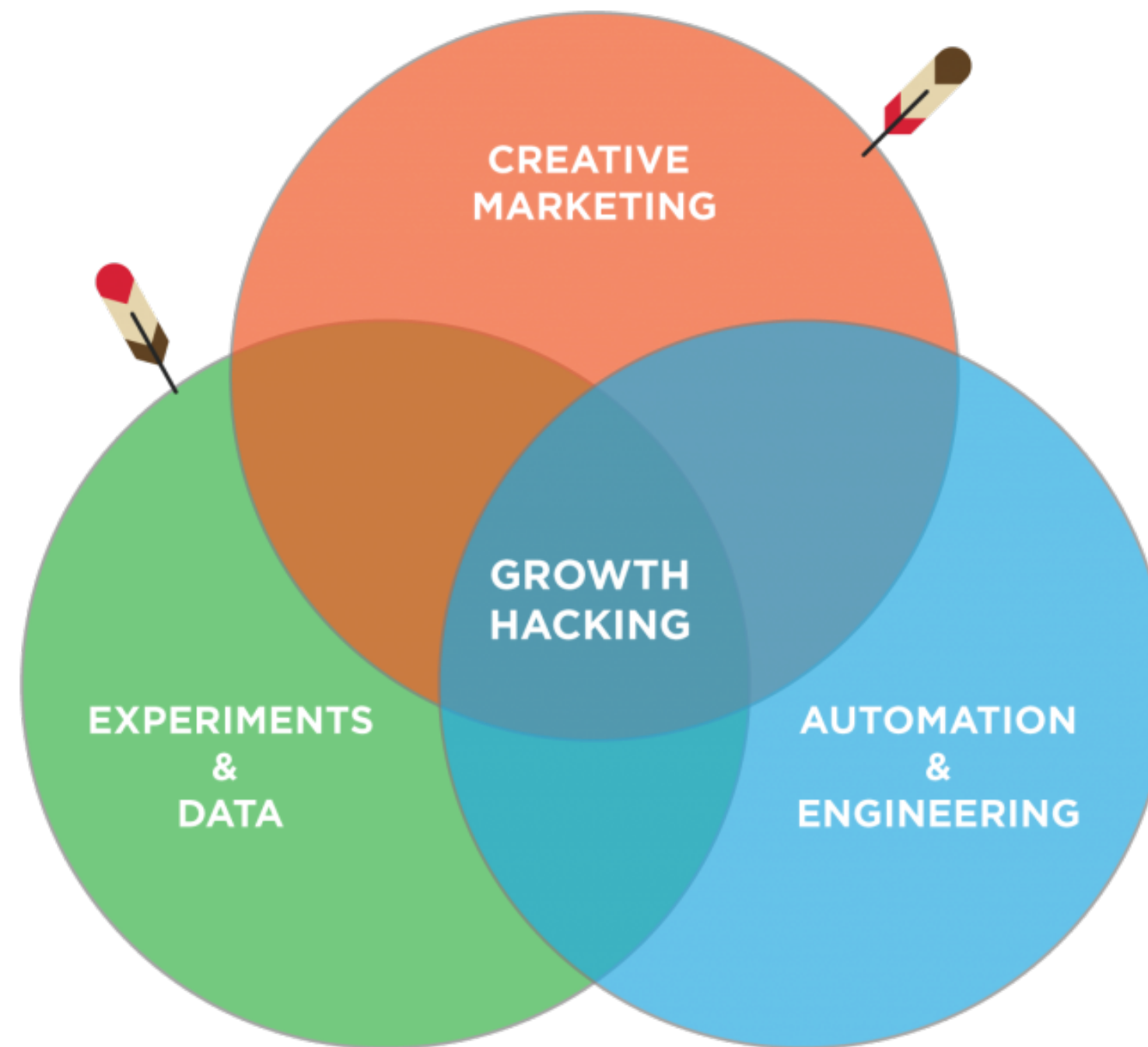


О чем будем разговаривать

- Как жить в эпоху, когда продакт менеджеры хотят много «простеньких» экспериментов
- Как уменьшить время на создание очередного нового проекта для тестирования гипотезы
- С чего начать создание собственной библиотеки компонентов
- Как обеспечить самодостаточность каждого компонента, чтоб его можно было переиспользовать

О продакт менеджменте

Почему иногда нужно «быстро и не качественно»

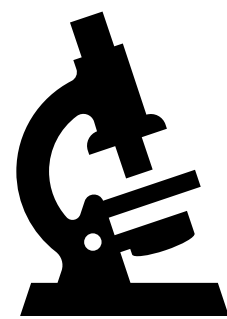


- Кто сможет запустить больше экспериментов, тот лучше узнает свою аудиторию -> сделает более востребованный продукт

