



Работа над поддержкой российских мобильных ОС
в кроссплатформенном фреймворке
Tau Платформа (RhoMobile)

XXI конференция разработчиков свободных программ

*Александр Епифанов
Tau Технологии
Санкт-Петербург*



Контекст: кроссплатформенная разработка

ПРОБЛЕМА:

- Множество платформ с разными SDK
- Отдельная команда и код для каждой ОС
- Появляются новые ОС - размывается труд и компетенции
- Высокие затраты и длительные сроки

РЕШЕНИЕ:

- **Одна кодовая база**
- **Нативные артефакты сборки**
- **Единый стек и прикладной API**

```

application.rb
You, 7 years ago | 5 authors (Jon Tara and others)
1 class AppApplication < Rho::RhoApplication
2   def initialize
3     Rho::Keyboard.keyboardDisplayRequiresUserAction = false;
4     @default_menu = {}
5
6     Rho::Application.setApplicationNotify( lambda do |callback_data; event_type, event_data|
7       event_type = callback_data['applicationEvent']
8
9       case event_type
10        when Rho::Application::APP_EVENT_ACTIVATED then on_activate_app
11        when Rho::Application::APP_EVENT_DEACTIVATED then on_deactivate_app
12        when Rho::Application::APP_EVENT_UICREATED then on_ui_created
13        when Rho::Application::APP_EVENT_UIDESTROYED then on_ui_destroyed
14
15      end # case event_type
16    end # lambda
17  )
18
19  User.on_app_initialize
20  Translation.set_language User.language
21
22  if (Rho::Push.enumerate().length > 0)
23    Rho::Push.startNotifications("/app/User/push_received_callback")
24    Rho::Timer.start(1000, /app/User/push_id_timer_callback, "rho_callback-1")
25  end
26
27  end # initialize
  
```



Кроссплатформенный фреймворк для мобильных и настольных приложений

- **Равнозначное название - RhoMobile. Tau Платформа это не форк или дериватив, а ребрендинг для РФ**
- **15+ лет развития** — с 2008 года
- **Независимая российская команда** — с 2015 года
- **Гибридная архитектура** — Ruby/JS + HTML5
- **Open Source** — MIT лицензия

ФОКУС НА БИЗНЕС-ЗАДАЧИ:

- Корпоративные мобильные приложения. Настольные тоже можно
- Терминалы сбора данных (ТСД)
- Промышленные решения: киоски, погрузчики
- Офлайн приложения с локальной обработкой данных и синхронизацией
- Защита кода и данных

УНИВЕРСАЛЬНОЕ СРЕДСТВО РАЗРАБОТКИ

МОБИЛЬНЫЕ:

Android

iOS

Аврора

Rosa Mobile

ALT Mobile

РЕД ОС М

ДЕСКТОП:

Linux
Ubuntu, ALT, Rosa, РЕД,
Астра

Windows

OS X



ОСНОВА:

- **Rake** - система сборки на Ruby
- **В идеале** - сгенерировать проект для родного инструмента (например, XCode)

КАСКАД ЗАДАЧ СБОРКИ:

Конфигурация окружения



Подготовка бандла (компиляция Ruby → ISEQ, минификация JS/CSS)



Копирование ассетов (картинки и др.)

