

ЭКОСИСТЕМА ОС АВРОРА

СТАТУС ПОДДЕРЖКИ ФРЕЙМВОРКА FLUTTER

Глазков Денис, разработчик ОС
Аврора

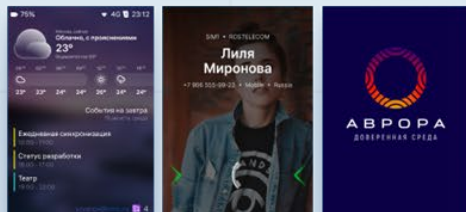
Сегодня обсудим

- ОС Аврора
- Архитектура Flutter
- Текущий статус портирования Flutter на ОС Аврора
- Результаты и будущие планы

ОПЕРАЦИОННАЯ СИСТЕМА

Четвертое поколение

Современный мобильный функционал с фокусом на безопасности



АВРОПА
СВОЯ СИСТЕМА

более 400 000
устройств
в промышленной эксплуатации

ЭКОСИСТЕМА ПРИЛОЖЕНИЙ

Более 100 партнеров

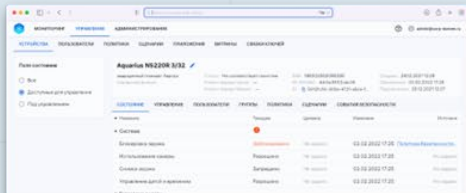
создают свои решения для ОС Аврора



МОБИЛЬНЫЕ СЕРВИСЫ И MDM

Аврора Центр

Управление парком устройств
Push-сервис, сервис обновлений,
магазин приложений



СРЕДСТВА РАЗРАБОТКИ

Аврора SDK

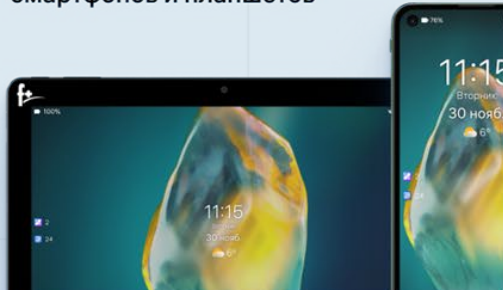
для Windows, MacOS и Linux



МОДЕЛЬНЫЙ РЯД

10 устройств

смартфонов и планшетов



- Дочерняя компания ПАО «Ростелеком»
- Является участником open-source проектов и организаций
- Ведет образовательные программы в ведущих ВУЗах страны
- Офисы разработки в Москве, Санкт-Петербурге, Нижнем Новгороде и Иннополисе



Фреймворк Qt

- Очень мощный и популярный кроссплатформенный фреймворк

Что не так с Qt?

- Не очень популярен среди Android и iOS разработчиков

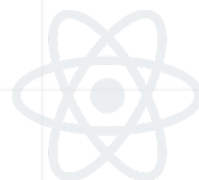


? Как привлечь Android и iOS решения в экосистему ОС Аврора?

Добавить поддержку альтернативного кроссплатформенного фреймворка, чтобы увеличить переиспользование кода между мобильными платформами

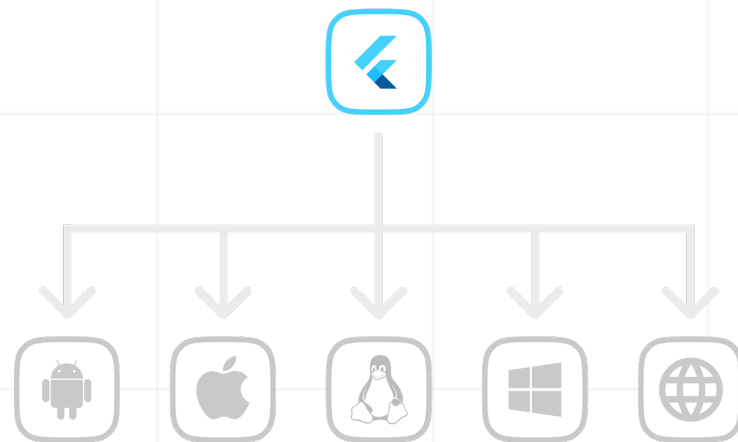
Требования к фреймворку

- Поддерживает популярные мобильные ОС
- Лицензия без юридических последствий
- Компилируемый
- Удобный для портирования на новую ОС



