

Кроссплатформенная разработка для российских операционных систем

Обзор решений кроссплатформенной разработки для российских и зарубежных ОС.

Кроссплатформенная разработка с Romobile для мобильных и настольных устройств на iOS, Android, WinCE/WM, Аврора/Sailfish, ALT Linux, Astra Linux, ROSA Linux, РЕД ОС, Linux, Windows, Mac OS

Кроссплатформенная разработка

Кроссплатформенная разработка это способ создания приложений когда вы разрабатываете приложение сразу для нескольких платформ и при этом частично или полностью ваш код один сразу для всех целевых платформ.

Плюсы:

- снижение стоимости разработки
- повышение надежности - особенно касается сложной бизнес-логики, ведь легче отлаживать один код, а не несколько реализаций на разных языках.
- нет необходимости в специалистах для каждой необходимой платформы и нет необходимости поддерживать новые требования в каждой платформе (особенно актуально для мобильных, где много изменений от версии к версии).

Минусы:

- как правило, нет доступа ко всем возможностям каждой платформы, но как правило можно написать "расширение" на каждой платформе (подобный код пришлось бы писать и при обычной разработке)
- пониженное быстродействие, особенно в пользовательском интерфейсе. Сейчас не столь актуально, так как производительность компьютеров и мобильных устройств избыточна для подобных задач.
- не "родной" вид приложения - не во всех решениях. Особенно актуально для мобильных платформ. В гибридных решается использованием javascript фреймворков, которые обеспечивают нужный вид и поведение интерфейса (активно используется в обычной веб разработке)

Итоги:

Разумеется имеющиеся кроссплатформенные решения на рынке не “серебряная пуля” и не решают все возможные задачи. И если вам нужно, например, разрабатывать игру с богатой графикой, то ни одно из рассматриваемых решений скорее всего не подойдет, хотя в этом сегменте есть свои кроссплатформенные решения, которые доминируют, то есть большинство продуктов разрабатывается именно на кроссплатформенных решениях. В игровом сегменте это например - Unity. Мы же сосредоточимся на решениях которые наиболее актуальны и используются прежде всего в корпоративном и государственном секторах. Подобные продукты идеально подходят именно для создания всевозможных АРМов(автоматизированное рабочее место) и тп.

Архитектуры решений

Рассмотрим какие бывают архитектуры разработки приложений. Все архитектуры сведены на Рисунке 1.

