



ОС Аврора и Flutter. Статус поддержки.



**Денис
Глазков**

Открытая Мобильная Платформа

 glazzkoff

 d.glazkov@omp.ru



Mobius
2023 Spring



ОТКРЫТАЯ
МОБИЛЬНАЯ
ПЛАТФОРМА



**Денис
Глазков**

✈ glazzkoff

О докладчике

- Окончил Московский
Авиационный Институт
- Системный разработчик
в компании «Открытая
Мобильная Платформа»
- Разрабатываю ОС «Аврора»

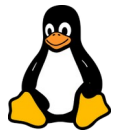
Сегодня обсудим

- Что такое ОС Аврора
- Что под капотом у фреймворка Flutter
- Сложности в поддержке Flutter на новой ОС
- Roadmap портирования фреймворка Flutter на ОС Аврора
- Подводные камни, которые мы встретили
- Как разрабатывать на Flutter под ОС Аврора
- Ближайшее будущее



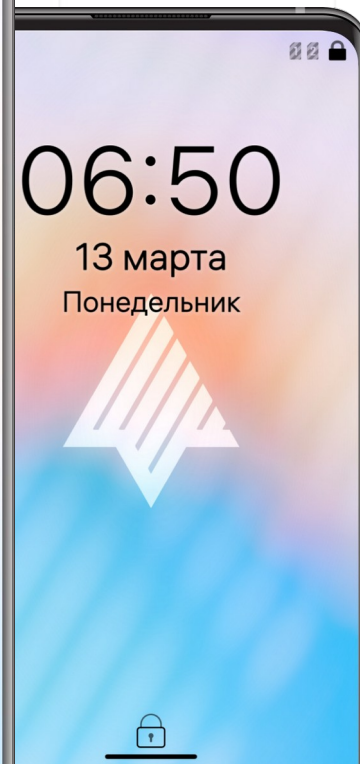
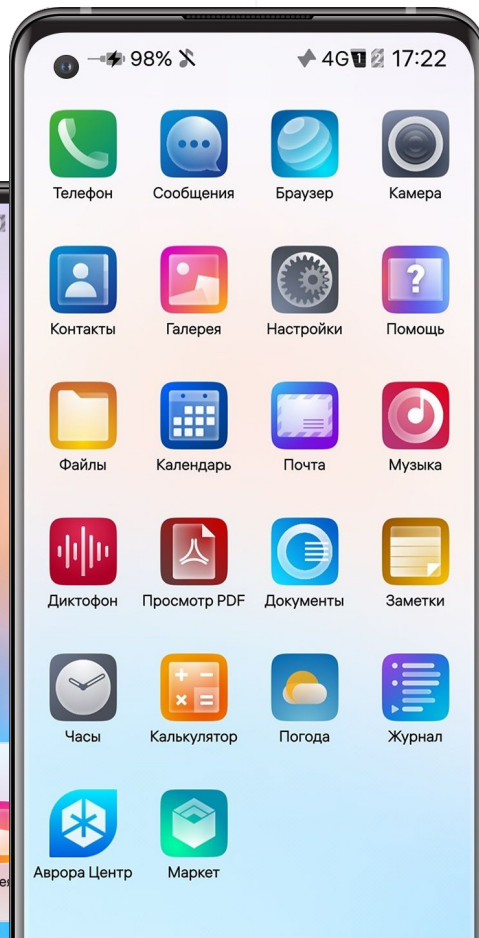
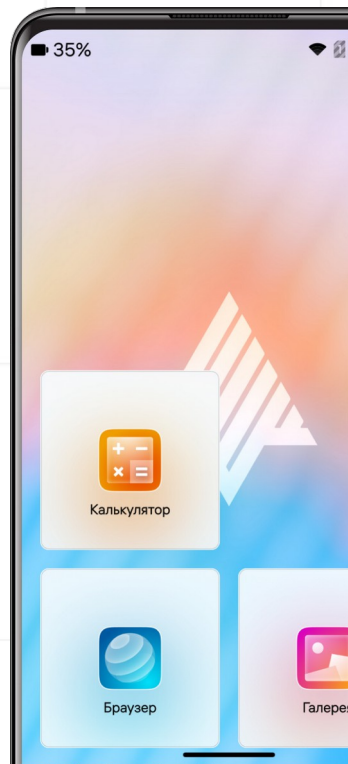
- Дочерняя компания ПАО «Ростелеком»
- Разрабатывает мобильную ОС Аврора
- Является участником open-source проектов и организаций
- Ведет образовательные программы в ведущих ВУЗах страны