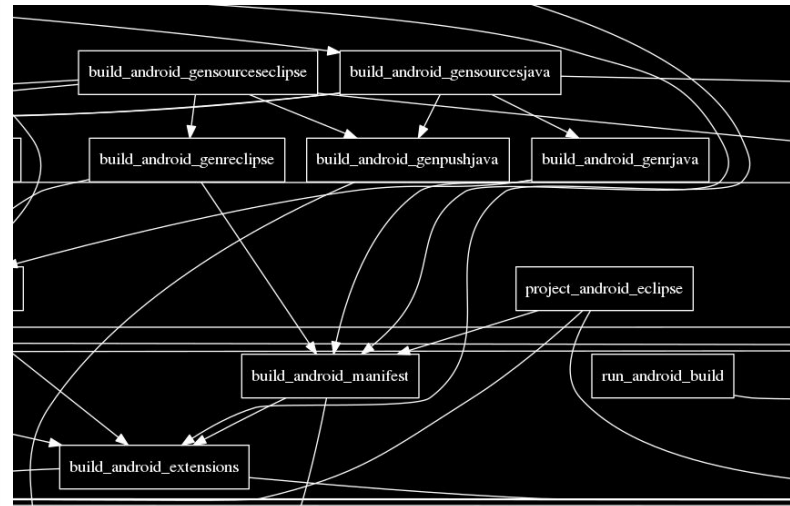


Сборка нативных компонентов и расширений



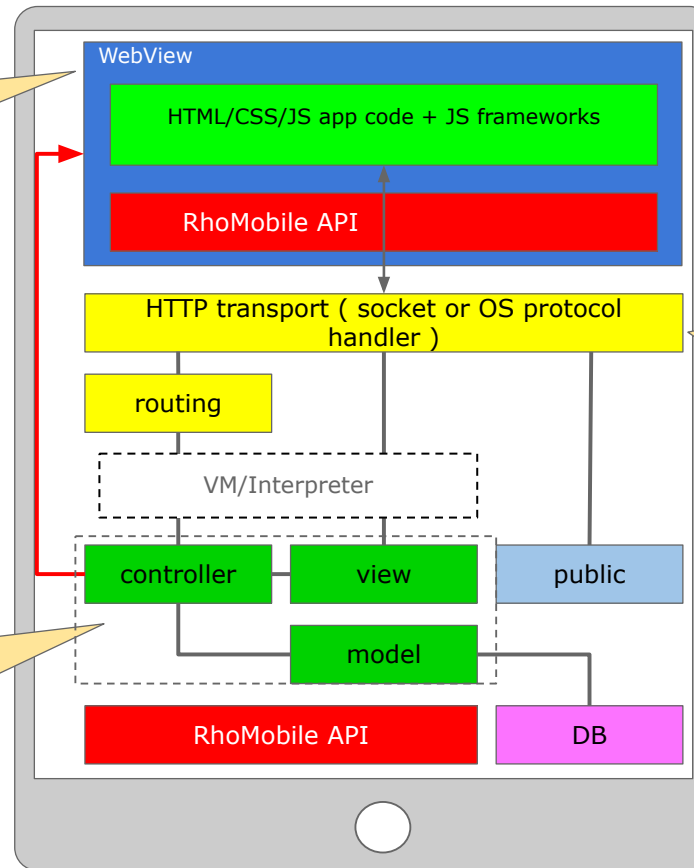
Упаковка в целевой артефакт

```
> rake device:android:production
> rake device:iphone:production
> rake build:altMobile:production
```



Гибридная архитектура. UI выполнен в WebView.

MVC фреймворк. В нашем случае - модифицированный Ruby-on-Rails.



Локальный HTTP транспорт с виртуальной машиной за ним, исполняющей код пользователя.

- **Аналогия:** похож на Electron, но с фокусом на мобильные платформы и enterprise
- В качестве интерпретатора используем MRI реализацию Ruby
- Экспериментировали с другими рантаймами, в частности Node/V8, в принципе архитектура позволяет реализовать различные интерпретаторы / фреймворки

1. НАТИВНЫЕ ФРЕЙМВОРКИ И ИНСТРУМЕНТЫ

- Android SDK/NDK
- iOS SDK
- Аврора SDK
- Компилятор, линкер

2. ПРИКЛАДНЫЕ ФРЕЙМВОРКИ

- Qt
- GTK
- ...