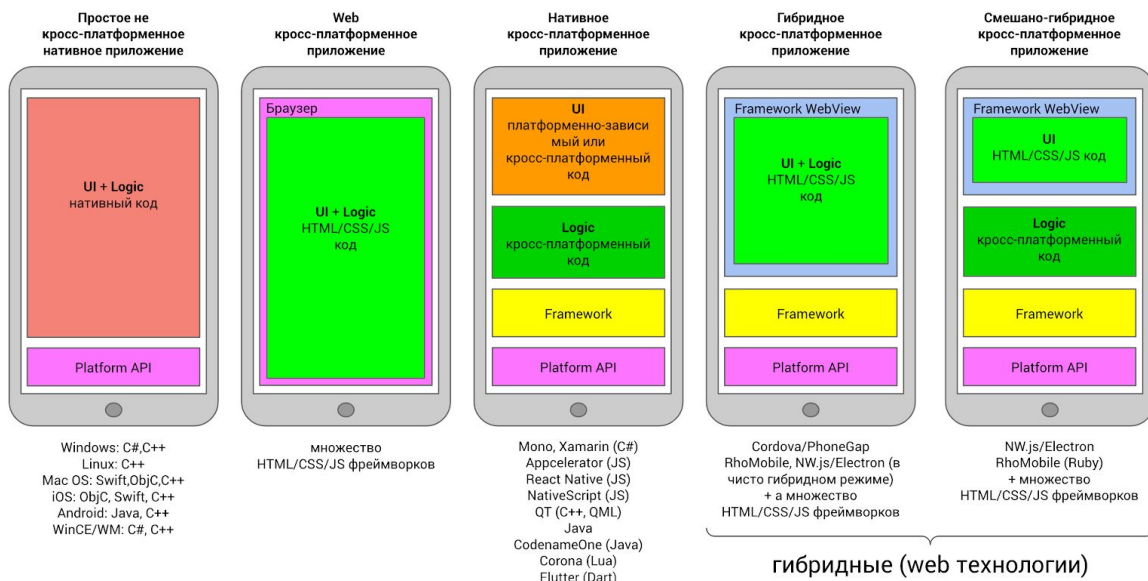


Рисунок 1. Архитектуры приложений.

Нативное (не кроссплатформенное) приложение

Это обычное приложение. Пишется на языке/языках и инструментах, предлагаемых для разработки у конкретной платформы (каждая ОС имеет SDK, компилятор, IDE и прочие инструменты). Для создания пользовательского интерфейса используется то API и объекты, которые предлагаются ОС. Чтобы такое приложение запустить на иной ОС обычно нужно полностью его переделывать на других языках/средствах разработки. Исключения возможны только для очень близких ОС - скажем, перенос между ОС, основывающихся на Linux, может потребовать лишь незначительных модификаций.

Кроссплатформенное веб-приложение

Не совсем нативное приложение - оно запускается в браузере, но по сути это приложение для запуска в определенном контейнере, поэтому устоялось название - веб приложение. Нужно отметить что сейчас это наверное вообще самый используемый тип приложений - пользователь из браузера обращается к удаленному серверу, с которого загружается как правило кроссплатформенный контент (HTML, Javascript, CSS). Кроссплатформенный в том смысле, что он работает в разных браузерах на разных ОС, что в настоящее время все проще получить, ведь стандартизация браузеров усиливается, то есть все современные ОС имеют более-менее современный браузер с поддержкой самых свежих API, в крайнем случае отставание незначительное (например у "Аврора" ОС). Приложением же его можно назвать потому, что в настоящее время это не просто статический контент а также код на javascript, который выполняет часть задач автономно (без обращения к удаленному серверу).

Подводя итоги - такой тип приложений это сейчас самый используемый, особенно в корпоративной сфере и соответственно на рынке присутствует большое число специалистов именно в веб технологиях, а javascript один из самых используемых языков программирования (по числу разработчиков на первом месте среди всех языков программирования), что породило большое количество решений иного типа (и для разработки приложений и для разработки серверного кода и тп), но базирующихся именно на javascript, как одного из самых широко используемых и знакомых языков.

нативное кроссплатформенное приложение

При разработке данного типа приложений используемое решение предлагает создавать приложение сразу для нескольких платформ, но с использованием одного языка/технологии. Логика и работа с данными реализуется единым кодом для всех ОС сразу. Пользовательский интерфейс же может быть как отдельным для каждой ОС, так и одним (построенном на одной библиотеке компонентов, одинаково работающих на всех платформах). Собственно, почти для каждого распространенного языка программирования есть кроссплатформенные решения, которое позволяет создавать приложения на одном языке для разных ОС.

гибридное кроссплатформенное приложение

Это приложение построенное с использованием веб технологий, но при этом оформленное в виде обычного нативного платформенного приложения пригодного для распространения обычным для ОС способом (включая магазины приложений и тп).

Грубо говоря, это выглядит так - приложение открывает окно WebView (по сути собственный веб браузер) в которое загружается веб контент из пакета приложения (просто как файлы или с создаваемого локального сервера).

Кроме стандартных возможностей браузера программисту предоставляются различные API доступные из кода на javascript, которые позволяют реализовать все доступные и обычному нативному приложению возможности. Пользовательский интерфейс при этом создается как и для веб-приложения с использованием

возможностей HTML/CSS, причем для этого имеется огромное количество готовых фреймворков на любой вкус.