А В Р О П А
СВОЯ СИСТЕМА

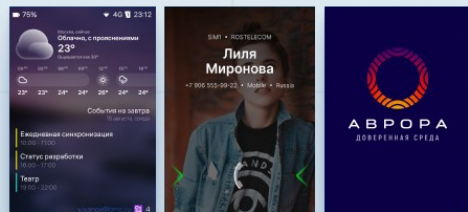




ОПЕРАЦИОННАЯ СИСТЕМА

Четвертое поколение

Современный мобильный функционал
с фокусом на безопасности



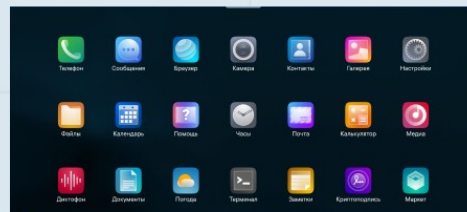
АВРОРА
СВОЯ СИСТЕМА

более 400 000
устройств
в промышленной эксплуатации

ЭКОСИСТЕМА ПРИЛОЖЕНИЙ

Более 100 партнеров

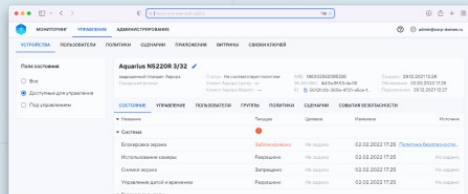
создают свои решения
для ОС Аврора



МОБИЛЬНЫЕ СЕРВИСЫ И MDM

Аврора Центр

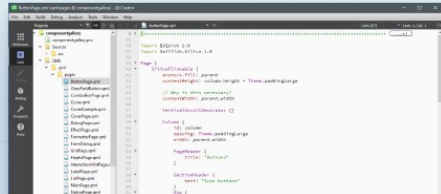
Управление парком устройств
Push-сервис, сервис обновлений,
магазин приложений



СРЕДСТВА РАЗРАБОТКИ

Аврора SDK

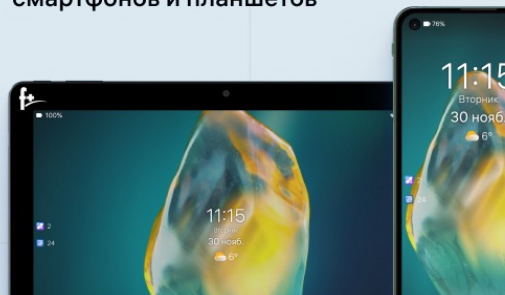
для Windows, MacOS и Linux



МОДЕЛЬНЫЙ РЯД

10 устройств

смартфонов и планшетов



Более подробно про ОС Аврора на **Mobius 2022 Autumn**





```
ApplicationWindow {  
    width: 640  
    height: 480  
  
    menuBar: MenuBar {  
        Menu {  
            title: "File"  
            MenuItem {  
                text: "Open"  
            }  
        }  
    }  
  
    Button {  
        text: "Hello World"  
        anchors.horizontalCenter: parent.horizontalCenter  
        anchors.verticalCenter: parent.verticalCenter  
    }  
}
```





```
#ifndef APP_MANAGER_H
#define APP_MANAGER_H

#include <QObject>

class AppManager : public QObject
{
    Q_OBJECT
    Q_PROPERTY(bool isDarkMode READ isDarkMode)

public:
    explicit AppManager(QObject *parent = nullptr);
    bool isDarkMode() const;

private:
    bool m_darkMode = false;
};

#endif /* APP_MANAGER_H */
```







```
ApplicationWindow {  
    ...  
  
    Components.KMMAgent {  
        id: agent  
        onCompleted: {  
            run(  
                "return kmm.AppConstants.links.API_URL",  
                function(result) {  
                    constants.apiUrl = result  
                }  
            )  
            ...  
        }  
    }  
}
```





Flutter

Сложности в поддержке Flutter на новой ОС

Фреймворк Flutter является кроссплатформенным,
только потому, что кто-то уже спортировал его на новую
платформу