3. WebView

- QtWebEngine
- WebKitGTK
- Системный компонент
- Кастомный компонент (например, Crosswalk)

4. Каркас приложения

- Окно с WebView
- Инициализация VM
- Инициализация HTTP транспорта

Процесс: создаем базовое приложение на целевой платформе → интегрируем компоненты Tau Платформы → настраиваем WebView → адаптируем сборку

Типичные проблемы при портировании

✗ Отсутствие нормальных средств разработки

либо вообще нет, либо рудиментарные, либо неудобные (сравниваем с iOS/Android)

✗ Отсутствие кросс-компиляции

приходится собирать приложение прямо на целевом устройстве, либо на эмуляторе

✗ Проблемы с WebView

падения, некорректный рендеринг, слабые возможности по управлению загрузкой контента

✗ Специфичная периферия

сканеры ШК, модули NFC и пр.



Попытка 1: Qt5 + QtWebEngine



Попытка 2: Qt6 + QtWebEngine



Решение: GTK4 + WebKitGTK



✗ При нажатии на текстовое поле `экранный клавиатура не открывается

- Включили **WAYLAND_DEBUG=client** для трассировки
- Увидели в логе композера что все необходимые протоколы поддержаны:
zwp_input_method_manager_v2
zwp_text_input_manager_v3
zwp_virtual_keyboard_manager_v1
- Однако мы не увидели строк **wl_registry.bind** к этим интерфейсам. То есть Qt в этом запуске не подключился к text input v2/v3, а выбрал более простой путь. В логе Qt это зафиксировано сообщением:

```
qt.qpa.wayland: using input method: QComposeInputContext
```

Причина: Qt не сделал бинд к новым протоколам **zwp_text_input** и остался на классическом **QComposeInputContext** (работа через wl_keyboard). Экранный клавиатура в таком режиме не появляется.

На x86 эмуляторе:

- ✓ Приложение работало стабильно, WebView отображался, клавиатура работала

На ARM (aarch64) эмулятор и Anbernic RG552:

- ✗ Белый экран вместо контента
- ✗ ProcessGone: 3 (139)
- ✗ QtWebEngineProcess падает с SIGSEGV

```
ProcessGone: 3 (139) Render process exited with code: 139
Vulkan disabled or failed to initialize
vaInitialize failed: unknown libva error
```

Backtrace подтвердил: падает QtWebEngineProcess --type=renderer внутри libQt6WebEngineCore.so.6

ГИПОТЕЗА:

На x86: llvmpipe (софт-рендер)

На ARM: ANGLE + OpenGL ES — здесь падения

ПОПЫТКИ ИСПРАВИТЬ РАЗНЫМИ КОМБИНАЦИЯМИ ФЛАГОВ:

```
QTWEBENGINE_CHROMIUM_FLAGS:  
  --use-gl=angle  
  --use-angle=gl  
  --disable-vulkan  
  --ozone-platform=x11 или --ozone-platform=wayland  
  --no-sandbox  
  --in-process-gpu  
  --enable-features =  
  UseOzonePlatform,NetworkServiceInProcess2,TracingServiceInProcess  
  --enable-logging=stderr --v=1
```

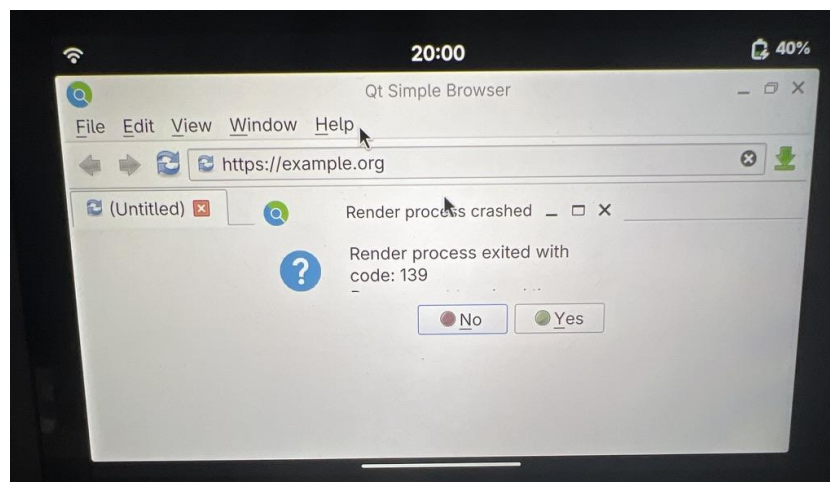
```
✘ [FATAL:platform_selection.cc(50)\] Invalid ozone platform: wayland  
✘ Render process exited with code: 139
```