Фреймворк	Лицензия	Язык	Платформы	Компилируемый
Xamarin	MIT	C#	Mobile, Desktop	-
Ionic	MIT	JavaScript	Mobile, Web	-
React Native	MIT	JavaScript	Mobile, UWP	-
Flutter	BSD 3	Dart	Mobile, Desktop, Web	+
KMM + Compose	Apache-2.0	Kotlin	Android, Desktop, Web	+

- Работает на Android, iOS, Windows, Linux и Web
- Имеет поддержку JIT и AOT компиляции
- Набирает популярность среди мобильных разработчиков
- Поддерживает систему плагинов
- Основной контрибьютор - Google, за плагины отвечает сообщество
- Имеет удобную архитектуру для портирования
- Имеет лицензию с отсутствием ограничений при использовании исходного кода



Flutter SDK

Скрипты для сборки/запуска приложений

Flutter Framework

UI библиотека на языке Dart

Flutter Engine

Рантайм для Dart кода - Библиотека на C++

Flutter Embedder

Компонент для запуска приложения

Flutter SDK

Скрипты для сборки/запуска приложений

Flutter Framework

UI библиотека на языке Dart

Flutter Engine

Рантайм для Dart кода - Библиотека на C++

Flutter Embedder

Компонент для запуска приложения



Начало 2022 года

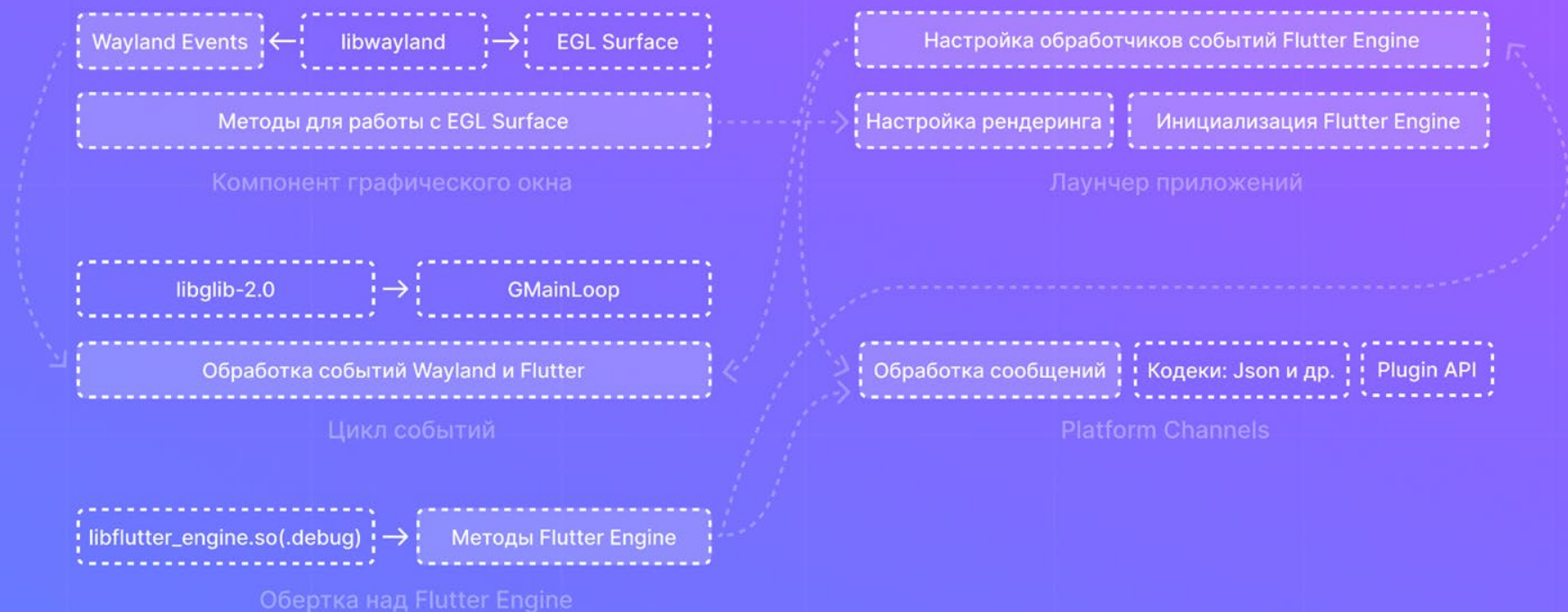
Старт разработки Flutter Embedder
для ОС Аврора