Flutter SDK

Скрипты для сборки/запуска приложений

Flutter Framework

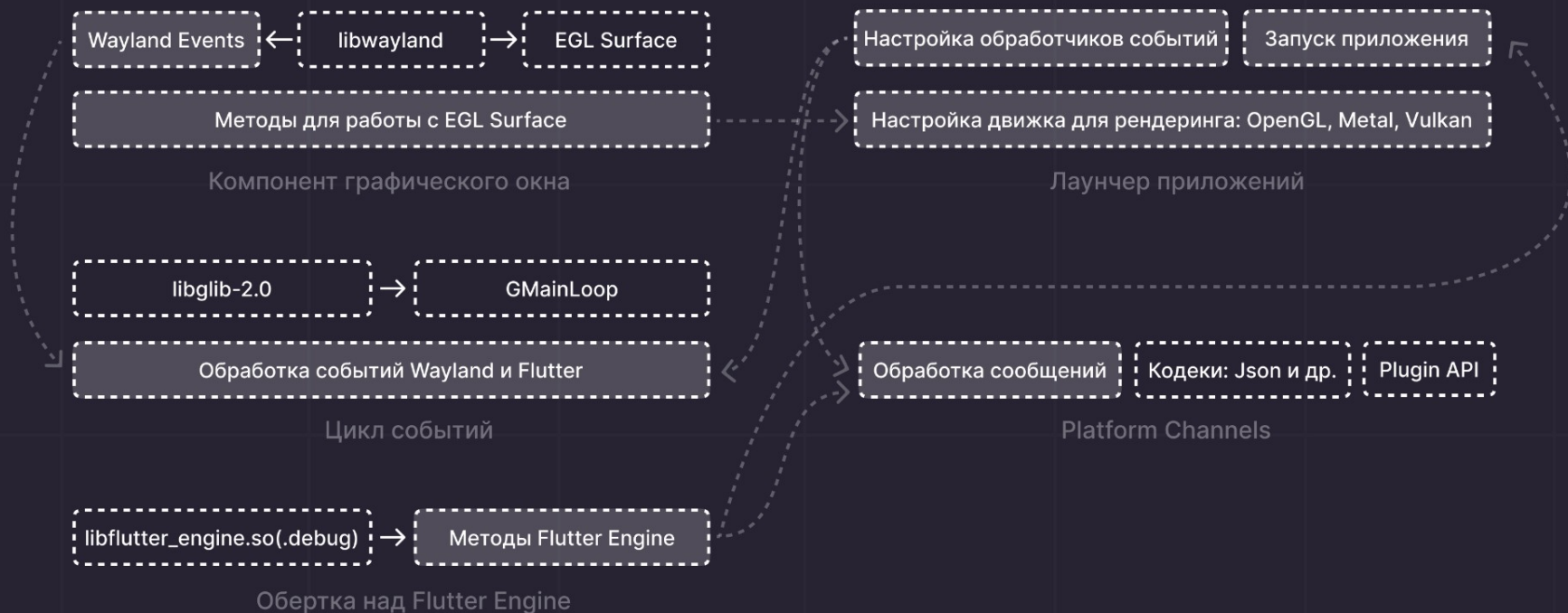
UI библиотека на языке Dart

Flutter Engine

Рантайм для dart кода + IO

Flutter Embedder

Компонент для запуска приложения



Flutter слишком быстро развивается

- **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 не может быть основным
способом написания приложений для
ОС Аврора