Цель тестов: выяснить, где именно ломается WebView - в графическом бэкенде, в sandbox или в выборе платформы.

ПРОВЕРКА НА Qt6 EXAMPLE:

Загрузили стандартное example-приложение **Qt6 SimpleBrowser**

✗ Та же проблема: белый экран, код 139



Дополнительное тестирование: Qt6 + QtWebEngine на ARM работает. Собирали Tay Платформу под Raspberry Pi 5 (Raspberry Pi OS) — стабильно работает

Вывод: проблема специфична для QtWebEngine на ALT Mobile ARM, не для Qt на ARM в целом

Багрепорт: bugzilla.altlinux.org/55845

АРХИТЕКТУРА

- **Qt** → используем .pro проекты для сборки внутренних библиотек (rholib, ruby, sqlite3, syncengine, curl)
- **WebView** → WebKitGTK
- **UI** → GTK4 + libadwaita

КЛЮЧЕВЫЕ ИЗМЕНЕНИЯ:

- **JS-мост:** prompt-бридж вместо QWebChannel **__rhoNativeApi.apiCall** из WebView → Ruby API. Для приложений интерфейс остался прозрачным — API работает без изменений
- **Меню:** GtkMenuButton + popover
- **UI-поток:** вызовы через **g_idle_add**

РЕЗУЛЬТАТ

- Успешная сборка и запуск тестового приложения на Anbernic RG552, поддержку периферии типа камеры и сенсоров пока не реализовали

! Собирали на ALT Linux p11

Нужны gcc-13 и g++-13

Можно собирать под **x86** или **aarch64**

В планах - завернуть зависимости сборки в Docker контейнер с QEMU для автоматической кросс-компиляции

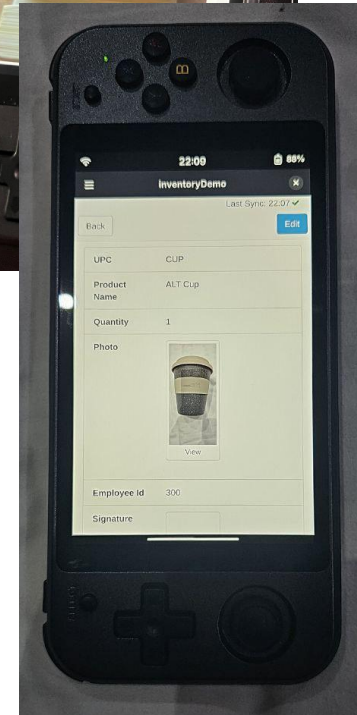
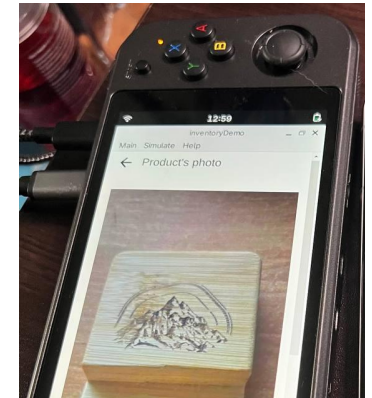
ЗАВИСИМОСТИ:

```
> sudo apt-get install git gcc gcc-c++ ruby libruby-devel irb \
gmp-devel libGL-devel libGLU-devel rpm-build zlib-devel \
qt5-base-devel qt5-webengine-devel qt5-multimedia-devel libglfw \
libglfw-devel \
libadwaita-devel libgtk4-devel libwebkitgtk6.0-devel
```

```
grep -qxF 'export QTDIR=/usr/share/qt5' ~/.bashrc || echo 'export
QTDIR=/usr/share/qt5' >> ~/.bashrc
```