Самым распространенным решением такого типа является Cordova/PhoneGap, который используется сам по себе или в составе различных комплексных решений типа Ionic и пр. Мы не будем рассматривать это решение, так как оно не поддерживает ни одной из российских ОС даже потенциально. В момент начальной популярности Sailfish(предок "Аврора" ОС) была попытка портирования этого решения на Sailfish, но этот проект давно заброшен.

Также стоит отметить основную проблему решений этого типа - ограничения виртуальной машины Javascript и WebView, особенно в реализации многопоточности. Подобные ограничения не позволяют реализовать сложные большие приложения.

Смешанно-гибридное кроссплатформенное приложение

Это развитие чисто гибридных решений, которые решают основную проблему данной архитектуры. А именно: предлагается WebView использовать только для пользовательского интерфейса, а все остальные задачи (работа с данными, логика) выводятся в отдельные потоки вовне WebView. То есть, получается смесь обычного нативного типа и гибридного. Взаимодействие между этими частями происходит как через прямой вызов предоставляемого API, так и путем обычного обращения к локальному серверу (что делает такие приложения очень похожими на обычные клиент-серверные веб приложения, только сервер не удаленный, а локальный). И это одно из преимуществ подобной архитектуры, ведь если сделать среду программирования знакомой, например идентичной какой-либо серверному решению, то для разработки приложения понадобится только знание веб технологий и серверных решений, а для этого имеется много готовых разработчиков, готового кода, библиотек и тп.

Рассматриваемые в данной статье решения, как раз используют такой подход. NW.js и Electron используют среду Node.js с программированием на языке Javascript, а решение Rhomobile среду Ruby on Rails с языком Ruby.

Open-source решения с поддержкой(включая потенциальную) российских ОС

Рассмотрим некоторые решения с открытым исходным кодом, которые поддерживают (включая потенциально) российские ОС. На мой взгляд интерес представляют только решения с открытым исходным кодом, так как на Российском рынке иначе маловероятно портирование решения на российские ОС и аппаратные платформы (Elbrus, Baikal).

Qt

вебсайт: <https://www.qt.io/>

разработчик: QT Company

Наверное для разработчиков приложений для российских ОС это решение воспринимается не как какое-то кроссплатформенное, а как стандартное.

Используется C++ и QML. Решение существует для всех ОС, включая мобильные. Нет поддержки разве что WinCE/WM. И есть нюансы, например для “Аврора” придется делать адаптацию интерфейса под компоненты Silica, которые унаследованы от обычных Qt компонентов и поэтому адаптация не слишком сложная.

Java

вебсайт: <https://libericajdk.ru/>

разработчик: BellSoft

Java это одновременно и язык и платформа(включающая виртуальную машину). Кстати под платформу Java есть компиляторы и для других языков. Широко используется в серверной разработке. Для российских ОС есть готовое адаптированное решение - Liberica JDK от компании BellSoft. Данное решение построенное на базе Open JDK поддерживает все российские настольные и серверные ОС, также поддерживаются и зарубежные ОС. Java в основном конечно используется для серверных решений.

Mono/Xamarin

вебсайт: <https://www.xamarin.com/>

разработчик: Microsoft

Очень популярное (на мобильных ОС) нативное решение на базе .NET платформы с языком C# (Си Шарп).

Решение похожее на Java по архитектуре. Платформа .NET, как известно, была разработана компанией Microsoft, после обнаружения широко известного "фатального недостатка" в платформе Java и по странному стечению обстоятельств весьма похожа на Java. Платформа .NET и язык программирования C# широко используются на настольных и серверных Windows ОС от Microsoft. Решение Mono реализует платформу .NET на иных ОС, в том числе на Linux и тп. Решение Xamarin изначально разработанное для мобильных платформ (iOS и Android) сейчас двигается и на настольные ОС, включая Linux (Xamarin Forms на базе Linux GTK). Стоит отметить что подобное движение сейчас наблюдается со многими решениями изначально разработанными для мобильных ОС, где происходит наиболее быстрое развитие.