2021

2022

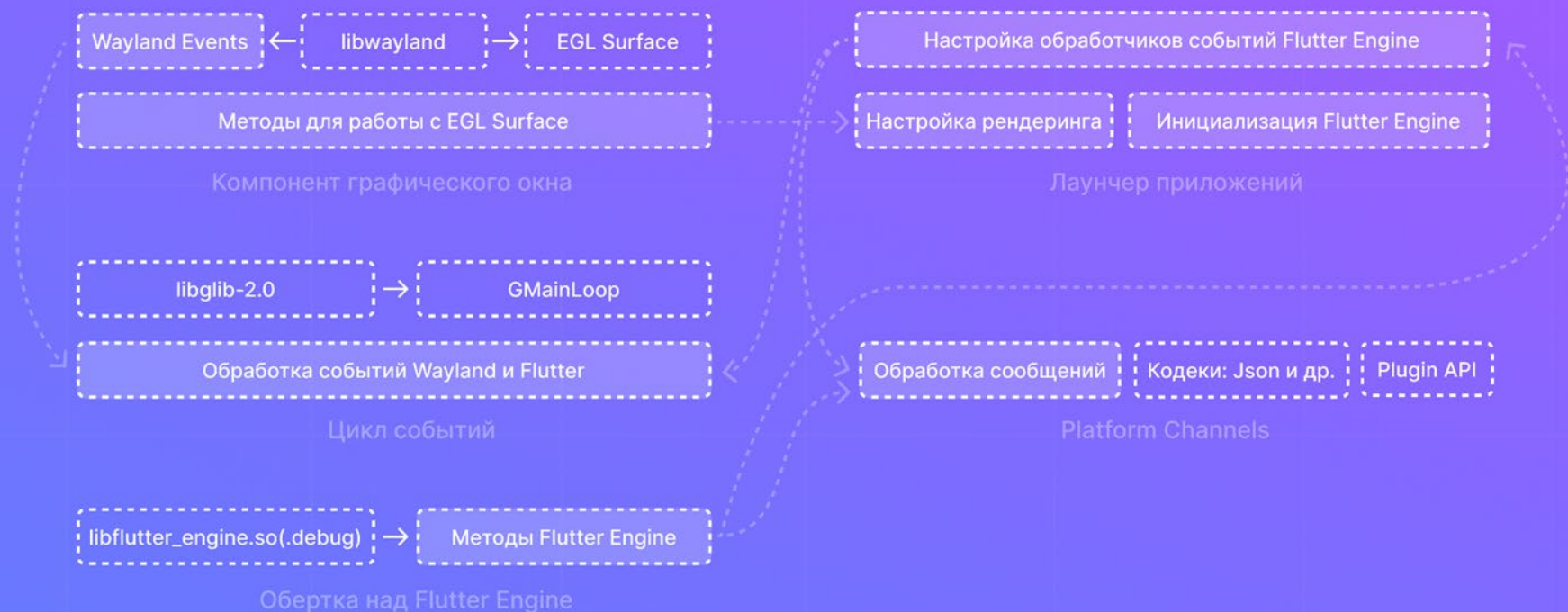
2023

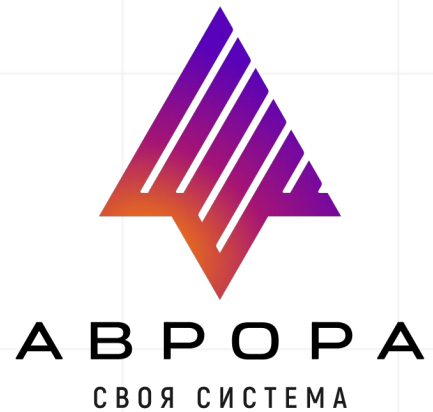
Почему не взять Flutter Embedder для Linux?











- **Конец 2022 года**

Старт работ над адаптацией Flutter SDK
под ОС Аврора

3.0.0

3.3.5



3.3.10

3.7.2

3.7.7



```
$ flutter create --platforms=aurora --org=com.example application
```

```
Creating project application...
```

```
Running "flutter pub get" in application...
```

```
Wrote 21 files.
```

```
All done!
```

```
In order to run your application, type:
```

```
  $ cd application
```

```
  $ flutter run
```

```
Your application code is in application/lib/main.dart
```



```
$ flutter build aurora --debug
```

```
👉 Building with sound null safety 👉  
Building Aurora application...
```



```
$ flutter build aurora --release
```

```
👉 Building with sound null safety 👉  
Building Aurora application...
```



```
$ flutter build aurora --profile
```

```
👉 Building with sound null safety 👉  
Building Aurora application...
```


`application/build/aurora/arm/release/RPMS`



`com.example.application-0.1.0-1.armv7hl.rpm`



```
$ flutter doctor
```

```
[x] Aurora toolchain - develop for Aurora OS
```

```
  x Platform SDK is required for Aurora development.
```

```
    Platform SDK setup:
```

```
    https://community.omprussia.ru/documentation/software\_development/psdk/setup.html
```

```
    You may not have set the PSDK_DIR environment variable.
```