Вопросы перед созданием модуля

Разработчикам полезно знать продуктовый менеджмент

- На какой вопрос мы хотим найти ответ, благодаря эксперименту?
- На каких пользователях мы будем тестировать гипотезу?
- На какие продуктовые метрики мы будем смотреть?
- Что будем делать в случае А или в случае Б



или



Эксперименты

Новое приложение

Изменение в
существующем продукте

Изменение настройки в
существующем продукте

Грабли

- Без понятного вопроса - непонятный ответ. На большинство вопросов ответ можно найти и без эксперимента
- Без потока новых пользователей, продуктовые эксперименты не имеют смысла (все тестировать на новых пользователях)
- Всегда разделять эксперименты и целевое решение
- Код из эксперимента не может быть кодом целевого решения.
- По возможности экспериментировать используя альтернативные технологии

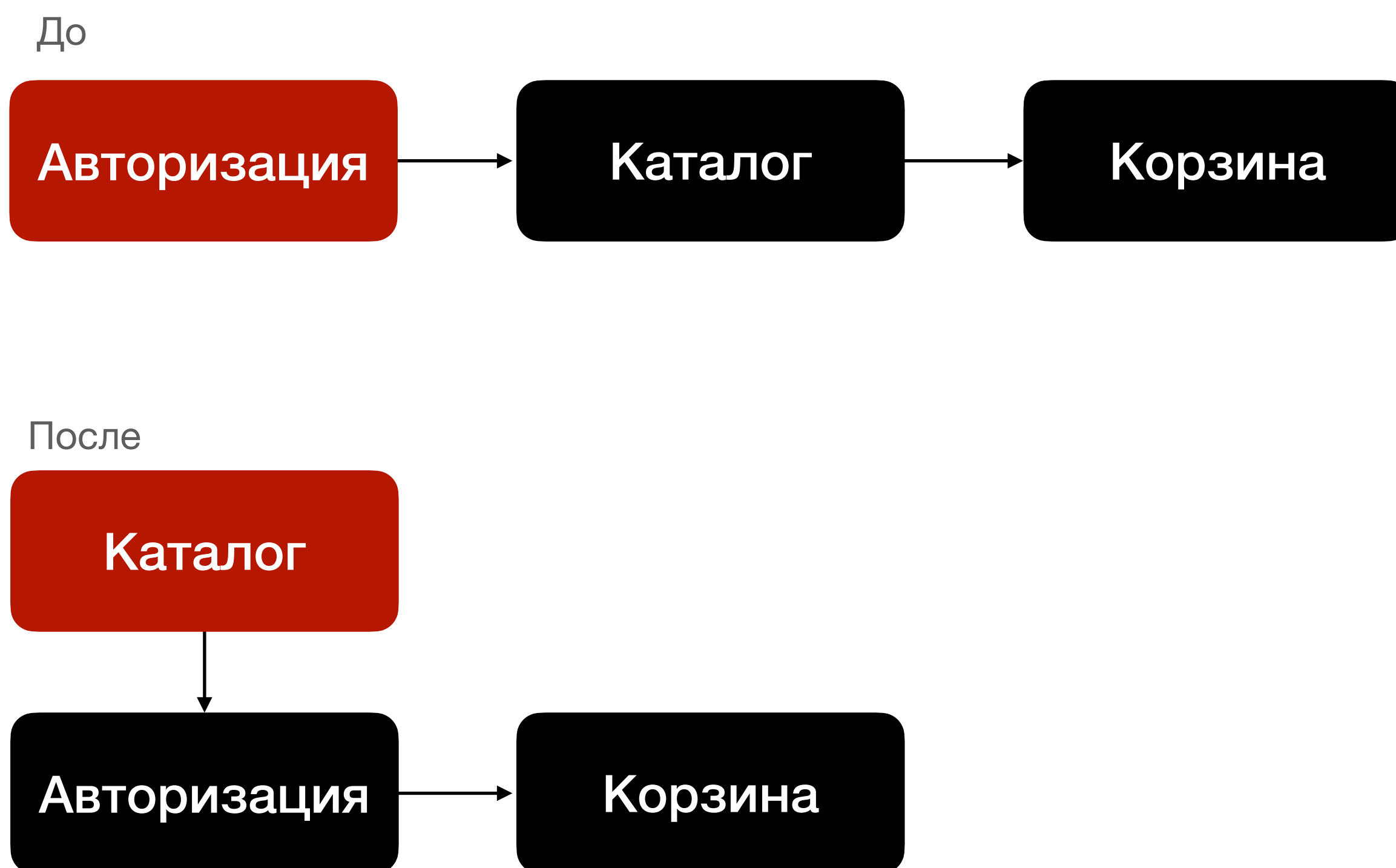
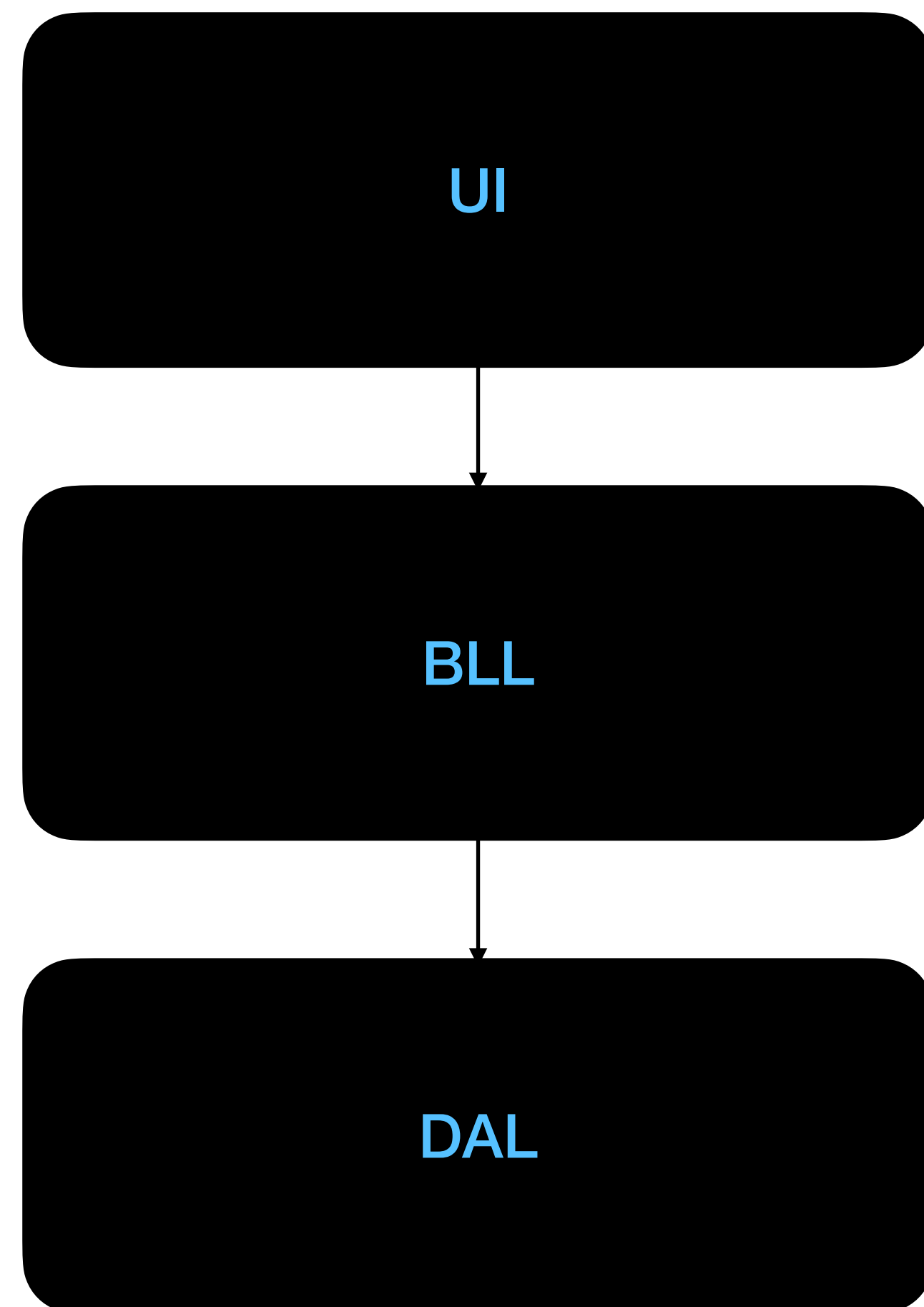