СБОРКА:

```
> rake build:altMobile:production      # RPM
> rake build:altMobile:all             # Binary
```



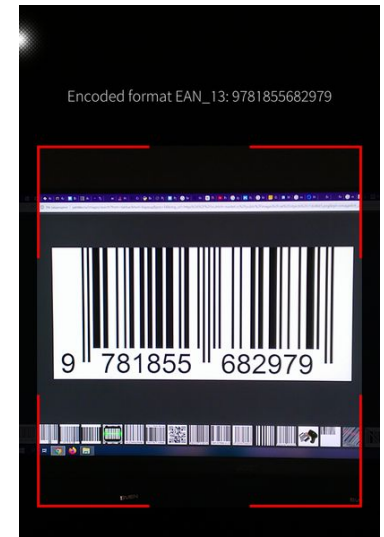


А В Р О Р А

- **Qt/QML + QtWebEngine**
- **SDK**
- **IDE (Qt Creator)**
- **Документация**

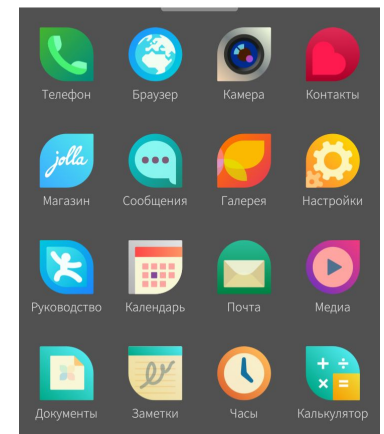
СБОРКА:

```
> rake device:sailfish:production
```



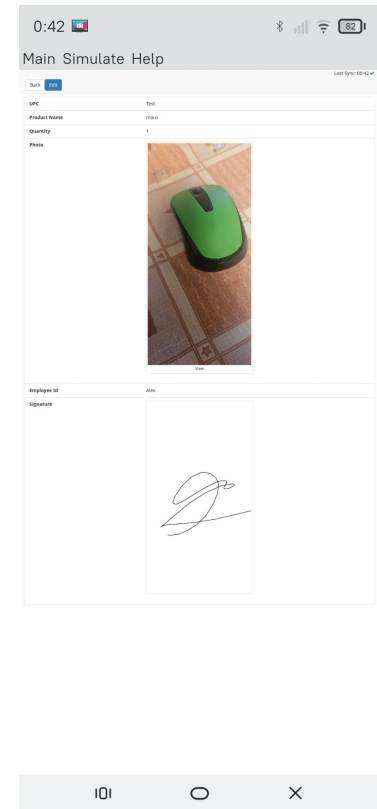
Call method take

```
{"barcode"=>"9781855682979",  
"format"=>"EAN_13", "status"=>"ok"}
```





- **Qt + QtWebEngine, как на десктопе**
- **Средства разработки - развиваются**
- **Вроде бы в новом SDK есть кросскомпиляция, но мы пока не пробовали, строили прямо на P-ФОН**

