

Rhomobile.

Разработка кросс-платформенных мобильных гибридных приложений с использованием Ruby



RailsClub 2016
22 октября 2016 года



Солдатенков Дмитрий

Сооснователь, Технический директор  TECHNOLOGIES

Еще в школе понял, что программирование это то, чем я хочу заниматься.

Работал в одной из первых в России фирм по разработке компьютерных игр. Затем работа в большой команде над крупным проектом для разработчиков. Постепенно сфера интересов сместилась на мобильные платформы, и вот уже более 12 лет я занимаюсь именно этим.

Работал в компаниях

AMI, TogetherSoft, Borland, LG Electronics, TWP, Rhomobile, Motorola Solutions, Zebra Technologies.

Подробнее : <https://www.linkedin.com/in/dsoldatenkov>

E-Mail: dsoldatenkov@tau-technologies.com

Вы любите Ruby и Ruby on Rails ?
Вы хотите создавать мобильные приложения на Ruby ?
Я покажу как это можно делать с open source решением Rhodes !



 <p>MobiRuby</p>	<p>http://mobiruby.org/</p>	 open source 	<p>iOS</p>	<p>Нативное, похоже на Xamarin, Appcelerator ограниченные возможности</p>
 <p>RUBOTO</p>	<p>http://ruboto.org/</p>	 open source 		<p>Нативное, похоже на Xamarin, Appcelerator ограниченные возможности</p>
 <p>RubyMotion</p>	<p>http://www.rubymotion.com/</p>	 	<p>iOS </p>	<p>Кросс-платформенное. Нативное, похоже на Xamarin, Appcelerator. Разработка только на Mac Вполне рабочее решение</p>
 <p>RHOMOBILE SUITE</p>	<p>http://tau-technologies.com/</p>	 open source 	<p>iOS   </p>	<p>Кросс-платформенное гибридное. Похуже на PhoneGap (Cordova), но дополнительно Ruby контроллеры на локальном сервере. Комплексное решение</p>

Потребительские устройства



- Благодаря программе BYOD (принеси свое устройство) сотрудники корпораций могут использовать свои смартфоны и планшеты для доступа в корпоративную сеть и запуска корпоративного ПО
- Также обычные потребительские устройства закупаются для работы сотрудников.
- Полное доминирование iOS и Android, с перевесом iOS. Доля Windows Phone (и прочих версий) менее 10%.

Специализированные устройства



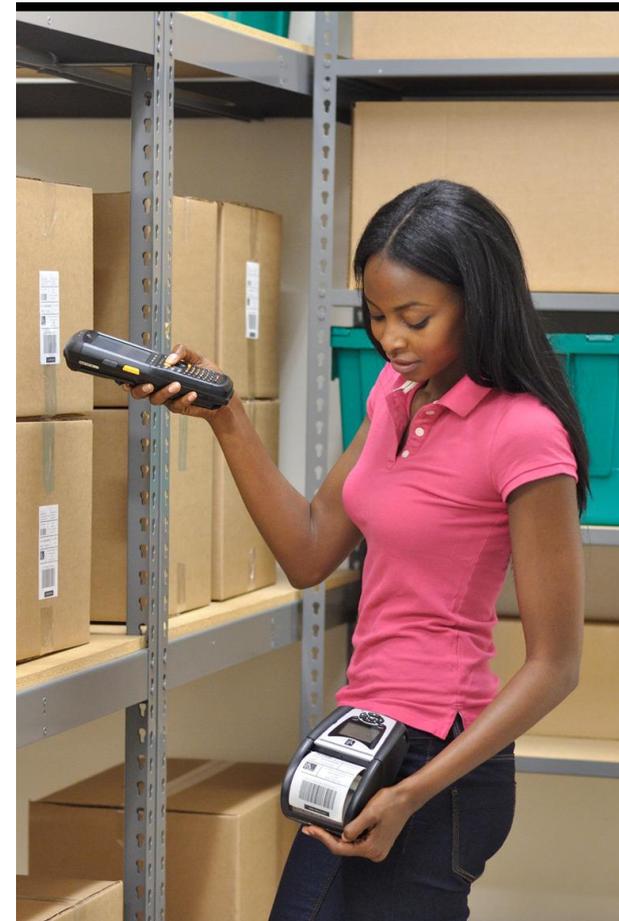
- Консервативный рынок - до сих пор значительная доля продаваемых устройств базируется на Windows Mobile/CE!
- В настоящее время заметна миграция на Android - можно с уверенностью сказать, что Android будет доминировать на этом рынке в ближайшие годы.
- Большое количество разработанного ПО под Windows Mobile/CE

Прежде всего это особенности парка устройств :

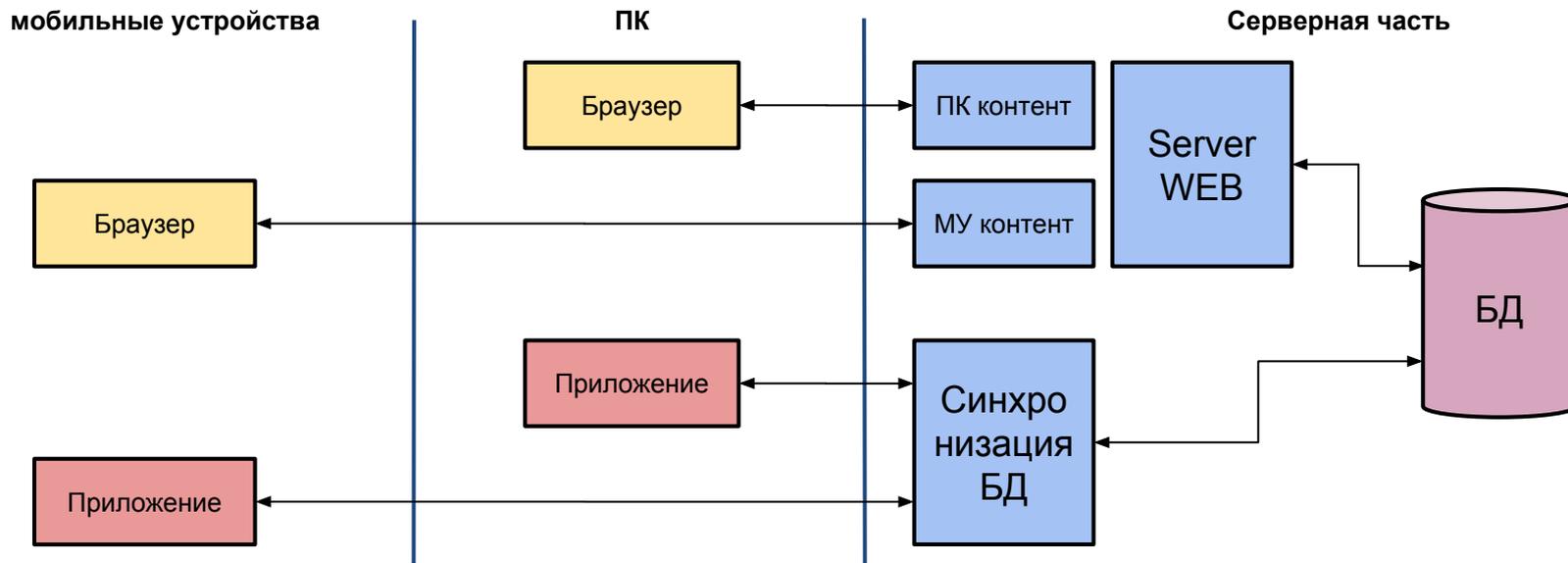
1. Смесь разных платформ как по операционным системам (iOS, Android, WinCE/WM), так и по аппаратным возможностям(слабые WinCE/WM).
2. Наличие специфических аппаратных расширений (сканеры и т.п.)
3. Наличие внешних устройств: RFID сканеров, Barcode сканеров, сканеров карт, портативных принтеров и т.п.

Кроме того:

1. Разработка приложения нужна не всегда ! Иногда достаточно WEB решения.
2. Доступ к аппаратным возможностям (сканеры, принтеры и т. п.)
3. Синхронизация данных (обычно с уже имеющимся бэк-эндом)
4. Мобильное приложение как правило это лишь часть большого интегрированного решения.



Это не просто приложение на мобильном устройстве, а часть большого комплекса ПО, включающего в себя различные части на разных платформах.





- дешевле (меньше кода)
- быстрее разработка (не нужно согласовывать на разных платформах)
- не нужно заботиться о поддержке новых ОС и новых версий ОС
- меньше разработчиков
- можно использовать имеющихся разработчиков и решения (для гибридных)



- хуже производительность (не всегда)
- менее привлекательный интерфейс и функционал (без использования самых свежих возможностей платформ)
- зависимость от поставщика решения (особенно для не open-source)



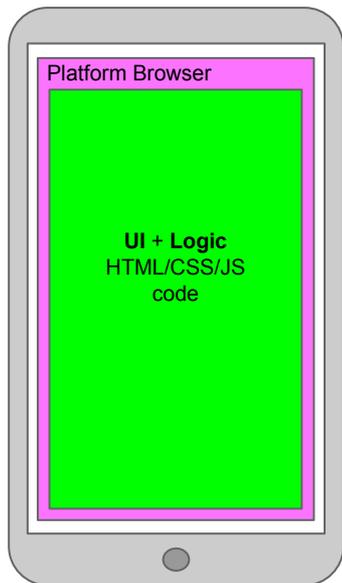
Идеально подходит для разработки в корпоративном секторе !

**Simple not cross-platform
Native Application**



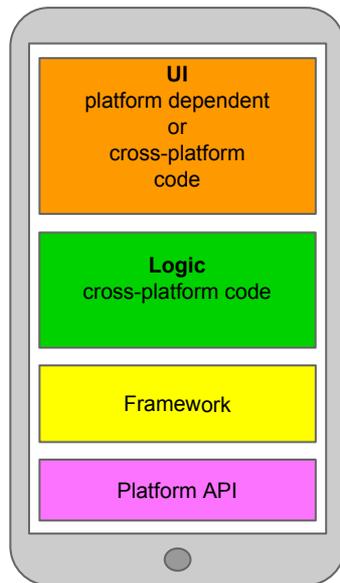
iOS: ObjC, Swift, C++
 Android: Java, C++
 WinCE/WM: C#, C++

**Web
cross-platform Application**



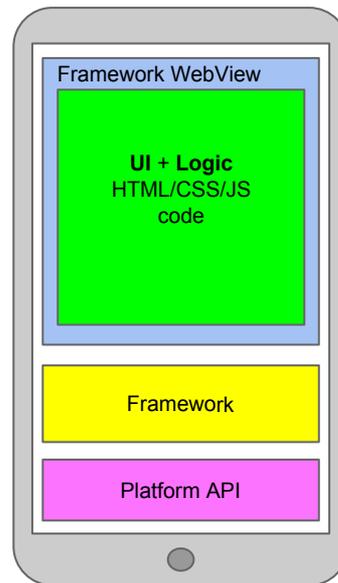
a lot of
 HTML/CSS/JS frameworks

**Native
cross-platform Application**



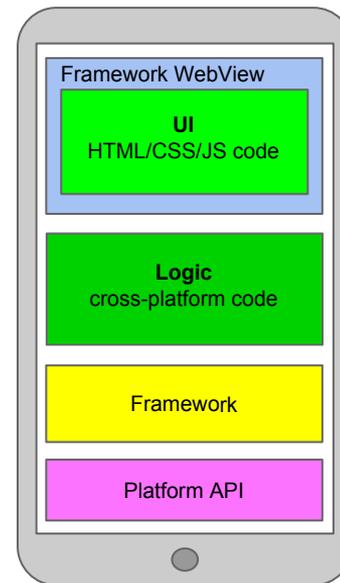
Xamarin (C#)
 Appcelerator (JS)
 React Native (JS)
 NativeScript (JS)
 QT (C++, QML)
 RubyMotion (Ruby)
 CodenameOne (Java)
 Corona (Lua)