~~android~~ ~~ios~~ ~~macos~~ ~~linux~~ ~~windows~~

Много платформозависимых плагинов

sqflite

PLATFORM | ANDROID IOS MACOS

3861
LIKES

140
PUB POINTS

100%
POPULARITY

path_provider

PLATFORM | ANDROID IOS MACOS LINUX

3769
LIKES

140
PUB POINTS

100%
POPULARITY

shared_preferences

PLATFORM | ANDROID IOS MACOS LINUX

7324
LIKES

140
PUB POINTS

100%
POPULARITY

sqflite

PLATFORM | ANDROID IOS MACOS AURORA

3861
LIKES

140
PUB POINTS

100%
POPULARITY

path_provider

PLATFORM | ANDROID IOS MACOS LINUX AURORA

3769
LIKES

140
PUB POINTS

100%
POPULARITY

shared_preferences

PLATFORM | ANDROID IOS MACOS LINUX AURORA

7324
LIKES

140
PUB POINTS

100%
POPULARITY

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

cam

PLATFORM

1656
LIKES

Roadmap по портированию Flutter



Подводные камни при портировании Flutter на ОС Аврора

Взять готовый Flutter Embedder для Linux

~~Взять готовый Flutter Embedder для Linux~~

Первая версия Flutter Embedder для ОС Аврора

Была написана на чистом Си

~~Первая версия Flutter Embedder для ОС Аврора~~

Была написана на чистом Си

Вторая версия Flutter Embedder для ОС Аврора

Была написана на C++ с поддержкой Platform Channels

sqlite

PLATFORM | **ANDROID** | IOS | MACOS

3861
LIKES

140
PUB POINTS

100%
POPULARITY

Но тут появился он...

~~Вторая версия Flutter Embedder для ОС Аврора~~

Была написана на C++ с поддержкой Platform Channels

Третья версия Flutter Embedder для ОС Аврора

Написана на C++ и вроде все работает ...

Разработка на Flutter под ОС Аврора



Программа бета-тестирования ОС Аврора





```
$ 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

application/aurora



desktop



icons



flutter



rpm



main.cpp



CMakeLists.txt

application/aurora



desktop



icons



flutter



rpm



main.cpp



CMakeLists.txt



```
[Desktop Entry]
Type=Application
Name=Flutter Example
Comment=The Flutter example project.
Icon=com.example.application
Exec=/usr/bin/com.example.application
X-Nemo-Application-Type=silica-qt5

[X-Application]
Permissions=UserDirs
OrganizationName=com.example
ApplicationName=application
```



```
[Desktop Entry]
Type=Application
Name=Flutter Example
Comment=The Flutter example project.
Icon=com.example.application
Exec=/usr/bin/com.example.application
X-Nemo-Application-Type=silica-qt5

[X-Application]
Permissions=UserDirs
OrganizationName=com.example
ApplicationName=application
```



```
[Desktop Entry]
Type=Application
Name=Flutter Example
Comment=The Flutter example project.
Icon=com.example.application
Exec=/usr/bin/com.example.application
X-Nemo-Application-Type=silica-qt5

[X-Application]
Permissions=UserDirs
OrganizationName=com.example
ApplicationName=application
```



```
[Desktop Entry]
Type=Application
Name=Flutter Example
Comment=The Flutter example project.
Icon=com.example.application
Exec=/usr/bin/com.example.application
X-Nemo-Application-Type=silica-qt5

[X-Application]
Permissions=UserDirs
OrganizationName=com.example
ApplicationName=application
```

application/aurora/icons



86x86.png



108x108.png



128x128.png



172x172.png



```
$ 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.
```

Что с поддержкой Platform Channel
плагинов?



```
$ 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);
}

```



```
void Plugin::onGetPlatformVersion(const MethodCall &call)
{
    utsname uname_data{};

    if (uname(&uname_data) == -1) {
        call.SendErrorResponse("uname-error", "uname failed", nullptr);
        return;
    }

    std::string preamble = "Aurora (Linux): ";
    std::string version = preamble + uname_data.version;

    call.SendSuccessResponse(version);
}

void Plugin::unimplemented(const MethodCall &call)
{
    call.SendSuccessResponse(nullptr);
}
```



```
void Plugin::onGetPlatformVersion(const MethodCall &call)
{
    utsname uname_data{};

    if (uname(&uname_data) == -1) {
        call.SendErrorResponse("uname-error", "uname failed", nullptr);
        return;
    }

    std::string preamble = "Aurora (Linux): ";
    std::string version = preamble + uname_data.version;

    call.SendSuccessResponse(version);
}

void Plugin::unimplemented(const MethodCall &call)
{
    call.SendSuccessResponse(nullptr);
}
```