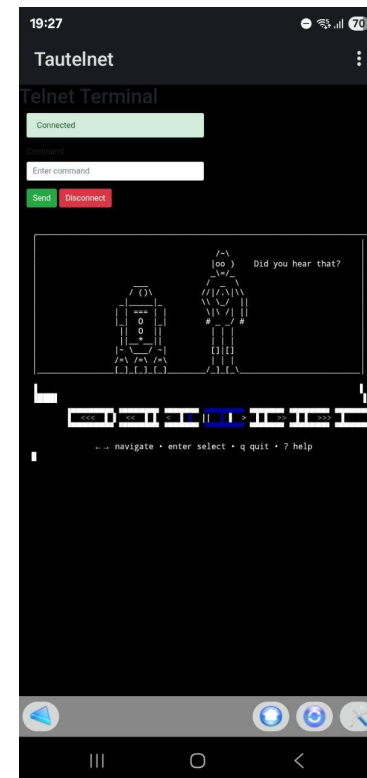


СБОРКА:

```
> rake device:linux:production
```



- **AOSP**
- **Весь тулчейн от Андроид**
- **Нет GMS: из-за этого может что-то не работать, например софтверный сканер ШК**

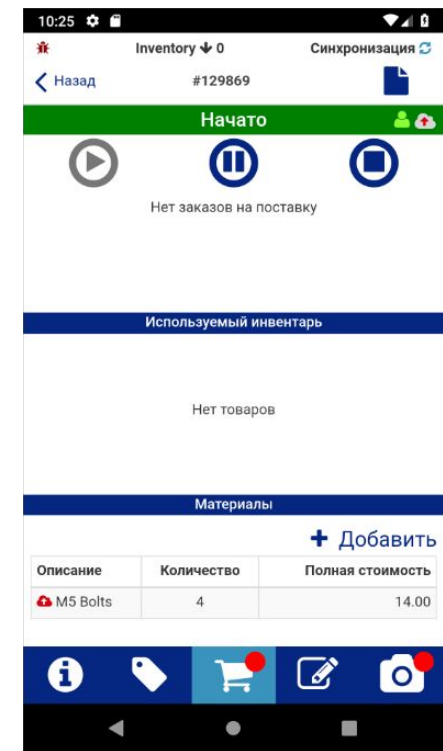
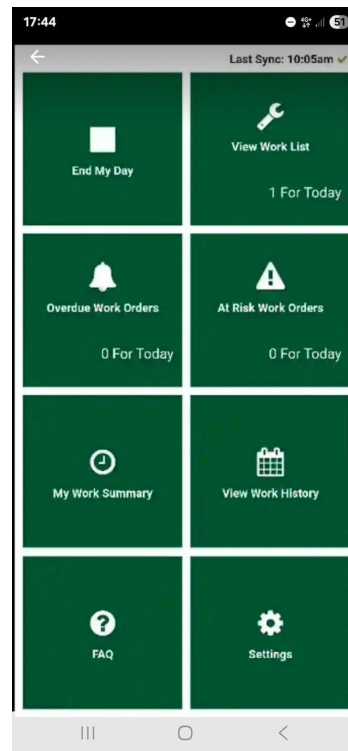


СБОРКА:

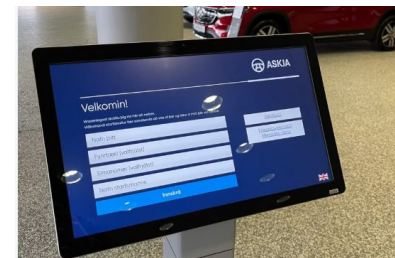
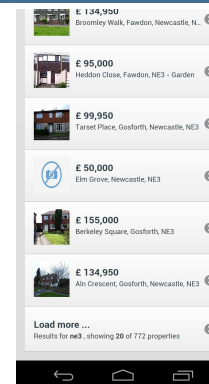
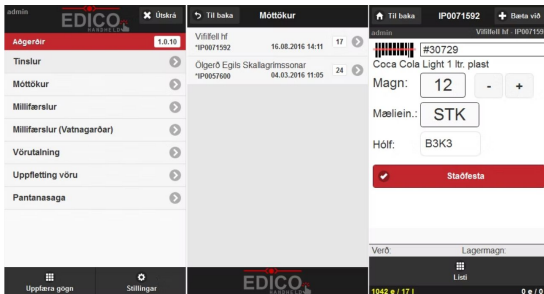
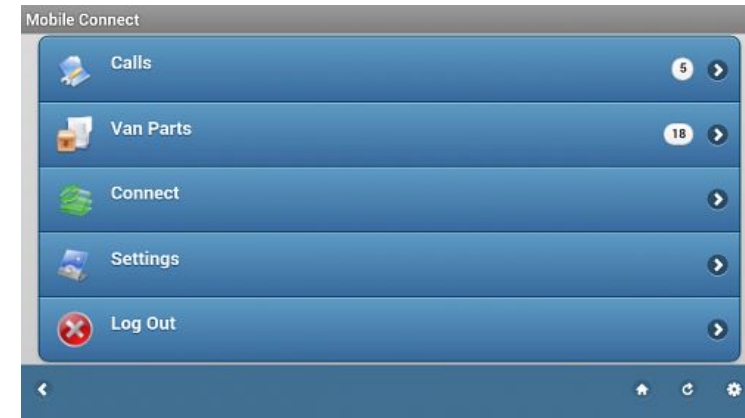
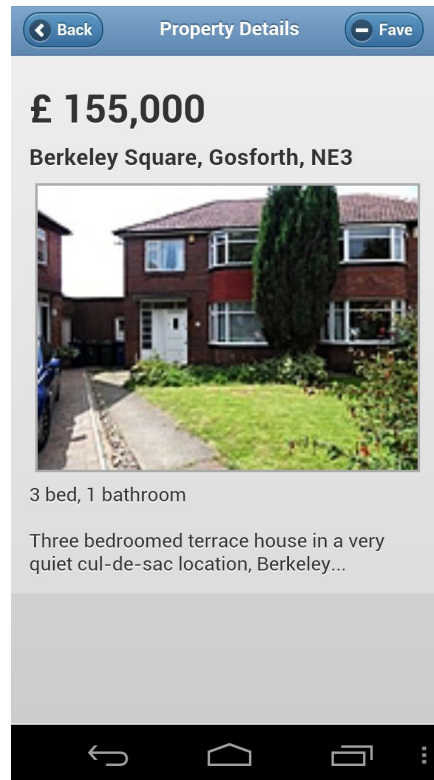
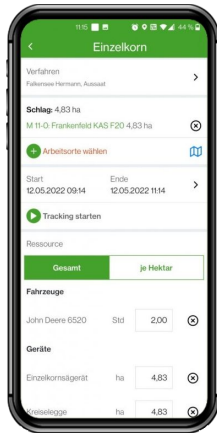
```
> rake device:redmobile:production
```

Что мы сами разрабатываем на своём фреймворке

- **Приложения** для платных клиентов - разработка и поддержка, например приложение для сервисных инженеров для зарубежного заказчика
- Собственный коробочный продукт - промышленный браузер **Тау Браузер**
- **Решения** в разработке, например Telnet клиент для ТСД