**Hybrid
cross-platform Application**



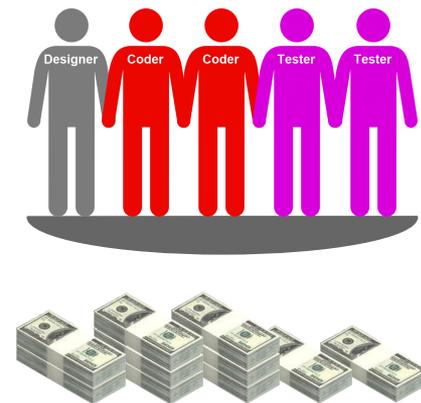
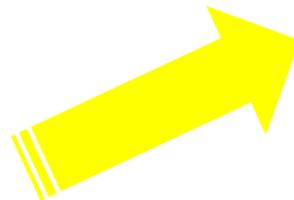
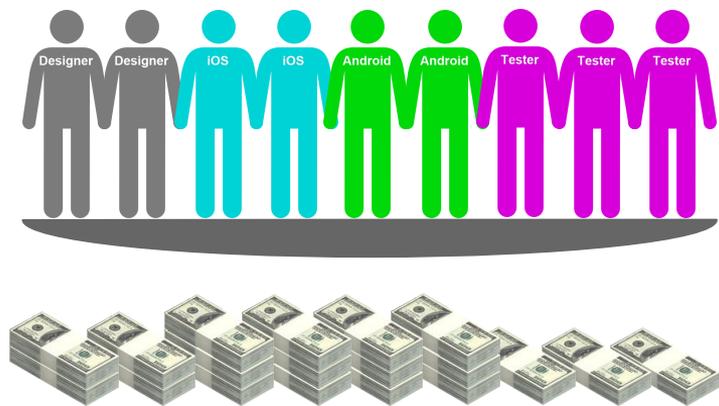
Cordova/PhoneGap Rhomobile
 + a lot of
 HTML/CSS/JS frameworks

**Mixed Hybrid
cross-platform Application**



Rhomobile (Ruby)
 + a lot of
 HTML/CSS/JS frameworks

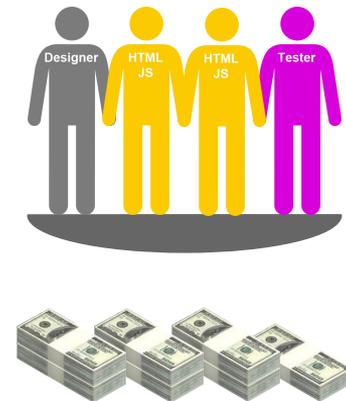
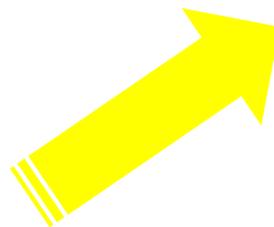
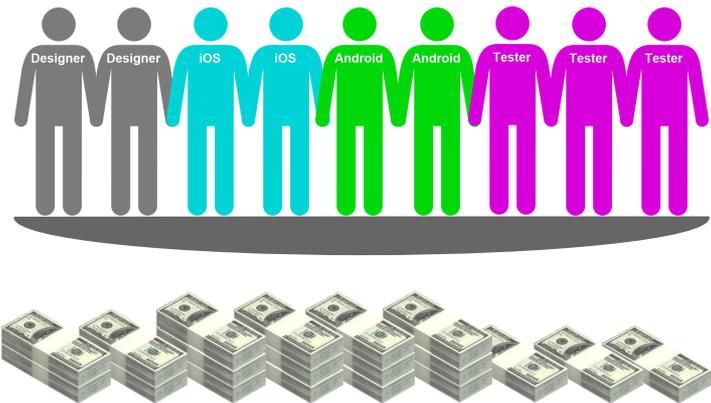
- Кросс-платформенные решения типа Appcelerator или Xamarin позволяют снизить стоимость разработки в два и более раз !



Проблемы негибридных решений :

- Программисты вынуждены осваивать новую специфичную платформу - это проблема
- Продукт привязан к используемой платформе

- Гибридные кросс-платформенные решения типа Cordova/PhoneGap или Rhomobile позволяют снизить стоимость разработки в три и более раз !
- Можно привлекать уже имеющих разработчиков веб порталов и веб приложений



- Можно использовать код веб порталов
- можно легко переносить код между разными решениями - нет жесткой привязки к решению

Мы рассмотрели наиболее распространенные решения, но также есть большое количество других продуктов. Такие решения, как правило, аналогичны рассмотренным, но уступают по ряду параметров.

Гибридные приложения наиболее рациональный выбор для корпоративной разработки.

- возможность использовать имеющихся HTML/CSS/JavaScript специалистов без дополнительного обучения
- можно использовать единый код и компоненты для мобильных приложений и веб-приложений
- нет “жесткой” привязки к используемому решению - например можно легко перенести основной массив HTML/CSS/JavaScript кода между Rhomobile и Cordova/PhoneGap вообще без изменений.
- проблема производительности, особенно UI, теряет актуальность из-за роста производительности аппаратной части и оптимизации WebView.
- в распоряжении разработчика - огромное количество разнообразных JavaScript фреймворков, модулей, готовых решений на любой вкус !



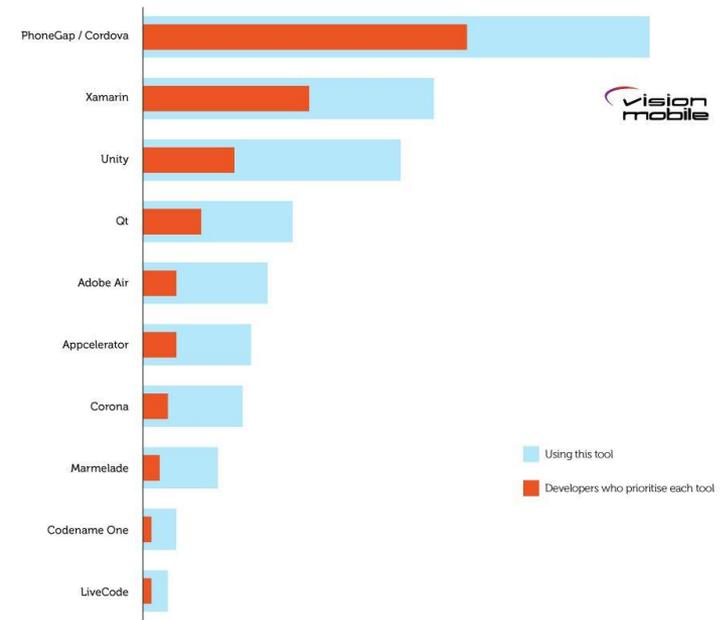
≠



- В настоящее время по отчету Gartner в корпоративной среде 90% приложений это веб приложения или гибридные
- JavaScript - один из самых популярных языков программирования
- Apache Cordova - самое популярное решение для построения гибридных кросс-платформенных платформ. Ведущие фирмы (Intel, IBM, Oracle и другие) предлагают свои решения на базе Cordova
- Стандарт де-факто для браузера и JavaScript VM - WebKit

THE TOP 3 CROSS PLATFORM TOOLS ACCOUNT FOR 70% OF PRIMARY USE

% of developers using cross-platform tools by primary tool and tools used (n=1,664)



Source: Cross-Platform Tools 2015 | vmob.me/CPT15 Copyright VisionMobile | All rights reserved

Проблемы гибридных решений аналогичны проблемам больших web сайтов/порталов.

Если у вас много контента, много данных, то все это “висит” в WebView и производительность падает.

При этом мы вынуждены держать все в WebView - и данные и логику и UI.

WebView однопоточный - мы не можем выполнять код требующий много времени - все “зависнет”.



Решение однозначное - выносить часть приложения из WebView !

Rhobile предлагает такое решение путем воссоздания серверного окружения RoR на локальном устройстве - чтобы разработчикам было привычнее и проще. При этом WebView становится легковесной так как содержит только UI, а данные, логика находятся вне WebView и в отдельных потоках.

2008 Основан успешный стартап Rhomobile

2011 был куплен фирмой Motorola Solutions

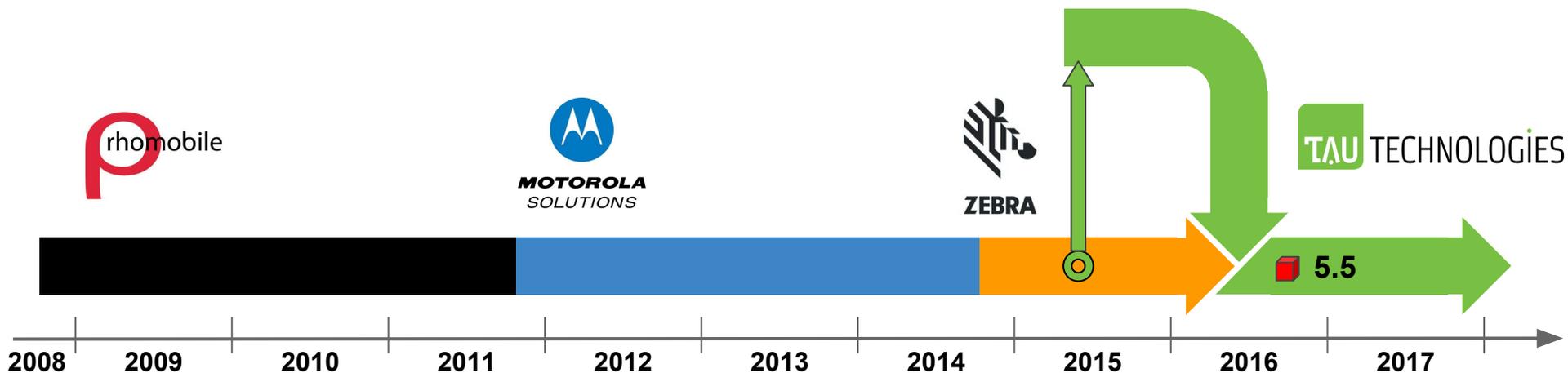
2014 как часть Motorola Solutions' Enterprise вошел в Zebra Technologies

2015 несколько членов команды Rhomobile основали TAU Technologies

2016 Zebra Technologies объявила о переводе решения Rhomobile полностью в open source и о том что TAU Technologies продолжит разработку и поддержку.

Подробнее: [Zebra: RhoMobile Open Source FAQ](#)

Последняя версия решения - **5.5** выпущена TAU Technologies в сентябре 2016.