```
void Plugin::onGetPlatformVersion(const MethodCall &call)
{
    utsname uname_data{};

    if (uname(&uname_data) == -1) {
        call.SendErrorResponse("uname-error", "uname failed", nullptr);
        return;
    }

    std::string preamble = "Aurora (Linux): ";
    std::string version = preamble + uname_data.version;

    call.SendSuccessResponse(version);
}

void Plugin::unimplemented(const MethodCall &call)
{
    call.SendSuccessResponse(nullptr);
}
```

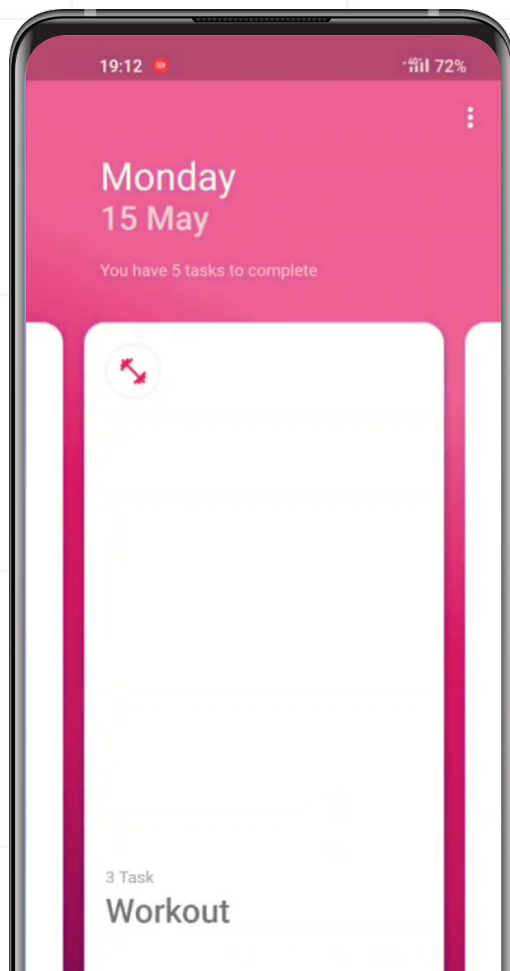


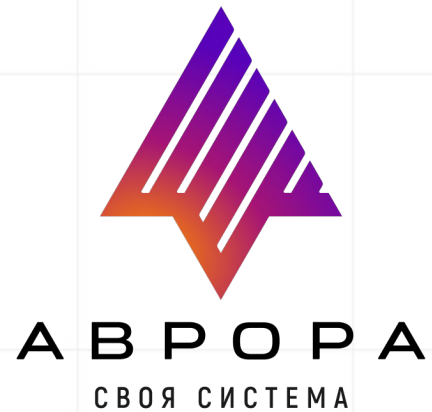
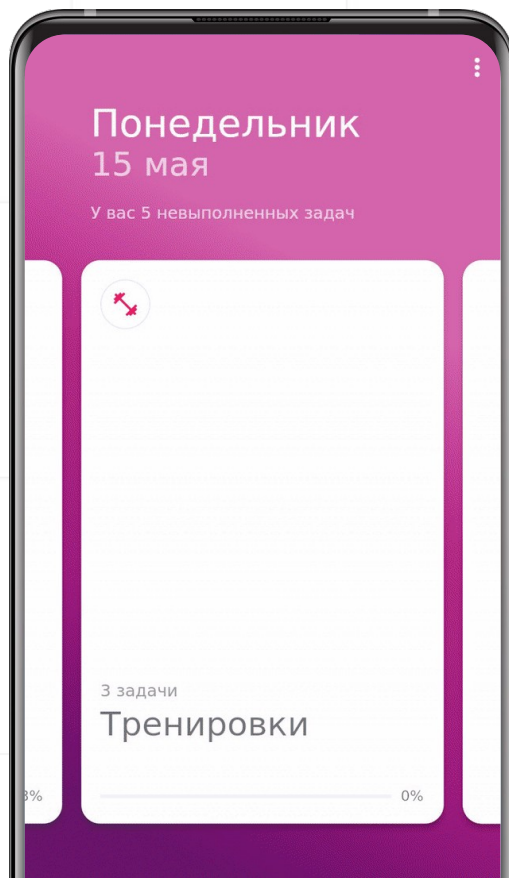
```
void Plugin::onGetPlatformVersion(const MethodCall &call)
{
    const auto args = call.GetArguments();
    const bool isRequired = args["isRequired"].GetBoolean();

    ...
}
```

Что с поддержкой FFI плагинов?

Какие Flutter приложения под ОС Аврора
есть на сегодняшний день?





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**

Будущие планы







А В Р О Р А
СВОЯ СИСТЕМА



Денис Глазков

Открытая Мобильная Платформа

 glazzkoff

 d.glazkov@omp.ru

**Спасибо за
внимание**



Mobius
2023 Spring



ОТКРЫТАЯ
МОБИЛЬНАЯ
ПЛАТФОРМА