«Код - временное искусство. Как ледяные скульптуры. Хотите оставаться в веках - идите в строительство, а не в программисты»

Андрей Дороничев, Director of Product, Google

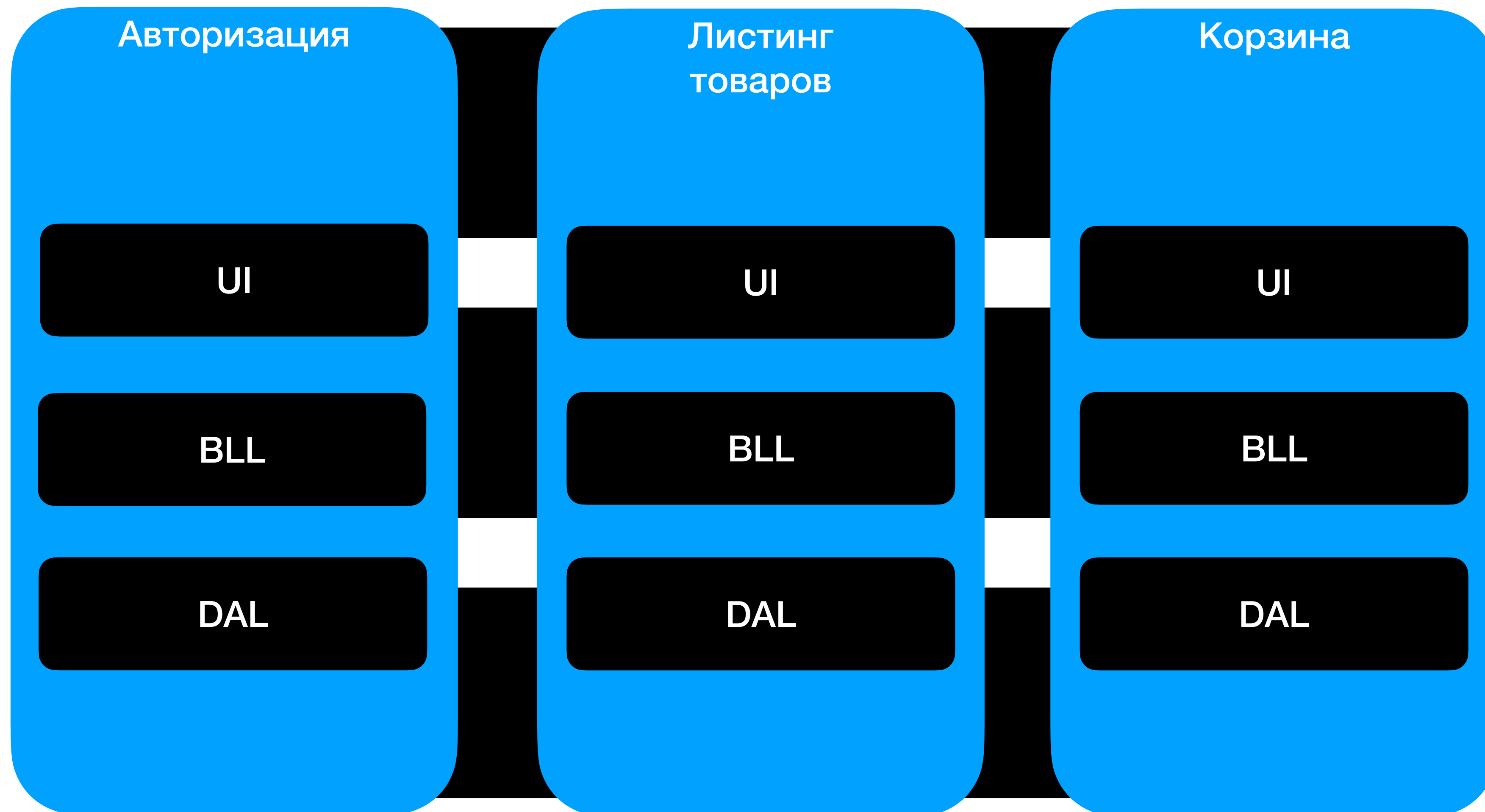
Эскиз

Как обычно подходим к архитектуре мобильных приложений?



Эскиз

При создании самостоятельных контролов



Самодостаточный контрол - это 1 бизнес функциональность, со своим UI BL DATA слоем, которая может существовать как отдельное приложение, так и в виде части другого приложения

UI

Что такое адаптивность?

- Не только любой размер экрана, но и любое место на любом из экранов

```
<?xml version="1.0" encoding="UTF-8"?>
<ContentView xmlns="http://xamarin.com/schemas/2014/forms"
              xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
              x:Class="testcontrols.AuthorizationView">
  <ContentView.Content>
    <Grid>
      <Grid.RowDefinitions>
        <RowDefinition/>
        <RowDefinition/>
      </Grid.RowDefinitions>
      <Label Text="Авторизация"
            FontSize="Large"
            VerticalOptions="CenterAndExpand"
            HorizontalOptions="CenterAndExpand"/>
      <StackLayout Grid.Row="1" Orientation="Horizontal">
        <Entry Text="{Binding Login, Mode=TwoWay}"
              VerticalOptions="CenterAndExpand"
              HorizontalOptions="FillAndExpand"/>
        <Button Text="OK"
              VerticalOptions="CenterAndExpand"
              Command="{Binding RegistrationCommand}"
              BackgroundColor="Red"
              TextColor="White"/>
      </StackLayout>
      <ActivityIndicator Grid.RowSpan="2"
            VerticalOptions="FillAndExpand"
            HorizontalOptions="FillAndExpand"
            Color="Red"
            IsEnabled="True"
            IsVisible="{Binding IsLoading}"
            IsRunning="True"/>
    </Grid>
  </ContentView.Content>
</ContentView>
```