Подробнее смотрите : <http://tau-technologies.com/>





RHOMOBILE SUITE



RHODES



RHO STUDIO



TAU EXTENSIONS



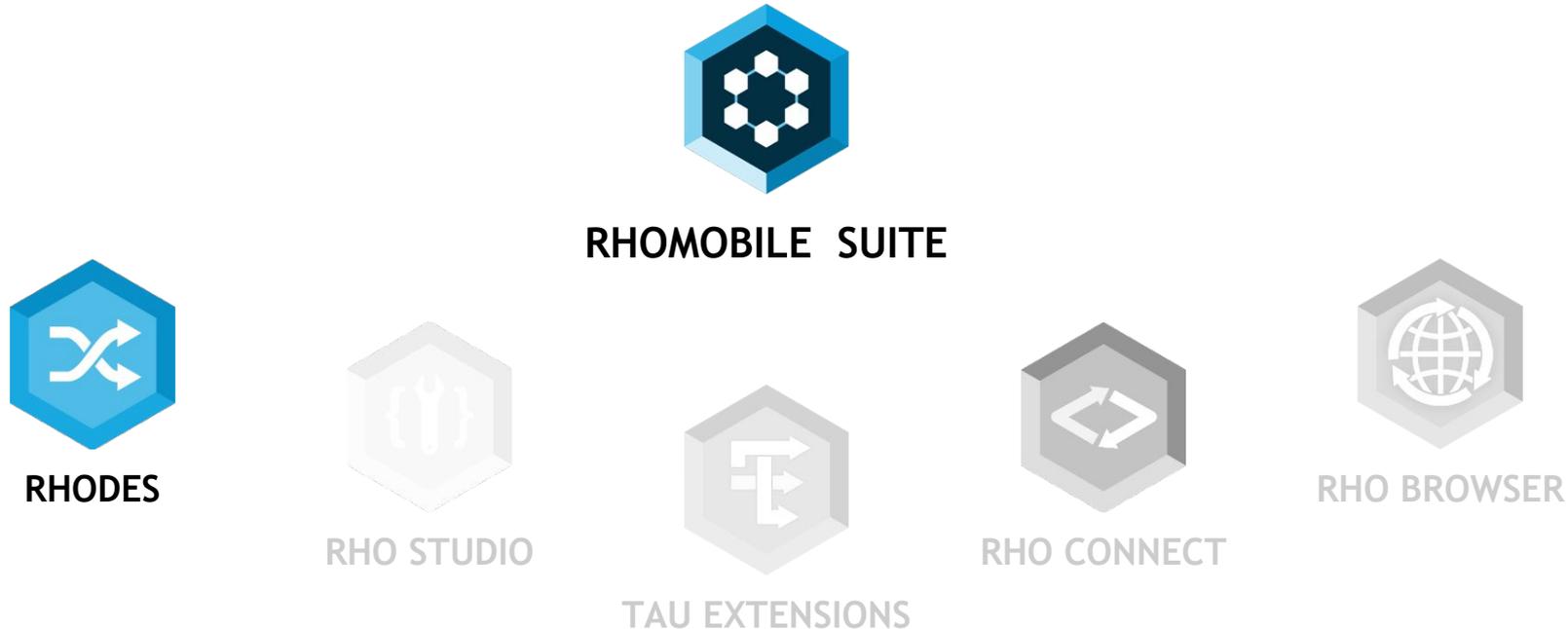
RHO CONNECT



RHO BROWSER

Полностью **open-source** решение для разработки кросс-платформенных гибридных приложений
Включает в себя платформу, IDE, решение для синхронизации данных и специализированный браузер.

Поддерживается **iOS, Android, WinCE/WM, WP**



Платформа для разработки кросс-платформенных гибридных приложений
Возможна как разработка только на **HTML/CSS/JS**, так и с использованием **Ruby**
Включает в себя большое количество модулей, включая поддержку Barcode, Printing и т.п.
Поддерживается **iOS, Android, WinCE/WM, WP**



RHOMOBILE SUITE



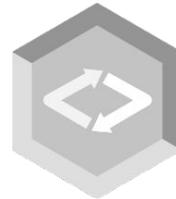
RHODES



RHO STUDIO



TAU EXTENSIONS

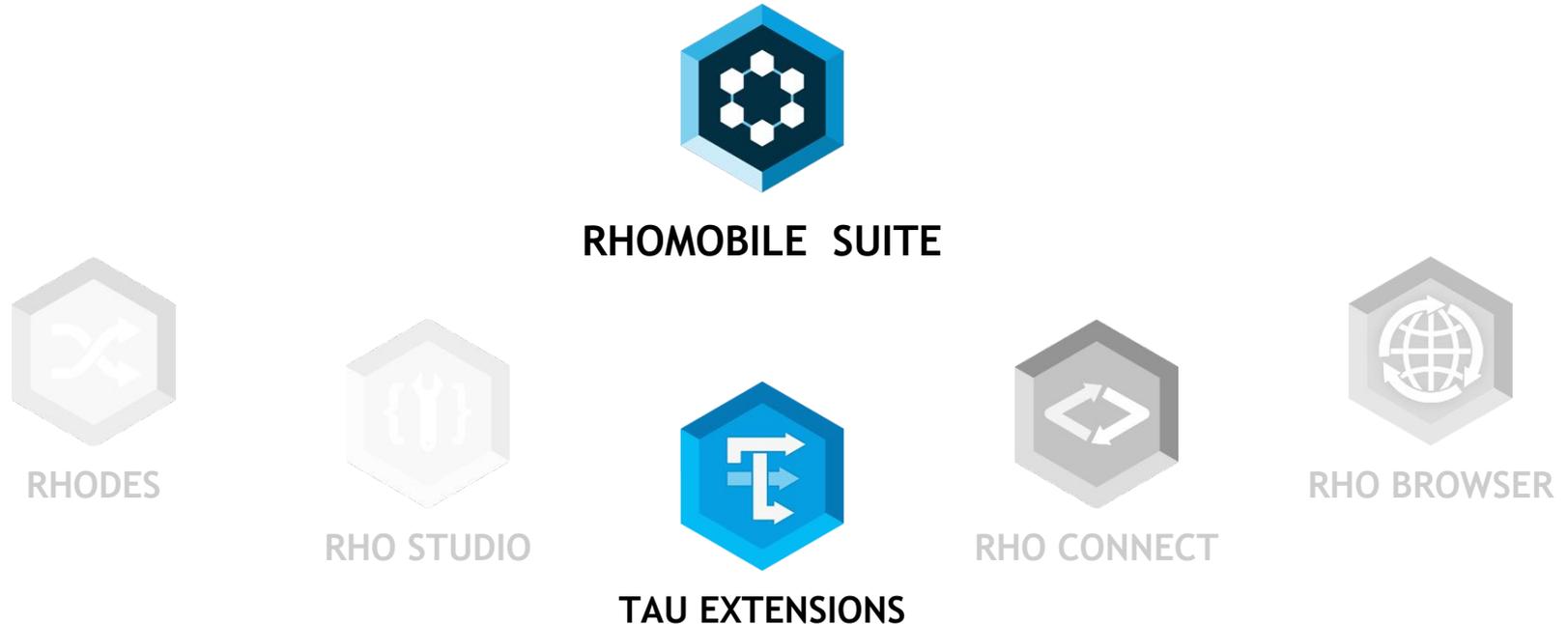


RHO CONNECT



RHO BROWSER

IDE на базе Eclipse для разработки приложений на Rhodes
поддерживается отладка Ruby кода
имеется собственный симулятор



Набор расширений, включает в себя :

- Crosswalk WebView для Android (замена системного в приложении)
- WebKit WebView для WinCE/WM (замена системного IE в приложении)
- улучшенный OpenSSL модуль для Android



Решение для легкой синхронизации данных с удаленным сервером
Локальный клиент интегрирован в Rhodes приложение, но возможно использование и вне Rhodes
Серверная часть реализована на Ruby и поддерживает основные DB (можно расширять)