React Native

вебсайт: <https://reactnative.dev/>

разработчик: Facebook

Нативное решение на базе веб javascript решения React. Язык программирования - Javascript. Еще одно изначально мобильное решение, весьма популярное на мобильных ОС. В настоящее время двигается на настольные ОС. Есть порты под основные настольные СО, включая Linux, например - <https://github.com/status-im/react-native-desktop-qt>

Flutter

вебсайт: <https://flutter.dev/>

разработчик: Google

Такая крупная корпорация как Google, конечно имеет свое собственное решение, причем полностью новое, даже язык программирования новый - Dart. Изначально мобильное решение, как и аналоги, движется на настольные ОС (заявлено буквально несколько месяцев назад). Но пока поддержка настольных ОС сыровата, более-менее работает только на Ubuntu. Однако решение и язык продвигается довольно активно и стоит ожидать что поддержка настольных ОС будет доведена до нормального уровня.

Uno Platform

вебсайт: <https://platform.uno/>

разработчик: UNO Platform Inc.

Это и несколько последующих решений проходят под грифом “экзотика”. Нативное решение на базе .NET. Запуск UWP (Universal Windows Platform) кода на разных платформах. Для iOS и Android используют Xamarin. Для Linux используют Mono и Skia.

Kivy

вебсайт: <https://kivy.org/>

разработчик: сообщество разработчиков

Опять “экзотика”. Нативное решение с использованием популярного языка Python. Собственное решение для UI с отрисовкой через OpenGL.

Lazarus

вебсайт: <https://www.lazarus-ide.org/>

разработчик: Lazarus and Free Pascal Team

Очередная “экзотика” для ностальгирующих по Delphi и Pascal (автор с теплотой вспоминает и то и другое). Нативное решение с собственной средой разработки похожей на Delphi. Язык программирования - конечно Pascal.

Proton Native

вебсайт: <https://proton-native.js.org/>

разработчик: сообщество разработчиков

Опять “экзотика”. Нативное решение. Позиционируется как React Native на настольных ОС, то есть код совместим с React Native, проект получил 9 тысяч звезд на GitHub, но не развивается.

NodeGui

вебсайт: <https://docs.nodegui.org/>

разработчик: сообщество разработчиков

“Экзотика”, но набирающая популярность. Gui для Node.js приложения на базе Qt.

NW.js

вебсайт: <https://nwjs.io/>

разработчик: сообщество разработчиков

Это и последующее решение смешано-гибридного типа и очень популярны.

JavaScript + HTML/CSS. Основан на Node.js и Chromium.

Приложение открывает WebView которое обращается к внутреннему Node.js серверу, таким образом приложение состоит из обычного веб контента (HTML/CSS/JS) и Node.js Javascript кода в обычной серверной среде.

Electron

вебсайт: <https://www.electronjs.org/>

разработчик: OpenJS Foundation

Смешано-гибридное решение на базе Node.js и Chromium. Приложение запускает Javascript код в Node.js среде, можно из кода открывать окна с WebView и в них уже обычный веб контент (HTML/CSS/JS).

Очень популярный, на нем сделаны следующие проекты(приложения для настольных ОС) : Atom Editor, Microsoft Visual Studio Code, Slack, WhatsApp, Skype, Microsoft Teams, Figma и пр.

Rhomobile

вебсайт: <https://tau-platform.com/ru/>

разработчик: Тау Технологии

Решение позволяет разрабатывать чисто гибридные приложения и смешано-гибридные. В случае смешано-гибридного варианта используется среда похожая на Ruby on Rails и язык Ruby для локального сервера. Пользовательский интерфейс обычным для гибридных решений - в WebView, то есть HTML/CSS/JS и возможно использование любых веб-фреймворков.

Единственное решение, которое поддерживает все основные зарубежные и российские ОС, включая мобильные и настольные, включая “Аврора” и даже WinCE/WM (все еще использующуюся на ТСД), для которого решение включает собственный порт WebKit для поддержки современного HTML 5.