Виды плагинов для фреймворка Flutter

- Dart-only плагины
- FFI плагины
- Platform Channel плагины



Flutter

webview_flutter

PLATFORM | ANDROID IOS

3080

LIKES

120

PUB POINTS

100%

POPULARITY

video_player

PLATFORM | ANDROID IOS WEB

2331

LIKES

140

PUB POINTS

100%

POPULARITY

geolocator

PLATFORM | ANDROID IOS MACOS LINUX

4210

LIKES

140

PUB POINTS

100%

POPULARITY

camera

PLATFORM | ANDROID IOS

1656

LIKES

140

PUB POINTS



```
$ flutter create --platforms=aurora --template=plugin plugin
```

```
Creating project plugin...
```

```
Running "flutter pub get" in plugin...
```

```
Running "flutter pub get" in example...
```

```
Wrote 37 files.
```

```
All done!
```

```
Your plugin code is in plugin/lib/plugin.dart
```



```
#include <plugin/plugin.h>
#include <flutter/method-channel.h>
#include <string>

namespace {
const std::string GET_PLATFORM_VERSION = "getPlatformVersion";
} /* namespace */

void Plugin::RegisterWithRegistrar(PluginRegistrar &registrar)
{
    registrar.RegisterMethodChannel(
        "plugin",
        MethodCodecType::Standard,
        [this](const MethodCall &call) {
            this->onMethodCall(call);
        }
    );
}

void Plugin::onMethodCall(const MethodCall &call)
{
    if (call.GetMethod() == GET_PLATFORM_VERSION) {
        onGetPlatformVersion(call);
        return;
    }

    unimplemented(call);
}
```



```
#include <plugin/plugin.h>
#include <flutter/method-channel.h>
#include <string>

namespace {
const std::string GET_PLATFORM_VERSION = "getPlatformVersion";
} /* namespace */

void Plugin::RegisterWithRegistrar(PluginRegistrar &registrar)
{
    registrar.RegisterMethodChannel(
        "plugin",
        MethodCodecType::Standard,
        [this](const MethodCall &call) {