RHOMOBILE SUITE



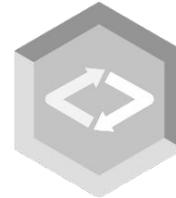
RHODES



RHO STUDIO



TAU EXTENSIONS

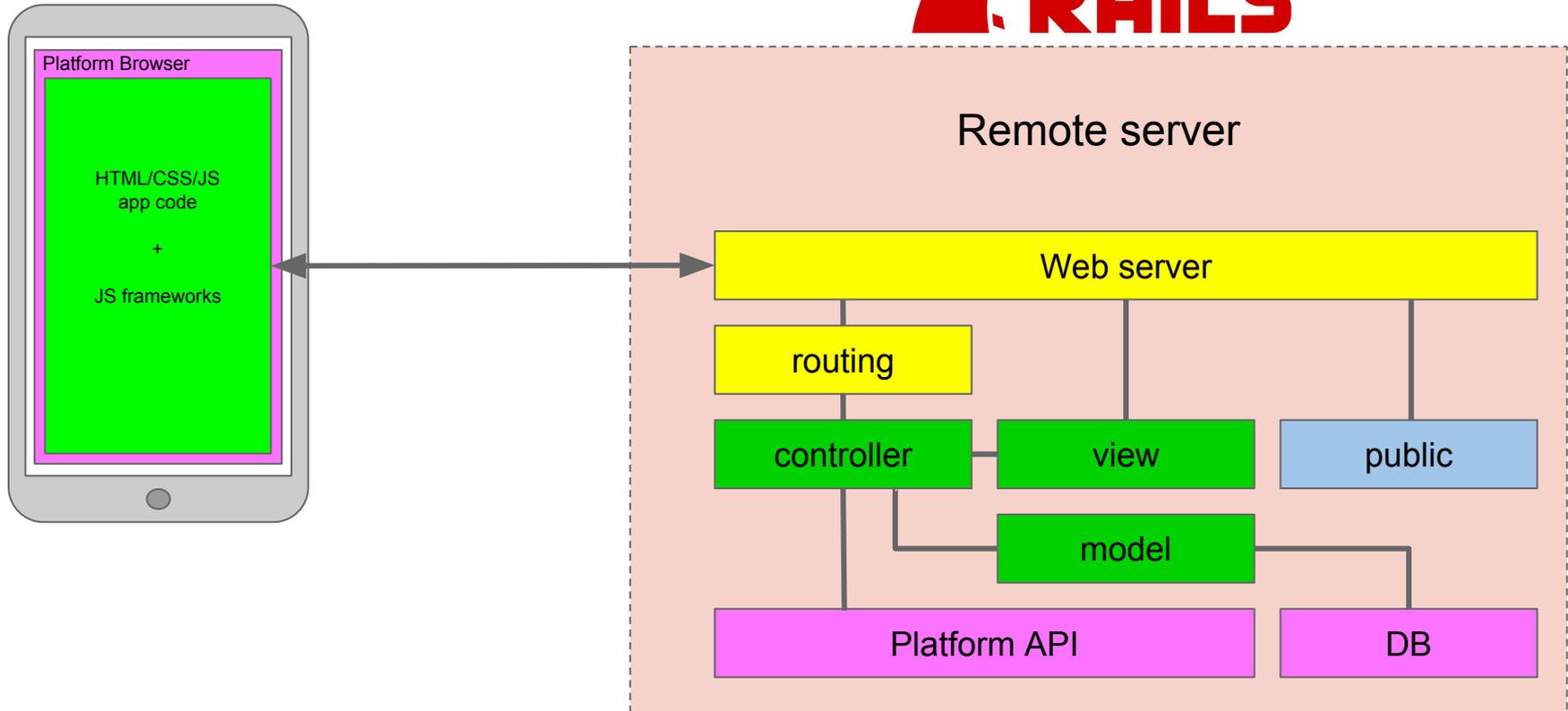


RHO CONNECT



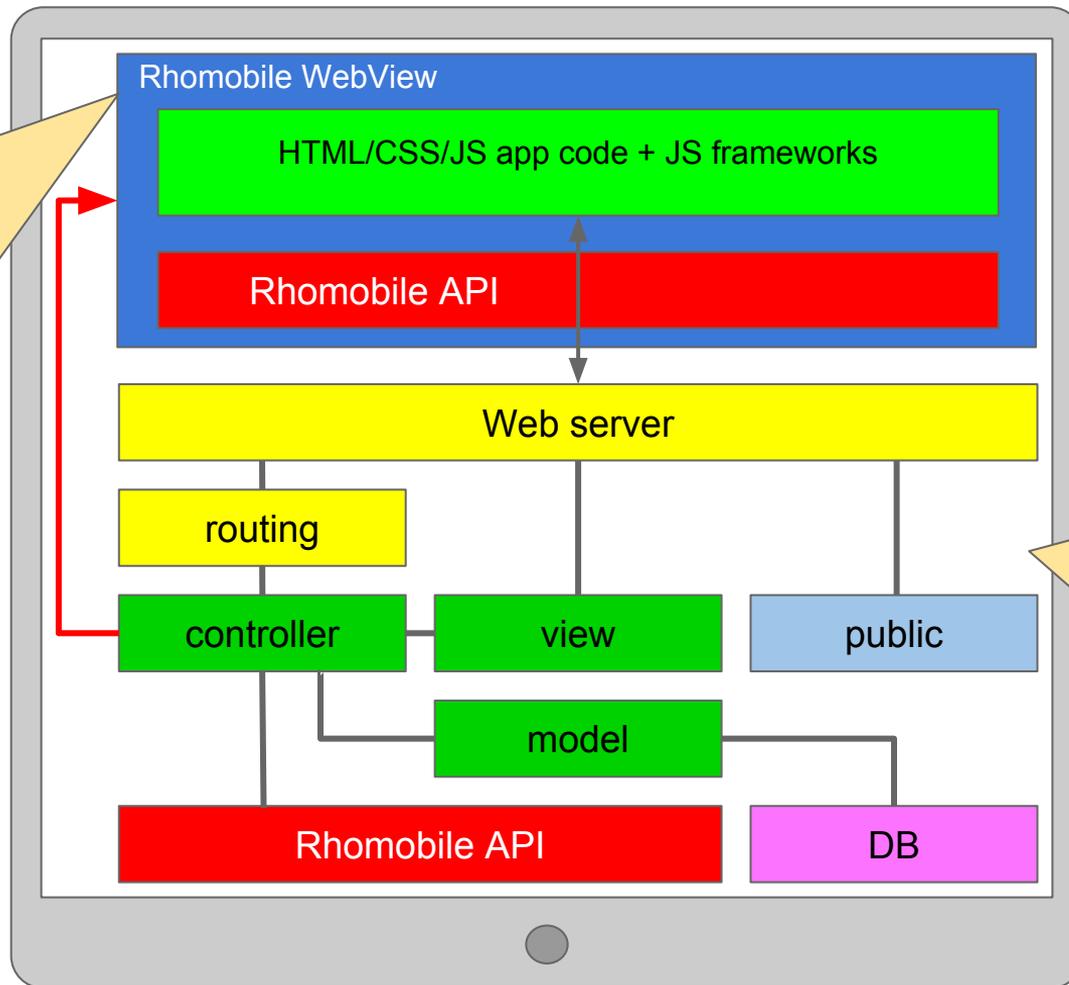
RHO BROWSER

Специализированный браузер с доступом к расширенному Rhomobile API из JavaScript позволяет обходиться без написания приложения вообще - доступ к device capability прямо из JavaScript HTML страницы открытой с сервера (или локально с файловой системы)
На WinCE/WM использует WebKit (поддержка современных версий HTML/CSS/JS в отличие от IE)



Гибридная архитектура. UI реализуется HTML/CSS/JS контентом в WebView

Разработчик приложения может использовать любой JavaScript UI framework по желанию



RHODES

На устройстве мы имеем локальный HTTP сервер с полноценной Ruby VM. Ruby код контроллеров выполняется прямо на устройстве.

Перед установкой пожалуйста настройте вашу систему. Подробнее:

<http://docs.tau-technologies.com/en/5.5/guide/rhobile-install>

<http://docs.tau-technologies.com/en/5.5/guide/nativesdksetup>

Есть три способа установки:

- Скачать и установить с помощью пакета установки.
URL для скачивания : <http://rhobile.tau-technologies.com/index.html>
- Установить основной и вспомогательные GEMs (для начала достаточно основного)



```
$ gem install rhodes
```

- Установить исходный код(в приложении придется вручную задавать путь к платформе) и вручную установить нужные GEMs
Исходный код : <https://github.com/rhobile/rhodes>

Затем необходимо настроить пути к платформенным SDK:

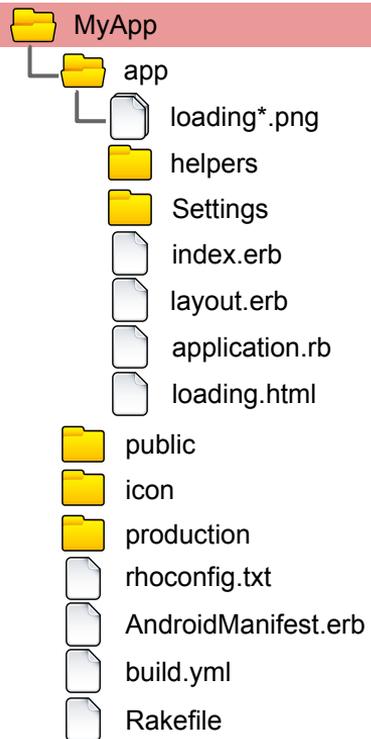
```
$ rhodes-setup
```

Создадим приложение

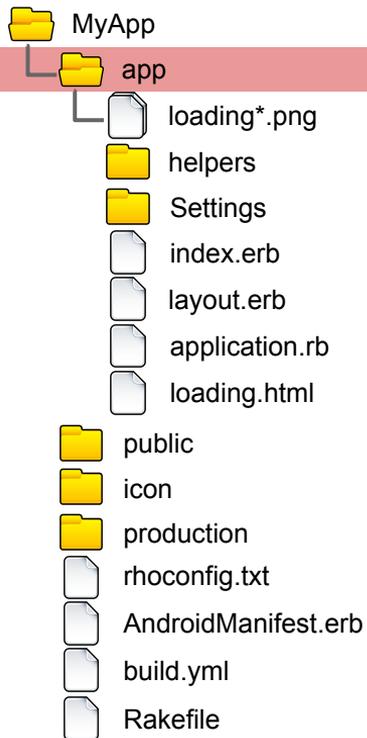
(подробнее http://docs.tau-technologies.com/en/5.5/guide/creating_a_project) :

```
$ rhodes app MyApp
```

rhodes - команда для генерации “болванок”:
приложения, модели, расширения (extension).
Сгенерированный код сразу работоспособен.

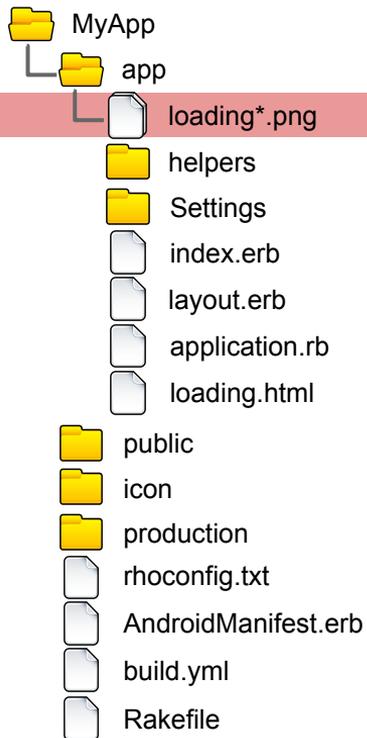


Мы получили папку MyApp внутри которой находится “болванка” Rhodes приложения



папка **app** содержит код приложения - **ruby** файлы
и файлы **view** шаблонов - **erb**

в **run-time** папка располагается в корне
локального **HTTP** сервера

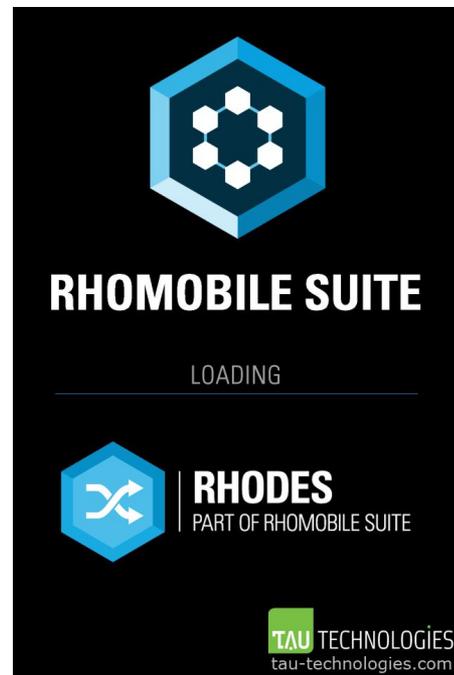


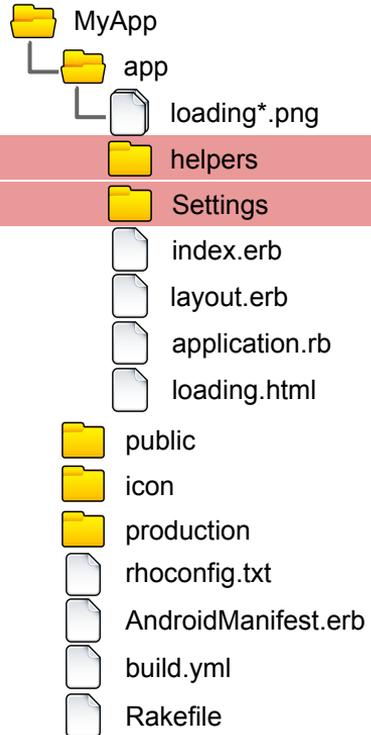
Изображение показываемое при загрузке приложения - набор разных разрешений и размеров

Подробнее:

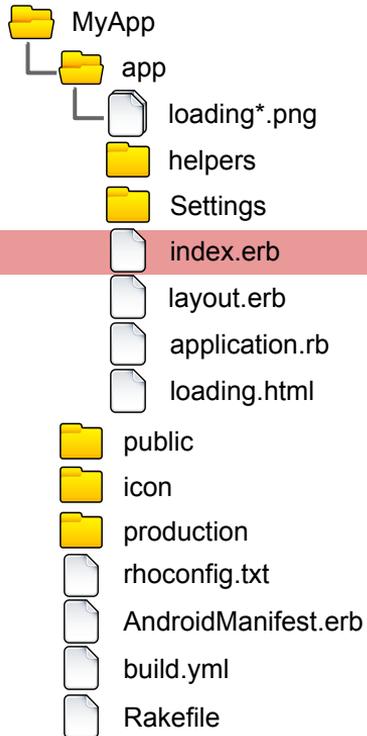
http://docs.tau-technologies.com/en/5.5/guide/app_icon_splash

loading-667h@2x.png
loading-Landscape@2x.png
loading-LandscapeRight.png
loading-Portrait@2x.png
loading-736h@3x.png
loading-LandscapeLeft.png
loading-LandscapeRight@2x.png
loading-PortraitUpsideDown.png
loading.png
loading-568h@2x.png
loading-Landscape.png
loading-LandscapeLeft@2x.png
loading-Portrait.png
loading-PortraitUpsideDown@2x.png
loading@2x.png





вспомогательный **ruby** код



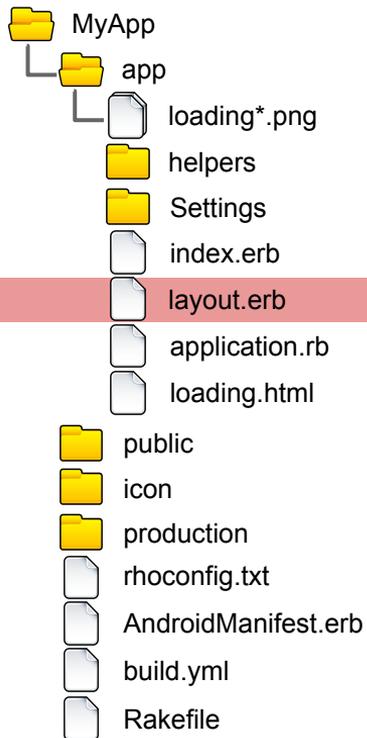
erb шаблон для генерации страницы (по умолчанию - открывается на старте приложения)

Используется двухступенчатая генерация HTML:

обычный шаблон страницы содержит только **content**, а общий код заголовка, загрузки **CSS,JS** в отдельном шаблоне **layout**

```
<div data-role="page" data-url="<%= Rho::Application.startURI %>/">  
  
  <div data-role="content">  
    <ul data-role="listview">  
      <li><a href="#">Add link here...</a></li>  
    </ul>  
  </div>  
  
</div>
```

erb шаблон layout - используется для генерации всех страниц.
Может быть переопределен для каждого контроллера



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>

загрузка CSS и JS ...