Авторизация

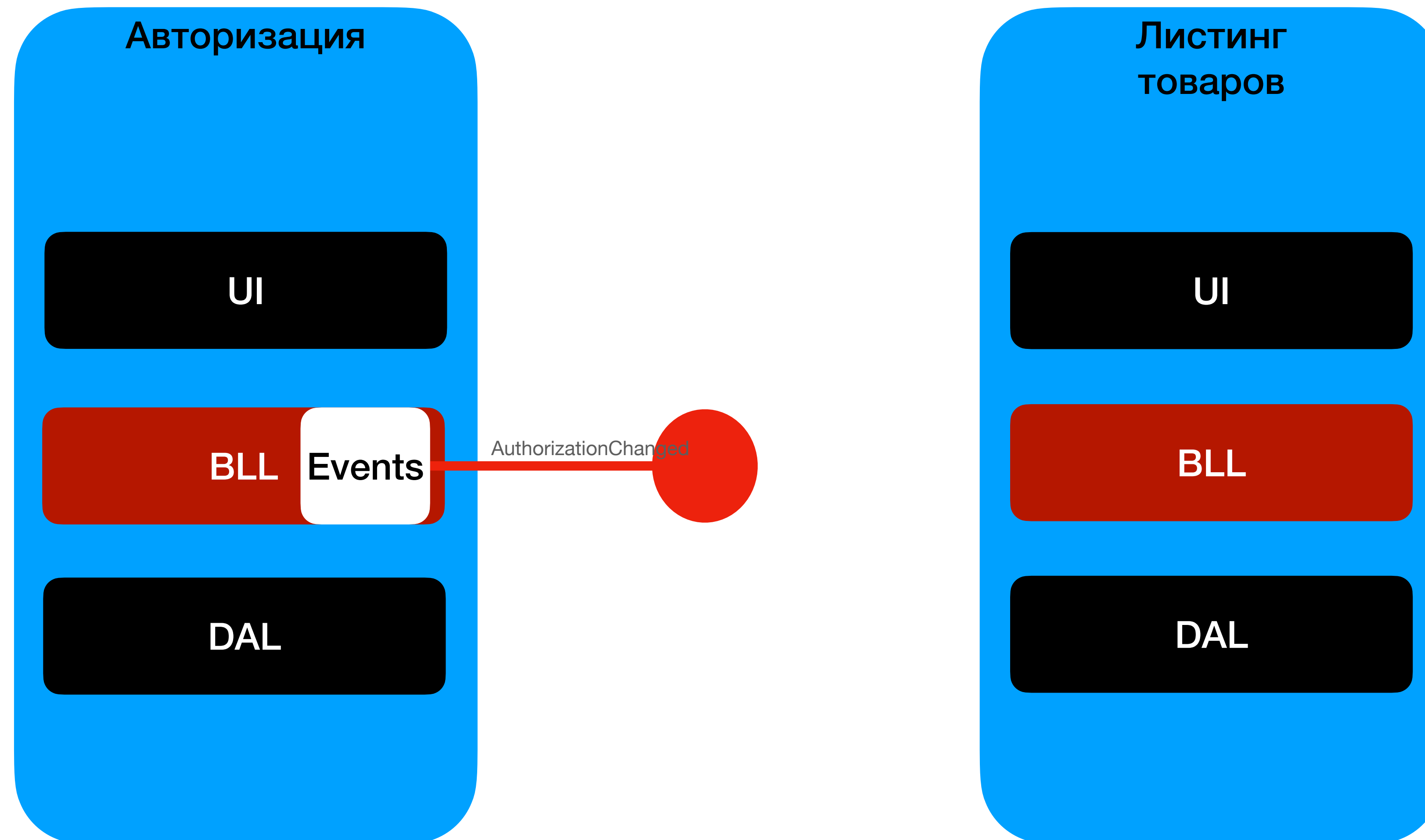
UI

BLL

DAL

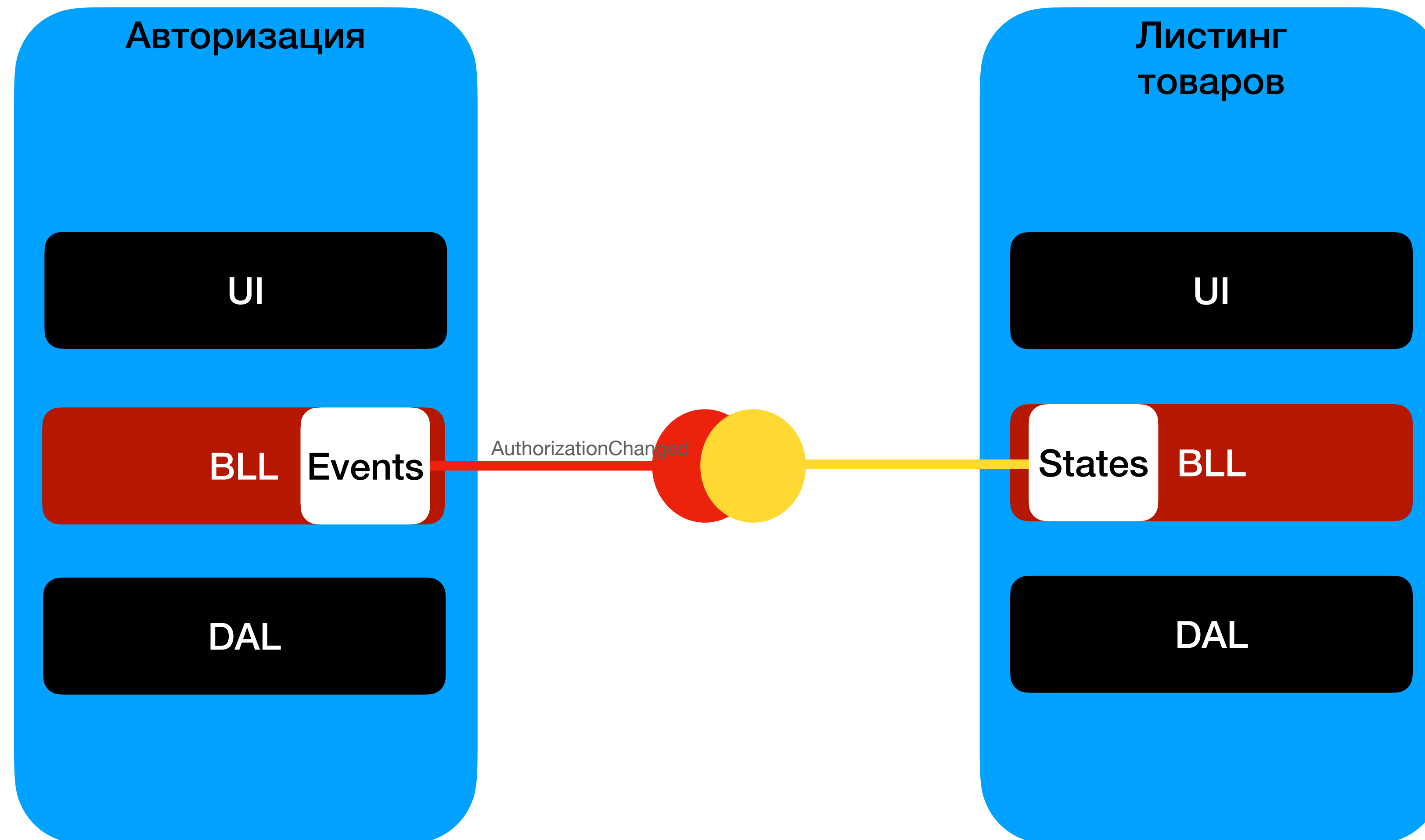
Эскиз

Взаимодействие контролов



Эскиз

Взаимодействие контролов



BLL

Взаимодействие контролов

Сервис AuthorizationService

```
public interface IAuthorizationService
{
    event Action<bool> AuthorizationChanged;
    void Login(string login);
    void Logout();
}
```

```
public void Login(string login)
{
    var loginTicket = _clientRepository.Auth(login);
    loginTicket.OnSuccess += (response) =>
    {
        AuthorizationChanged?.Invoke(response.Data.IsAuthorized);
    };
}
```