            this->onMethodCall(call);
        }
    );
}

void Plugin::onMethodCall(const MethodCall &call)
{
    if (call.GetMethod() == GET_PLATFORM_VERSION) {
        onGetPlatformVersion(call);
        return;
    }

    unimplemented(call);
}
```




```
#include <plugin/plugin.h>
#include <flutter/method-channel.h>
#include <string>

namespace {
const std::string GET_PLATFORM_VERSION = "getPlatformVersion";
} /* namespace */

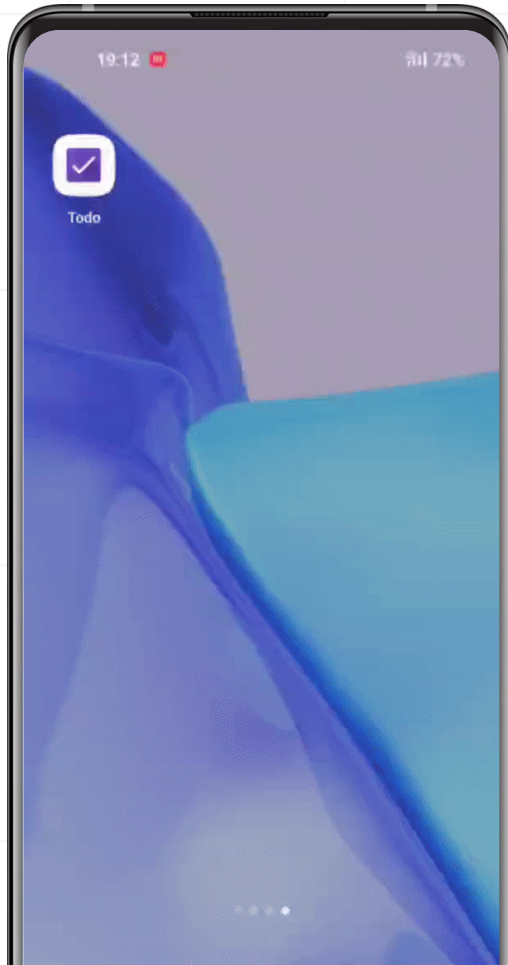
void Plugin::RegisterWithRegistrar(PluginRegistrar &registrar)
{
    registrar.RegisterMethodChannel(
        "plugin",
        MethodCodecType::Standard,
        [this](const MethodCall &call) {
            this->onMethodCall(call);
        }
    );
}

void Plugin::onMethodCall(const MethodCall &call)
{
    if (call.GetMethod() == GET_PLATFORM_VERSION) {
        onGetPlatformVersion(call);
        return;
    }

    unimplemented(call);
}
```

Текущий план по адаптации Flutter








А В Р О Р А

СВОЯ СИСТЕМА

Что осталось реализовать для поддержки фреймворка Flutter на ОС Аврора

- Добавить поддержку эмулятора с  **Hot Reload**
- Добавить поддержку **Dart DevTools**
- Добавить поддержку сборки под **aarch64** и **x86_64**
- Создать комьюнити вокруг Flutter для ОС Аврора

Будущие планы




Заключение

- Платформа Аврора - современная мобильная ОС и сервисы для корпоративного рынка
- ОС Аврора - самостоятельный дистрибутив с собственной историей и развитием
- ОС Аврора использует популярный кроссплатформенный фреймворк Qt как основной инструмент для разработки приложений
- Команда «Открытой мобильной платформы» внимательно следит за развитием кроссплатформенных средств разработки



А В Р О Р А
СВОЯ СИСТЕМА



СПАСИБО ЗА ВНИМАНИЕ
ВОПРОСЫ?