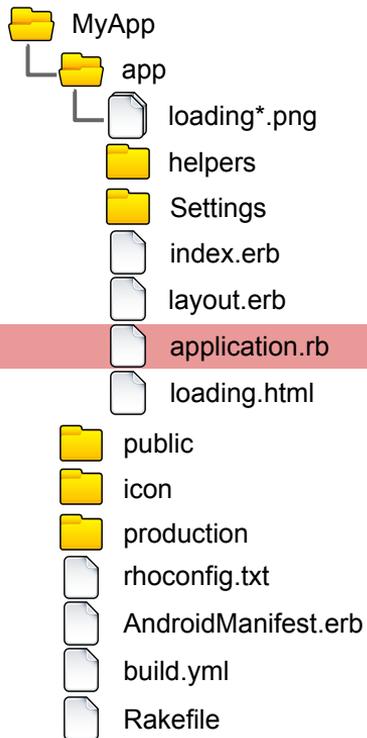
</head>

<body data-platform="<%= Rho::System.getProperty('platform') %>">
    <%= @content %>
</body>

</html>
```

здесь вставляется результат генерации шаблона страницы

application.rb - класс приложения, код инициализации, активации/деактивации



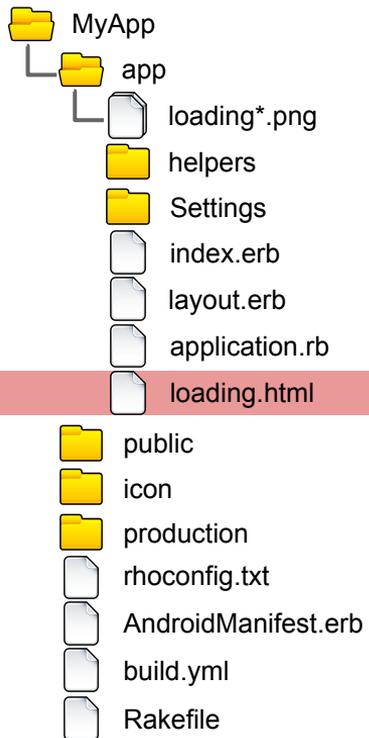
```
require 'rho/rhoapplication'

class AppApplication < Rho::RhoApplication

def initialize
  # Tab items are loaded left->right, @tabs[0] is leftmost tab in the tab-bar
  # Super must be called *after* settings @tabs!
  @tabs = nil
  #To remove default toolbar uncomment next line:
  #@@@toolbar = nil
  super
end

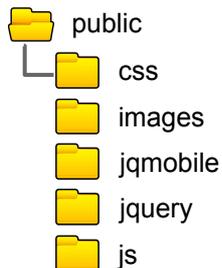
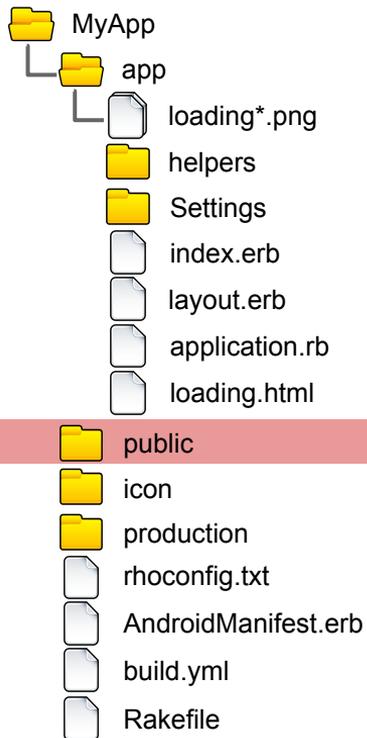
end
```

В отсутствие изображений используется HTML файл показываемый во время загрузки приложения (до показа стартовой страницы)



папка где хранятся файлы **CSS, JS, Images** и т.п.

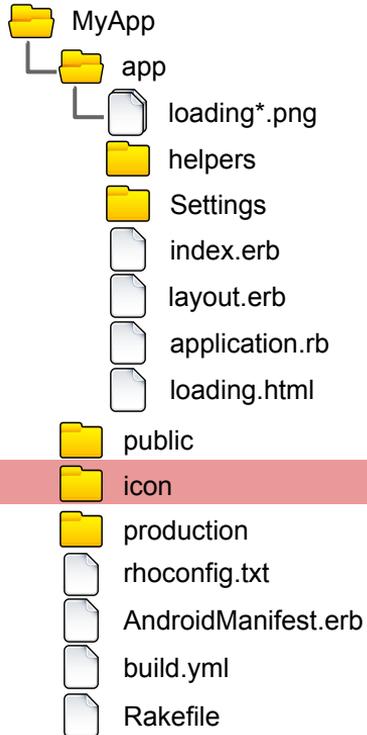
в **run-time** папка располагается в корне локального **HTTP** сервера



Папка содержит набор иконок разного размера

Подробнее:

http://docs.tau-technologies.com/en/5.5/guide/app_icon_splash

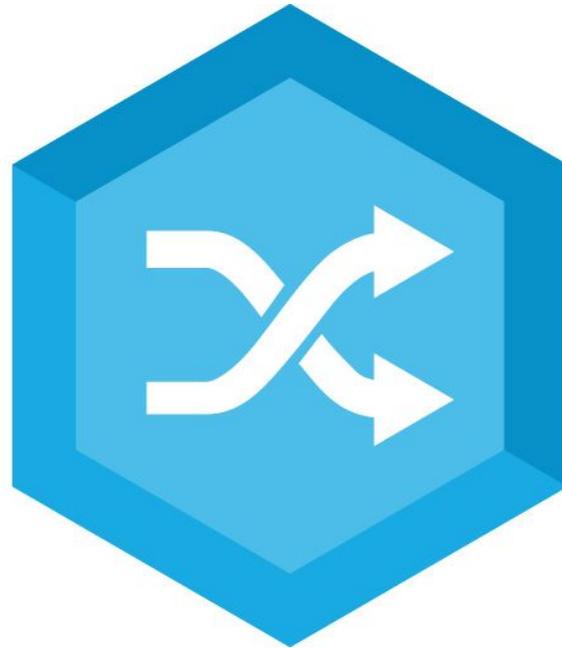
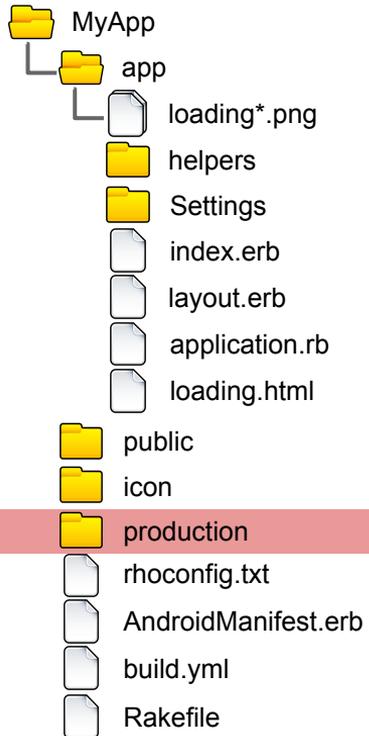


icon.ico
icon.png
icon.svg
icon114.png
icon120.png
icon144.png
icon152.png
icon180.png
icon57.png
icon60.png
icon72.png
icon76.png

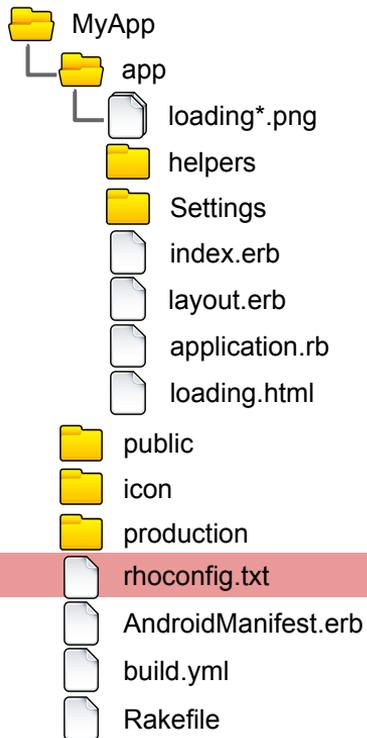


Папка содержит изображение большой иконки для iTunes
Подробнее: http://docs.tau-technologies.com/en/5.5/guide/build_ios

iTunesArtwork.png
iTunesArtwork@2x.png



rhoconfig.txt - файл содержит различные настройки, используемые во время работы приложения.



```
# startup page for your application
start_path = '/app'

options_path = '/app/Settings'

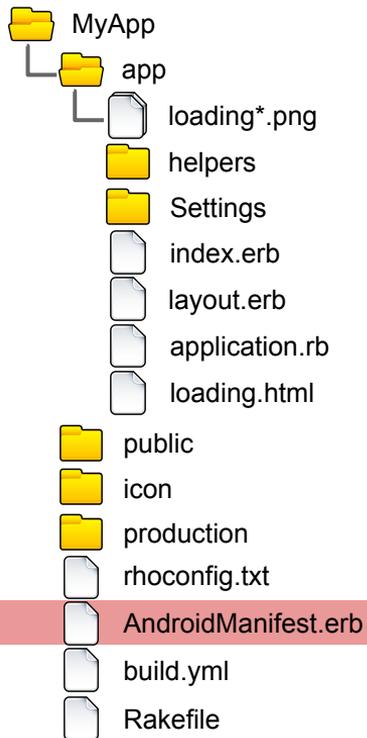
# Rhodes log properties
MinSeverity = 1
LogToOutput = 1
MaxLogFileSize=50000
logserver = 'http://rhologs.heroku.com'
logname='MyApp'

syncserver = "
sync_poll_interval=0

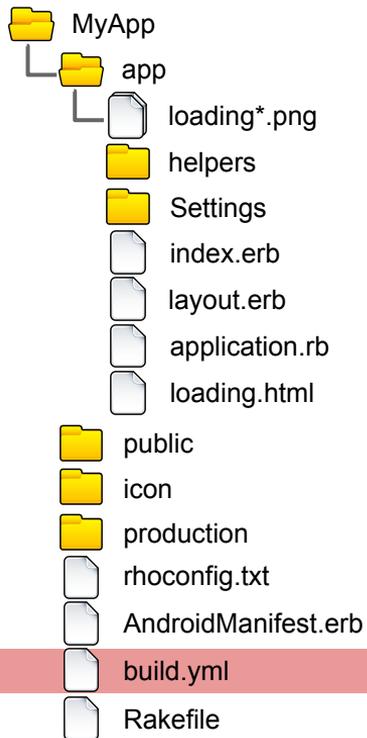
...
```

AndroidManifest.erb - шаблон для генерации Manifest для Android платформы - можно редактировать для дополнительной настройки.

Подробнее; http://docs.tau-technologies.com/en/5.5/guide/build_android



build.yml - файл содержит различные настройки приложения, необходимые для сборки или генерации платформенных файлов.



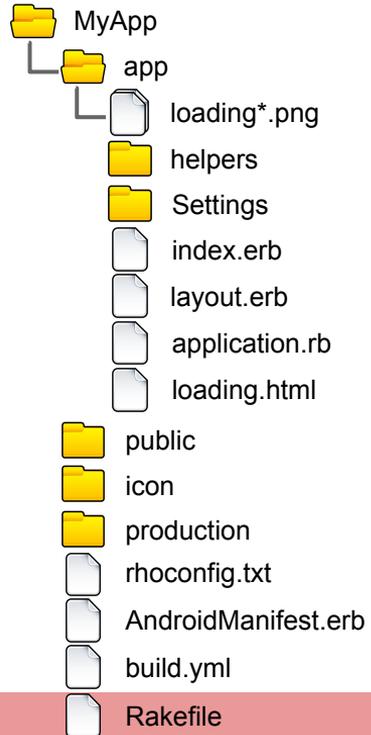
```
sdkversion: 5.5.0
name: MyApp
version: 1.0
vendor: rhomobile
build: debug
applog: rholog.txt
iphone:
  configuration: Release
  sdk: iphonesimulator10.0
  BundleIdentifier: com.rhomobile.myapp
  BundleURLScheme: myapp
  production:
    ipa_itunesartwork_image: "./production/iTunesArtwork.png"