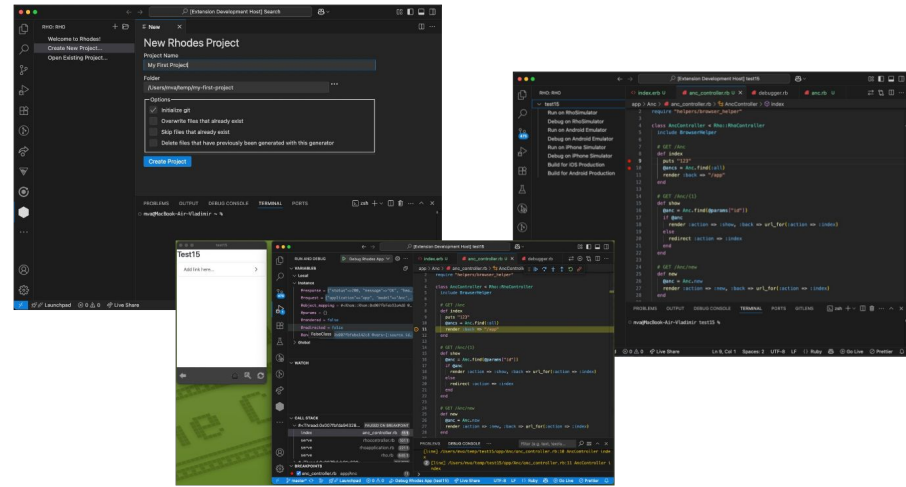


- **Разработчики:** от команд из одного человека до крупных компаний
- **География:** Россия, Сингапур, Индия, Германия, Исландия, США
- **Спектр решений:** мобильный сервис, страхование, банкинг, сельское хозяйство, ритейл, энергетика, производство, обслуживание



ПОРТИРОВАНИЕ

- ✓ **ALT Mobile** — завершено
- ! **Astra Mobile** — в работе

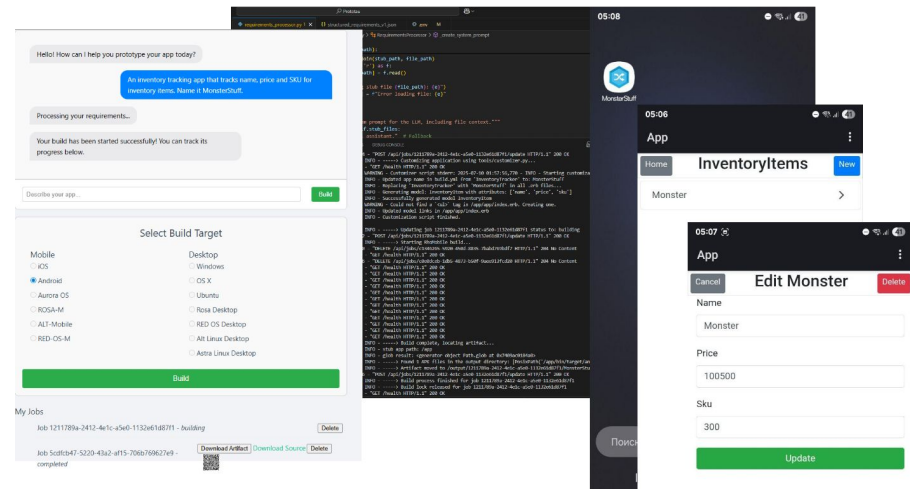


ОПТИМИЗАЦИЯ

- **Ускорение холодной сборки**

ИНСТРУМЕНТЫ

- **Универсальная IDE с поддержкой всех российских ОС**
- **Улучшение DevEx: AI помощники, инструменты быстрого прототипирования**



РАЗРАБОТЧИКИ

- Пишите переносимый прикладной софт
- Поддерживайте новые платформы
- Развивайте инструментарий

КОМПАНИИ

- Создавайте решения, работающие на всех ОС и устройствах
- Делитесь опытом
- Приходите за новыми фичами :)

- <https://github.com/rhobile/rhodes> - основной репозиторий
- <https://tau-platform.ru/ru/developers/downloads/> - релизы
- <http://docs.tau-platform.com/> - документация
- <https://forums.tau-platform.com/> - форум
- <https://tau-platform.ru> - сайт компании
- <https://tau-browser.ru> - браузер на Тау Платформе
- **Примеры приложений:**
 - <https://github.com/tauplatform/rhodes-system-api-samples>
 - <https://github.com/tauplatform/kitchensinkRuby>
 - <https://github.com/tauplatform/inventoryDemo-mobileApp>
 - <https://github.com/tauplatform/kitchensinkJS>
 - <https://github.com/tauplatform/universal-push-example>

- info@tau-platform.com
- tg: @Iskandar_Epiphanes