Кроме непосредственно фреймворка для разработки кроссплатформенных приложений, Тау Технологии предлагает другие продукты образующие комплексное решение для корпоративного сектора:

- RhoConnect - решение для синхронизации данных между устройством и удаленным сервером (в Rhomobile изначально имеется поддержка этого продукта)

- RhoBrowser - современный индустриальный браузер, на всех платформах. В отличие от обычного браузера предоставляет ряд дополнительных возможностей, позволяющих зачастую обойтись только чисто веб-приложением, без разработки нативных приложений.

решение	Поддержка операционных систем										
	настольные							мобильные			
	Windows	Mac OS X	Linux	ALT Linux	Astra Linux	РЕД ОС	ROSA Linux	iOS	Android	Аврора	WinCE WM
Qt	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Red
Java	Green	Green	Green	Green	Green	Green	Green	Red	Yellow	Red	Red
Mono/Xamarin	Green	Green	Yellow	Yellow	Yellow	Yellow	Yellow	Green	Green	Red	Red
React Native	Green	Green	Green	Green	Green	Green	Green	Green	Green	Red	Red
Flutter	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Green	Green	Red	Red
Uno Platform	Green	Green	Green	Yellow	Yellow	Yellow	Yellow	Green	Green	Red	Red
Kivy	Green	Green	Green	Yellow	Yellow	Yellow	Yellow	Green	Green	Red	Red
Lazarus	Green	Green	Green	Yellow	Yellow	Yellow	Yellow	Red	Red	Red	Red
Proton Native	Green	Green	Green	Yellow	Yellow	Yellow	Yellow	Red	Red	Red	Red
NodeGui	Green	Green	Green	Yellow	Yellow	Yellow	Yellow	Red	Red	Red	Red
NW.js	Green	Green	Green	Yellow	Yellow	Yellow	Yellow	Red	Red	Red	Red
Electron	Green	Green	Green	Yellow	Yellow	Yellow	Yellow	Red	Red	Red	Red
Rhomobile	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green

Рисунок 2. Сводная таблица по кроссплатформенным решениям

Кроссплатформенная разработка с Rhomobile

Подробнее о Rhomobile и Tay Технологии

Рассмотрим более подробно решение Rhomobile. Немного истории - стартап Rhomobile был создан еще в 2008 году. В 2011 году стартап был приобретен фирмой Motorola. В 2014 году фирма Zebra приобрела у Motorola подразделение в которое

входила команда Rhomobile. В 2015 году несколько участников команды Rhomobile основали компанию Tay Технологии. В 2016 году Zebra прекращает разработку Rhomobile (последняя версия 5.4), делает весь код полностью открытым. Компания Tay Технологии становится технологическим партнером Zebra и получает полный доступ к основному репозиторию решения, продолжая развивать и поддерживать его. За несколько лет команда Tay Технологии выпустила несколько новых версий продукта. Последняя на момент написания этой статьи - 7.3 выпущена в декабре 2020 года.

Была добавлена поддержка всех основных российских ОС, включая мобильную “Аврора”, а также ряд новых возможностей. На сегодняшний день Rhomobile это единственное решение, которое позволяет разрабатывать кроссплатформенные приложения с единым кодом под все основные зарубежные и российские ОС.

Архитектура Rhomobile

Архитектура похожа на NW.js и Cordova/PhoneGap. Приложение на старте создает WebView в которую загружает контент с локального веб-сервера. Далее веб контент взаимодействует с локальным сервером как обычное веб приложение с удаленным сервером, только в данном случае сервер локальный. На локальном сервере создается среда похожая на широко известное решение Ruby on Rails - виртуальная машина с поддержкой языка Ruby на котором реализован веб сервер и на котором разработчик пишет код работающий с данными, реализующий бизнес-логику и управляющий/взаимодействующий с веб контентом в WebView.

Вам кажется что слишком сложно переносить серверные технологии на устройство и отделять UI от логики в разных средах - UI в среде WebView (по сути аналогичном обычному браузеру), а логика в среде Ruby с похожим на Ruby on Rails фреймворком ?