wp8:
  productid: 'a48b8410-70dc-0134-ab82-482a1411841f'

android:
  version: 4.1.0

extensions: []
```

стандартный скрипт для работы **rake** команд

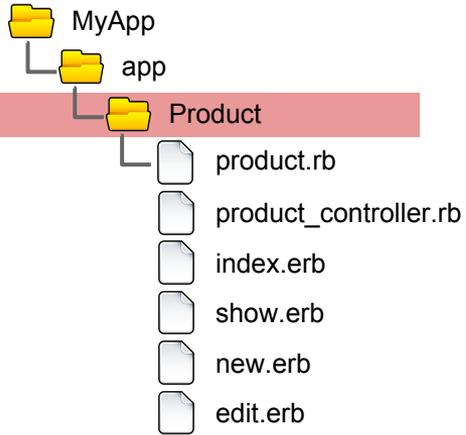


Создадим новую модель - кроме ruby файла с описанием модели, генератор сразу добавит контроллер и набор view для работы с моделью. (подробнее http://docs.tau-technologies.com/en/5.5/guide/rhom_ruby)

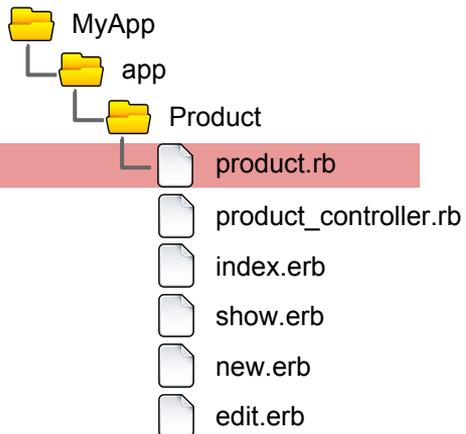
Для этого вызовем генератор из папки приложения :

```
$ rhodes model Product name,brand,price
```

в папку **app** добавилась папка **Product** с набором файлов



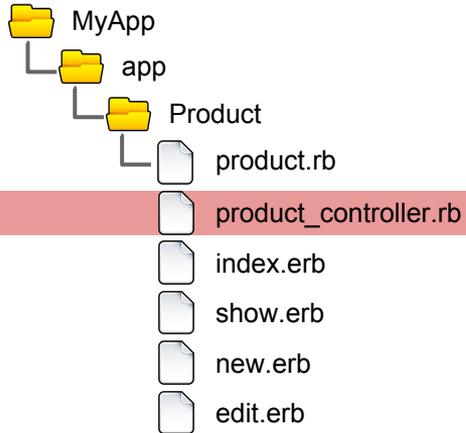
product.rb - определение класса модели. По умолчанию используется Property Bag схема.



```
# The model has already been created by the framework, and extends Rhom::RhomObject
# You can add more methods here
class Product
  include Rhom::PropertyBag

  # Uncomment the following line to enable sync with Product.
  # enable :sync

  #add model specific code here
end
```



product_controller.rb - определение класса контроллера. Методы контроллера вызываются из **WebView** по **URL** вида **/app/Controller/method** и возвращают **HTML** страницу (по умолчанию генерируемую из erb шаблона с именем **method.erb**). Если метод отсутствует то выполняется простая генерация из шаблона **erb**.

```
require 'rho/rhocontroller'
require 'helpers/browser_helper'

class ProductController < Rho::RhoController
  include BrowserHelper

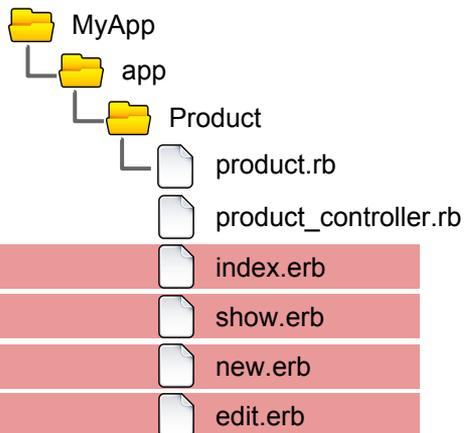
  def index
    @products = Product.find(:all)
    render :back => '/app'
  end

  def edit
    @product = Product.find(@params['id'])
    if @product
      render :action => :edit, :back => url_for(:action => :index)
    else
      redirect :action => :index
    end
  end
end
```

Получение всех объектов модели **Product**

@params - hash содержащий все параметры из query запроса

...



набор **view - erb** шаблонов для генерации страниц для соответствующих действий над моделью.

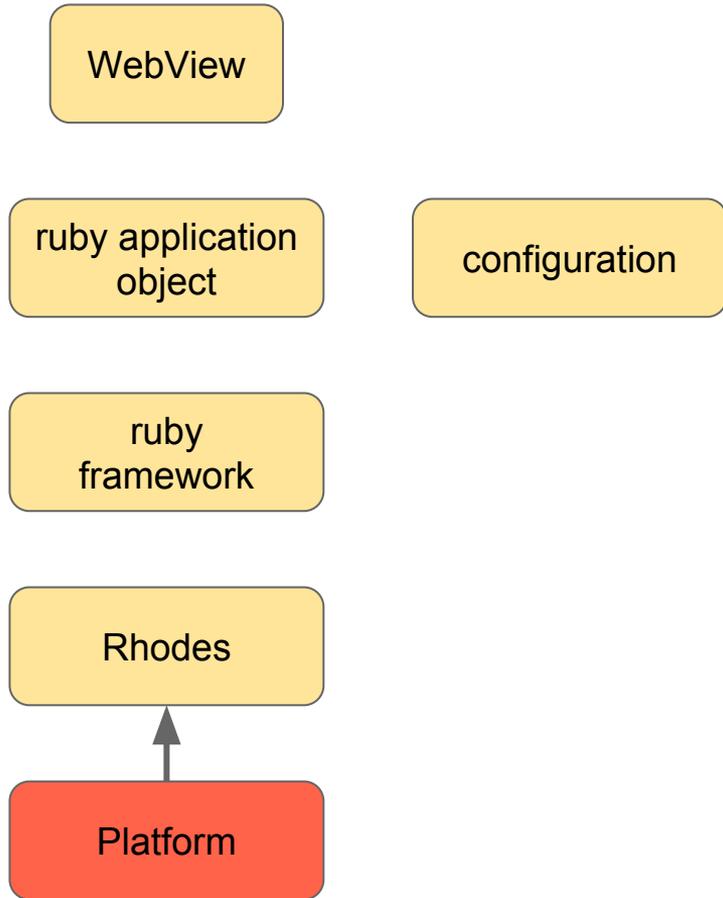
index.erb - страница по умолчанию. метод контроллера **index** тоже вызывается по умолчанию, то есть когда локальный сервер отвечает на запрос вида **/app/Controller/**

```
<div data-role="page">
  <div data-role="header" data-position="inline">
  ...
  </div>

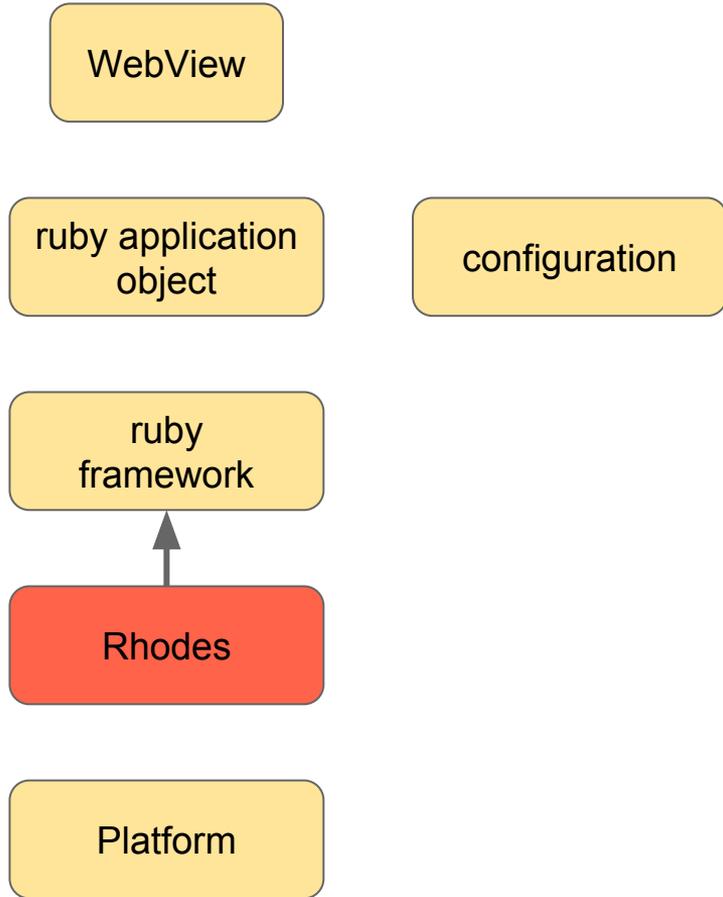
  <div data-role="content">
  <ul data-role="listview">
    <% @products.each do |product| %>
      <li>
        <a href="<%= url_for :action => :show, :id => product.object %>">
          <%= product.name %>
        </a>
      </li>
    <% end %>
  </ul>
</div>
</div>
```

перебор всех объектов модели и создание для каждого элемента списка со ссылкой на URL для показа объекта

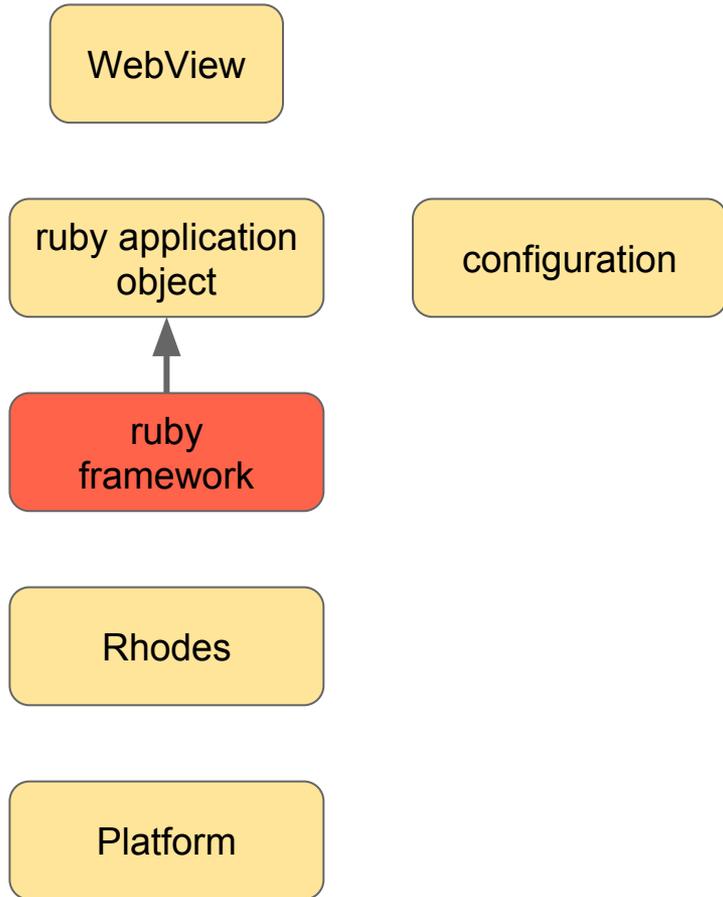
index.erb



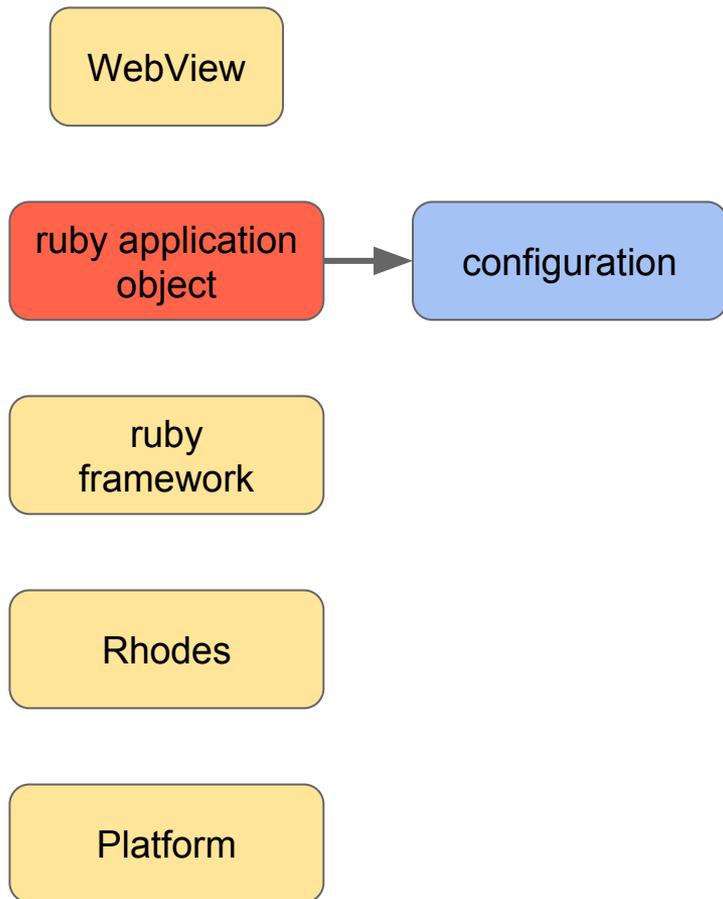
инициализация приложения на уровне платформы



инициализация Ruby фреймворка
(и HTTP сервера) и
вызов старта Ruby фреймворка



инициализация фреймворка и вызов
инициализации объекта приложения



инициализации объекта приложения где можно создать тулбар, таббар, вызвать инициализацию нужных объектов и т.п.

из конфигурации читается стартовый **URL**

application.rb

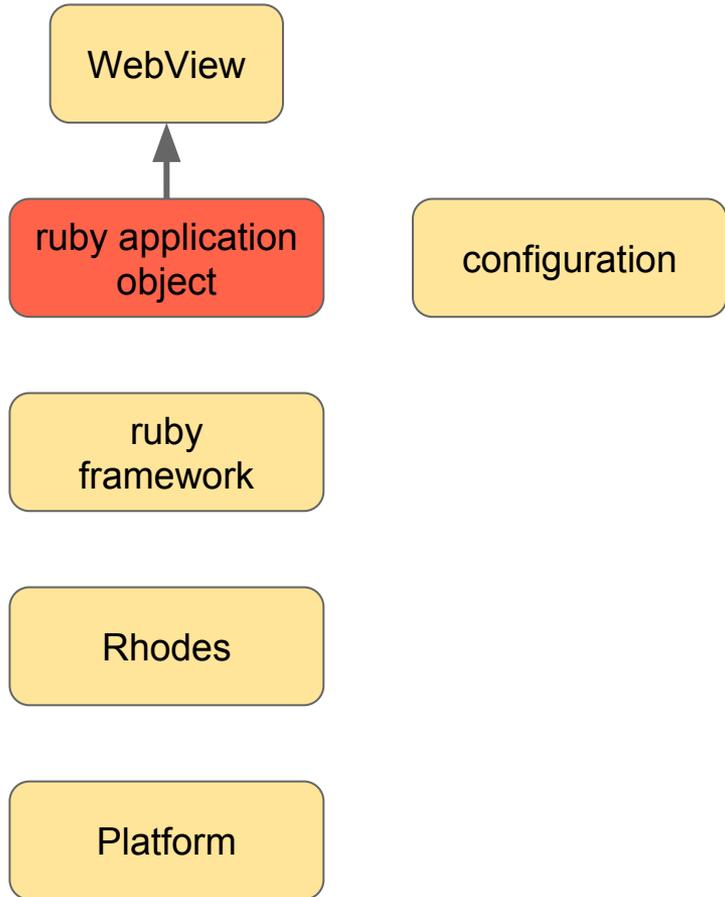
```
require 'rho/rhoapplication'