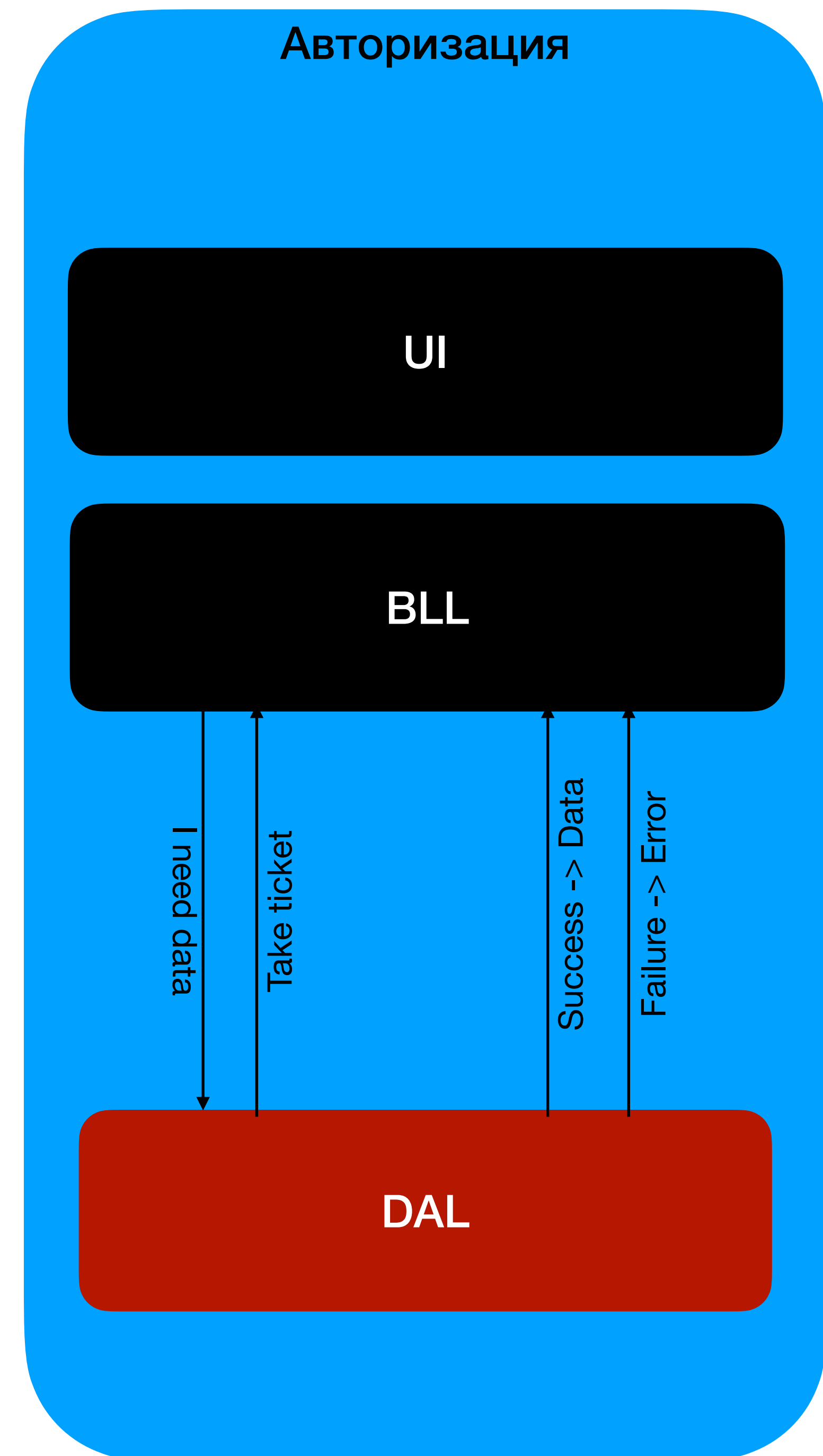
Сервис BasketService

```
public void StartAddingProduct(string sku)
{
    var newProduct = new BasketProduct() { Sku = sku, State = Enums.RequestState.InProgress };
    _products.Add(newProduct);
    var tiket = _basketRepository.AddToBasket(sku);
    tiket.OnSuccess += (response) =>
    {
        if(response.Data.Succseeded != null && response.Data.Succseeded.Value)
        {
            newProduct.PositionId = response.Data.PositionId;
            newProduct.State = Enums.RequestState.Succseeded;
            OnProductAddedSuccessfully?.Invoke(newProduct.Sku, newProduct.PositionId);
        }
        else
        {
            OnProductAddedFailure?.Invoke(sku);
            newProduct.State = Enums.RequestState.Failed;
        }
    };
}
```

Эскиз

Данные

- Нужны данные - обращаемся к DAL
 - DAL возвращает Ticket
 - BLL подписывается на events:
 - Success(data)
 - Success(error)
 - DAL обязательно рейзит один из events
-
- В каждом контролле свой DAL, нельзя обращаться к чужому DAL



DAL

Данные

Тикет

```
public class Ticket<T> where T: IResponseBody
{
    public Response<T> Response;
    public event Action<Response<T>> OnSuccess;
    public event Action<Response<T>> OnFailure;

    public void DoJob()
    {
        //make request
        OnSuccess?.Invoke(Response);
    }
}
```

Авторизация

```
public Ticket<AuthorizationResponse> Auth(string login)
{
    var response = new Response<AuthorizationResponse>();
    var ticket = new Ticket<AuthorizationResponse>();
    ticket.Response = response;
    ticket.DoJob();
    return ticket;
}
```

Использование

```
public void Login(string login)
{
    var loginTicket = _clientRepository.Auth(login);
    loginTicket.OnSuccess += (response) =>
    {
        AuthorizationChanged?.Invoke(response.Data.IsAuthorized);
    };
}
```

Основное приложение

- Добавить контролы
- Настроить взаимодействие
(Правильно подписать одни контролы на другие)