Но если подумать, то у такого решения есть много преимуществ :

- Обе среды хорошо знакомы, особенно WebView(HTML/JS/CSS) - веб разработчиков очень много на рынке, в любом корпоративном решении наверняка есть и веб решения, с веб приложениями и тп, и все эти наработки и даже напрямую код, можно использовать при создании и нативного приложения. Само собой что и веб разработчики смогут, без переподготовки, принять участие в разработке приложений.
- Что касается Ruby и Ruby on Rails, то эта среда конечно менее распространена по сравнению с HTML/JS, но тоже весьма популярна среди серверных разработчиков. Язык Ruby довольно простой, но мощный ООП язык, очень удобен при работе с моделями баз данных, скриптования бизнес логики и тп.
- Таким образом соединяя хорошо известные среды с огромным количеством готовых решений, кода, примеров, большим комьюнити мы получаем решение для разработки приложений с единым кодом сразу для всех основных операционных систем, при этом разработчикам не нужно вникать в особенности конкретных систем + можно использовать веб наработки, веб разработчиков и серверных разработчиков без переучивания.

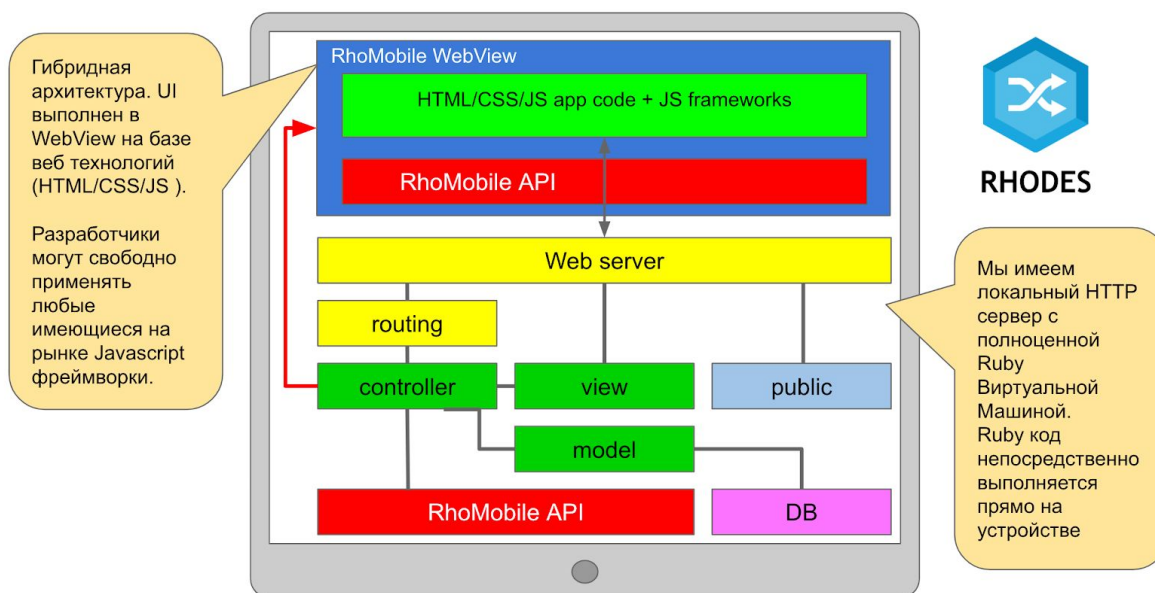


Рисунок 3. Архитектура Rhomobile приложения.

Разработка с Rhomobile

В рамках данной статьи мы не можем показать детальный процесс разработки с примерами кода и тп. Поэтому предлагаю посмотреть другие материалы на нашем сайте - <https://tau-platform.com/ru/>.

Статья написана на основе доклада на конференции OS Day 2020.

Видео доклада - <https://youtu.be/v4Q2F6WTPpU>

Разработка с Rhomobile для Российских ОС, с примером приложения - http://files.tau-platform.com/Documents%20/2021_Rhomobile_and_Russian_OS_Tau_Tech_nologies_RUS.pdf

Спасибо за внимание !