class AppApplication < Rho::RhoApplication

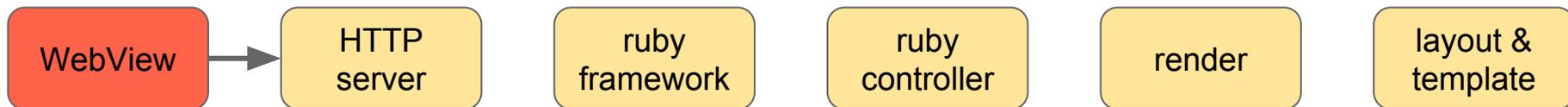
  def initialize
    # Tab items are loaded left->right, @tabs[0] is leftmost tab in the tab-bar
    # Super must be called *after* settings @tabs!
    @tabs = nil
    #To remove default toolbar uncomment next line:
    #@@toolbar = nil
    super
  end

end

end
```



открытие стартового URL в WebView



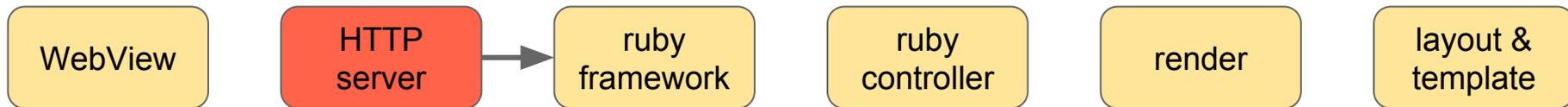
запрос локальному серверу (при навигации в **WebView** или **Ajax** запросе)

Рассмотрим подробнее обработку запроса [/app/Product/](#)

В Модели при этом имеются два объекта

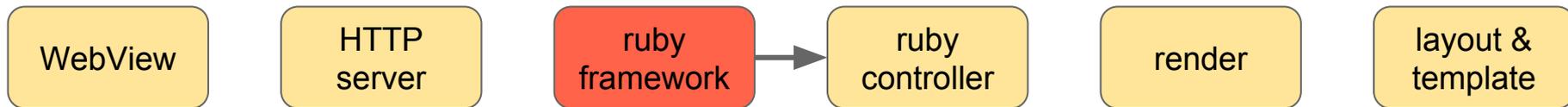
Также модифицируем наше приложение чтобы этот URL открывался сразу на старте, для этого поменяем в **rhoconfig.txt** параметр **start_path** :

```
# startup page for your application  
start_path = '/app/Product'
```



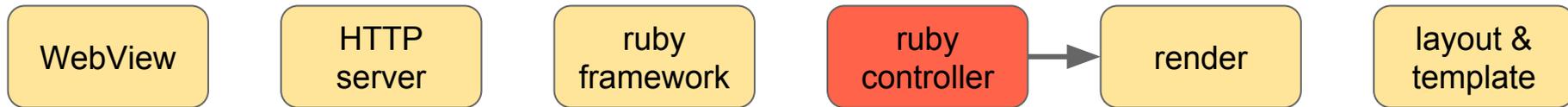
Сервер определяет файл это или руби контроллер
если файл то просто отдает
если руби контроллер то вызывает контроллер передавая **action** и **query**

Сервер получает запрос **/app/Product/**
В нашем случае это запрос контроллеру



Идет обработка запроса `/app/Product/`
В нашем случае это запрос контроллеру

Создается объект класса **ProductController** и у него вызывается метод **index** для генерации **HTML**, который будет возвращен как ответ на запрос серверу



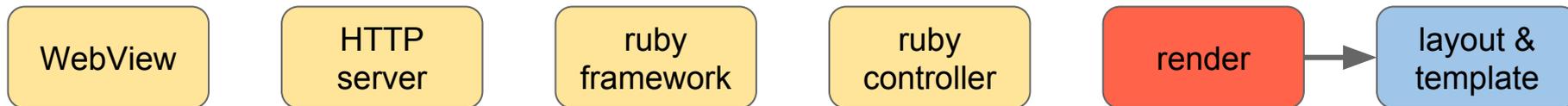
```
require 'rho/rhocontroller'
require 'helpers/browser_helper'

class ProductController < Rho::RhoController
  include BrowserHelper

  def index
    @products = Product.find(:all)
    render :back => '/app'
  end

  def edit
    @product = Product.find(@params['id'])
    if @product
      render :action => :edit, :back => url_for(:action => :index)
    else
      redirect :action => :index
    end
  end
end

...
```



/app/layout.erb

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
  загрузка CSS и JS ...
</head>

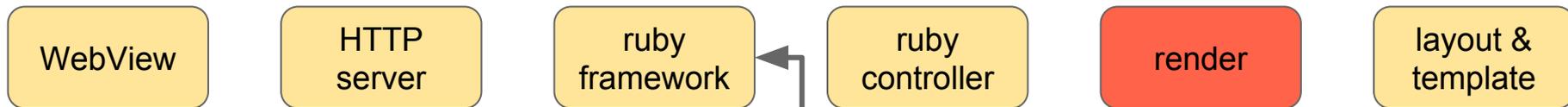
<body data-platform="<%= Rho::System.getProperty('platform')
%>">
  <%= @content %>
</body>

</html>
```

/app/Product/index.erb

```
<div data-role="page">
  <div data-role="header" data-position="inline">
  ...
  </div>

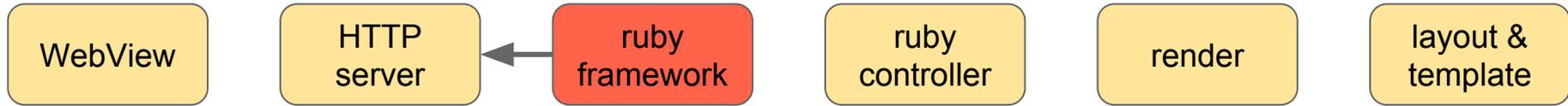
  <div data-role="content">
    <ul data-role="listview">
      <% @products.each do |product| %>
        <li>
          <a href="<%= url_for :action => :show, :id =>
product.object %>">
            <%= product.name %>
          </a>
        </li>
      <% end %>
    </ul>
  </div>
</div>
```



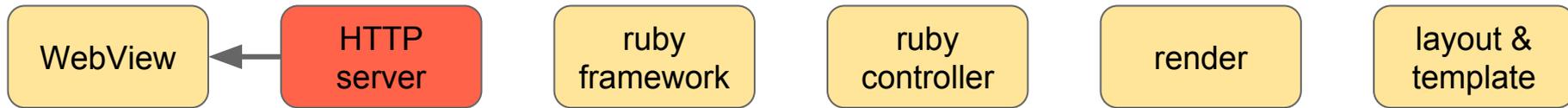
```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
...
</head>
<body data-platform="APPLE">
  <div data-role="page">
    <div data-role="header" data-position="inline">
      ...
    </div>
    <div data-role="content">
      <ul data-role="listview">
        <li>
          <a href="/app/Product/{245337759802112.9}/show">
            item1
          </a>
        </li>
        <li>
          <a href="/app/Product/{245337759802114.9}/show">
            Item2
          </a>
        </li>
      </ul>
    </div>
  </div>
</body>
</html>
```

URL страницы показа объекта модели.
245337759802112.9 - id объекта модели

в теле метода **id** будет одним из параметров и его можно получить так :
@params['id']



возврат результата серверу

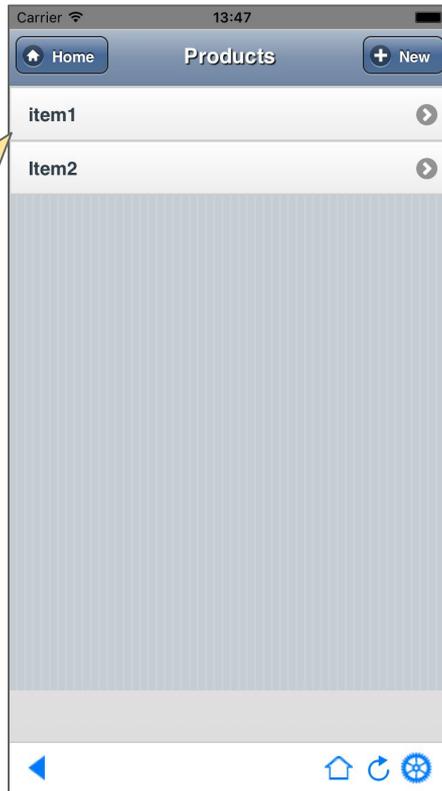


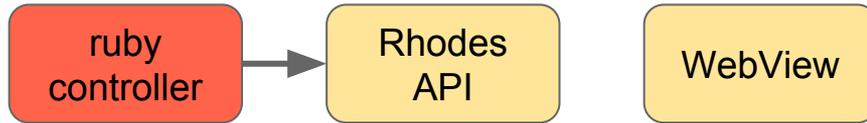
Гибридная архитектура.
UI реализуется HTML/CSS/JS
контентом в WebView

То есть этот UI это не
системные UI компоненты, а
содержимое WebView.