Авторизация

Листинг

Корзина



Основное приложение

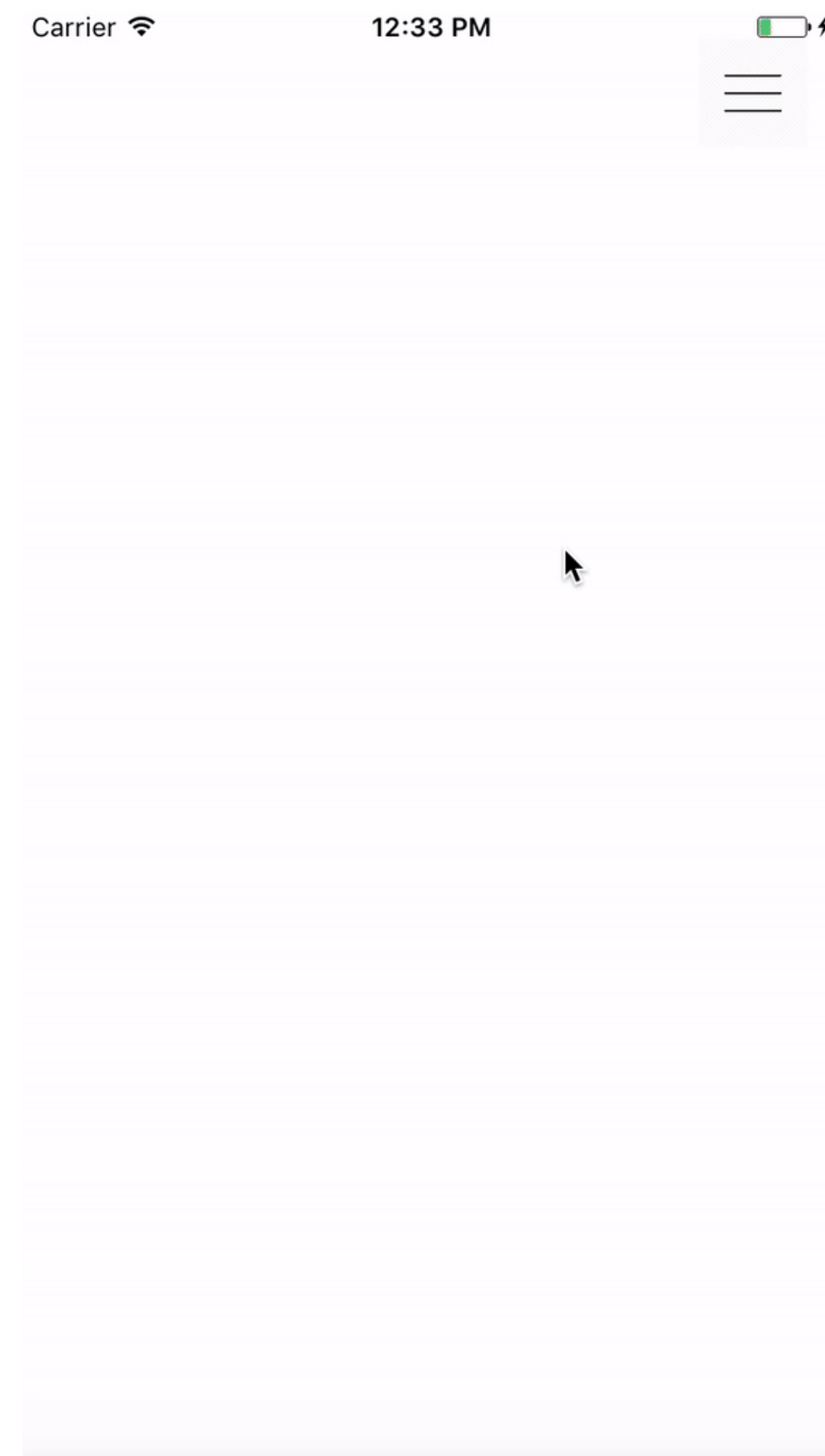
Объявление контролов

```
<Grid>
  <controls:ProductsListing x:Name="listing"></controls:ProductsListing>
  <Grid x:Name="menu"
    VerticalOptions="FillAndExpand"
    HorizontalOptions="End"
    WidthRequest="0"
    BackgroundColor="Teal">
    <controls:AuthorizationView x:Name="auth"
      VerticalOptions="CenterAndExpand"
      HorizontalOptions="FillAndExpand" >

    </controls:AuthorizationView>
    <controls:BasketView x:Name="basket"
      IsVisible="False"
      Margin="0, 200, 0, 0">

    </controls:BasketView>
  </Grid>
  <Button Clicked="Handle_Clicked"
    VerticalOptions="Start"
    HorizontalOptions="End"
    WidthRequest="50"
    HeightRequest="50"
    Margin="0, 20, 20, 0"
    Image="burger.png"
    BackgroundColor="Transparent">

  </Button>
</Grid>
```



Спасибо за внимание

@sprodan

- Github

<https://github.com/sprodan/XamarinControlsToolkit>