В данном случае используется
jQuery Mobile для создания UI
компонентов в WebView

Разработчик может
использовать любой JavaScript
UI фреймворк по своему
желанию





```
require 'rho/rhocontroller'  
require 'helpers/browser_helper'
```

```
class ProductController < Rho::RhoController  
  include BrowserHelper
```

```
  def index  
    @products = Product.find(:all)  
    render :back => 'app'  
  end
```

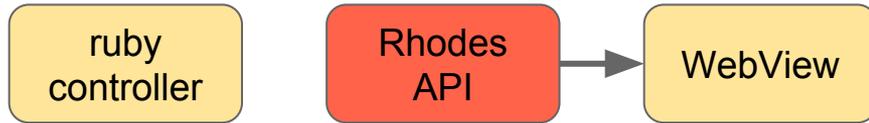
```
  def do_navigate  
    Rho::WebView.navigate('https://google.com')  
  end
```

```
  def do_execute_js  
    Rho::WebView.executeJavascript('alert("test message!");')  
    redirect :action => :index  
  end
```

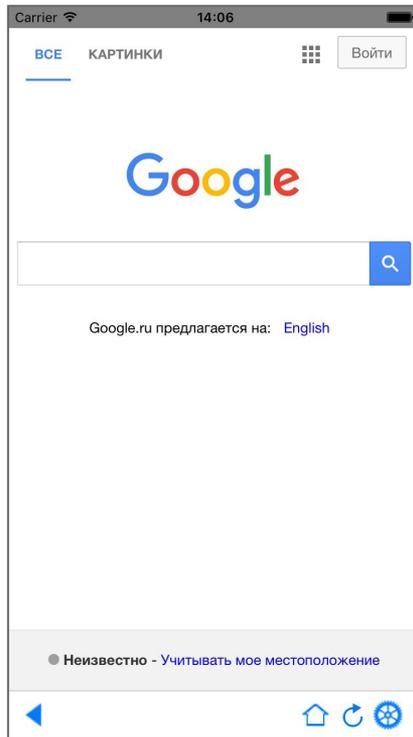
```
  ...
```

URL для вызова
`/app/Product/do_navigate`

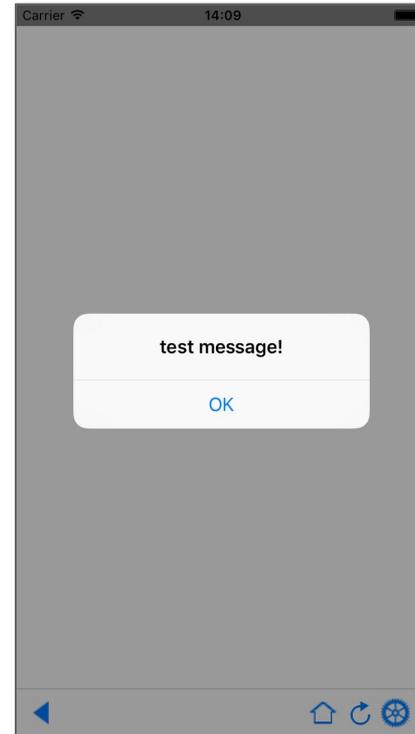
URL для вызова
`/app/Product/do_execute_js`



/app/Product/do_navigate



/app/Product/do_execute_js



А можно не использовать Ruby ?

Можно !

В этом случае платформа становится архитектурно полностью аналогичной Cordova/PhoneGap.

Чтобы полностью выключить **ruby** функционал нужно :

1. добавить в **build.yml** параметр: **javascript_application: true**
2. разместить в папке **public HTML/CSS/JS** контент и установить стартовый **URL** на один из **HTML** файлов.

Подробнее: http://docs.tau-technologies.com/en/5.5/guide/api_js

Запускаем наше приложение на iPhone Simulator:

```
$ rake run:iphone
```

Кроме того, можно генерировать XCode проект, чтобы открыть его в XCode и собирать/запускать приложение на симуляторе и устройстве.

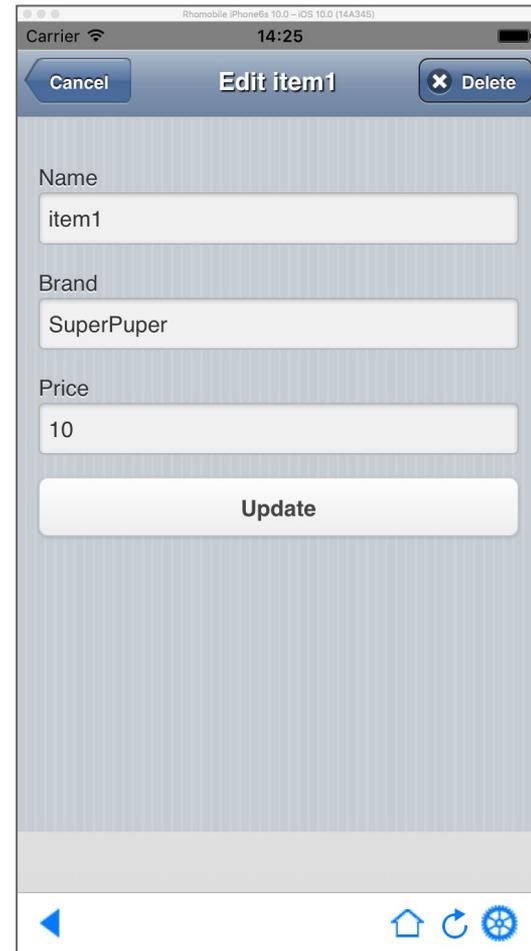
Генерация XCode проекта :

```
$ rake rake build:iphone:setup_xcode_project
```

Созданный проект располагается:



Подробнее: http://docs.tau-technologies.com/en/5.5/guide/build_ios



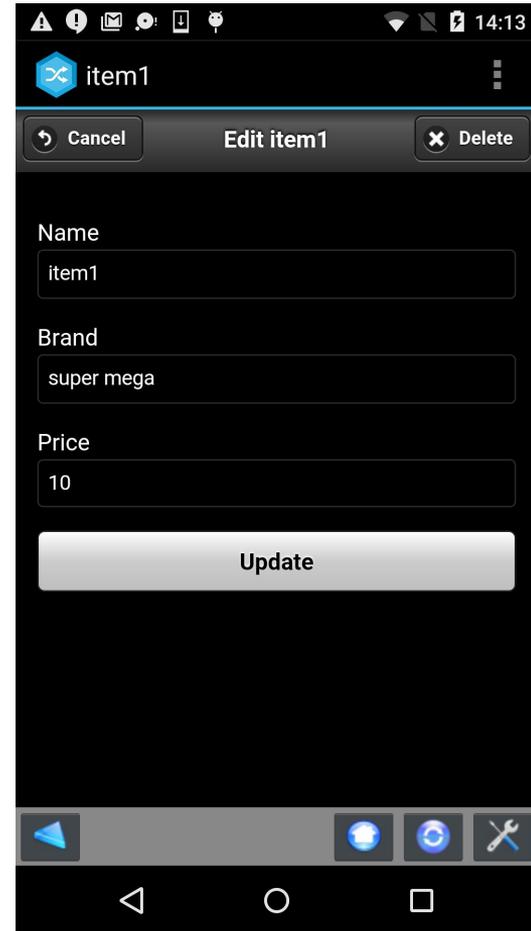
Запускаем наше приложение на Android Simulator:

```
$ rake run:android
```

Запускаем приложение на подключенном устройстве :

```
$ rake run:android:device
```

Подробнее: http://docs.tau-technologies.com/en/5.5/guide/build_android



Список стандартных модулей :

<http://docs.tau-technologies.com/en/5.5/guide/apisummary>

Работа с RestAPI удаленного сервера из Ruby и JS кода в Rhodes:

http://docs.tau-technologies.com/en/5.5/guide/web_services

Как использовать ruby расширения/gem в приложении Rhodes:

http://docs.tau-technologies.com/en/5.5/guide/ruby_extensions

Как расширять функционал создавая новые нативные расширения Rhodes:

http://docs.tau-technologies.com/en/5.5/guide/native_extensions

Как легко синхронизировать данные с удаленным сервером ?

<http://docs.tau-technologies.com/en/5.5/guide/synchronization>

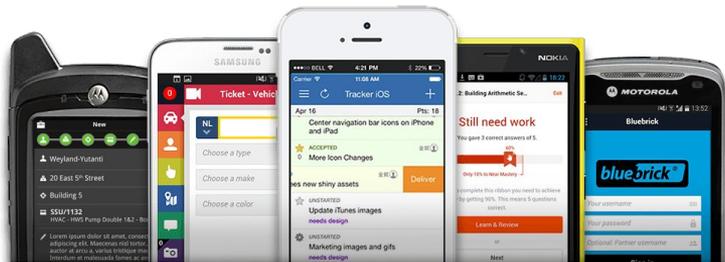
Как автоматически менять контент на устройствах без пересборки приложения ?

<http://docs.tau-technologies.com/en/5.5/guide/live-update>

Где можно посмотреть примеры использования Rhodes API ?

<https://github.com/tauplatform/rhodes-system-api-samples>

<https://github.com/tauplatform/kitchensink>



16:05 58% Last Sync: 22-Sep 4:38pm ✓

Start My Day ▶

View Work List 🔧

0 For Today

My Work Summary 🕒

View Work History 📅

FAQ ?

Settings ⚙️

16:05 58% Last Sync: 22-Sep 4:38pm ✓

◀ Back #11903

Complete

Details

Please provide locker for desk#60 at L31.

Work Type: SR

▶ Contact: [Secure Value_Secure Value](#)

Schedule: 25-Jul 11:30am - 12:30pm

Duration: 01:00

Location

Marina Bay Financial Cntr (L31)

Standard Chartered Bank Singapore

Marina Bay Financial Centre
8 Marina Boulevard Tower 1
#01-01
SINGAPORE
018981

Job Plan

📄 🛒 📝

16:06 58%

Device Name

ANDROID

System ID

USER

Register

1640422 Options

21 Details Forms Parts Reports

Active

Call	
Reference	1640422 - KT6 7EH
Call for	Client 228
Call from	JTM
Booked	21 September 2016 04:34 PM
Appointment	21 September 2016
Slot Times	10:00 AM - 10:40 AM
Slot	Morning AM Call (40 mins)
Notes	Client Ref/Id: 1640422 / 228
Customer Fault	New Service Call This is a service call to

<http://tau-technologies.com>

