



# ЗНАКОМСТВО С ОПЕРАЦИОННОЙ СИСТЕМОЙ АВРОРА

# ОС Аврора

♪ Российская операционная система

♪ Для смартфонов и планшетов

♪ POSIX-совместимый дистрибутив Linux

♪ Встроенные СКЗИ

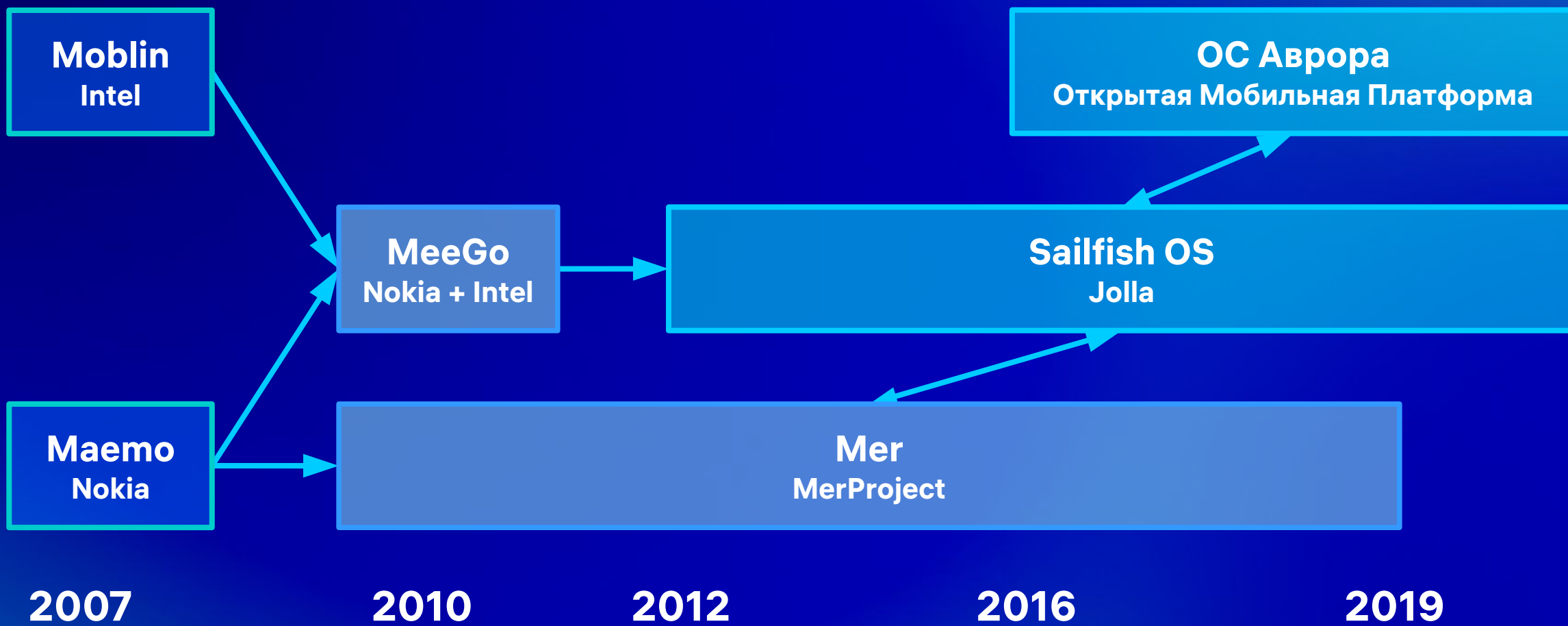
♪ Сервис PUSH-уведомлений

♪ Поддержка корпоративных политик управления



А В Р О Р А

# История ОС Аврора



# Особенности ОС Аврора

## ♪ Полноценный GNU/LINUX

♪ POSIX-совместимая ОС

♪ Система пакетов RPM

♪ Режим разработчика: bash, terminal, ssh и др.

## ♪ Разработка

♪ Фреймворк Qt

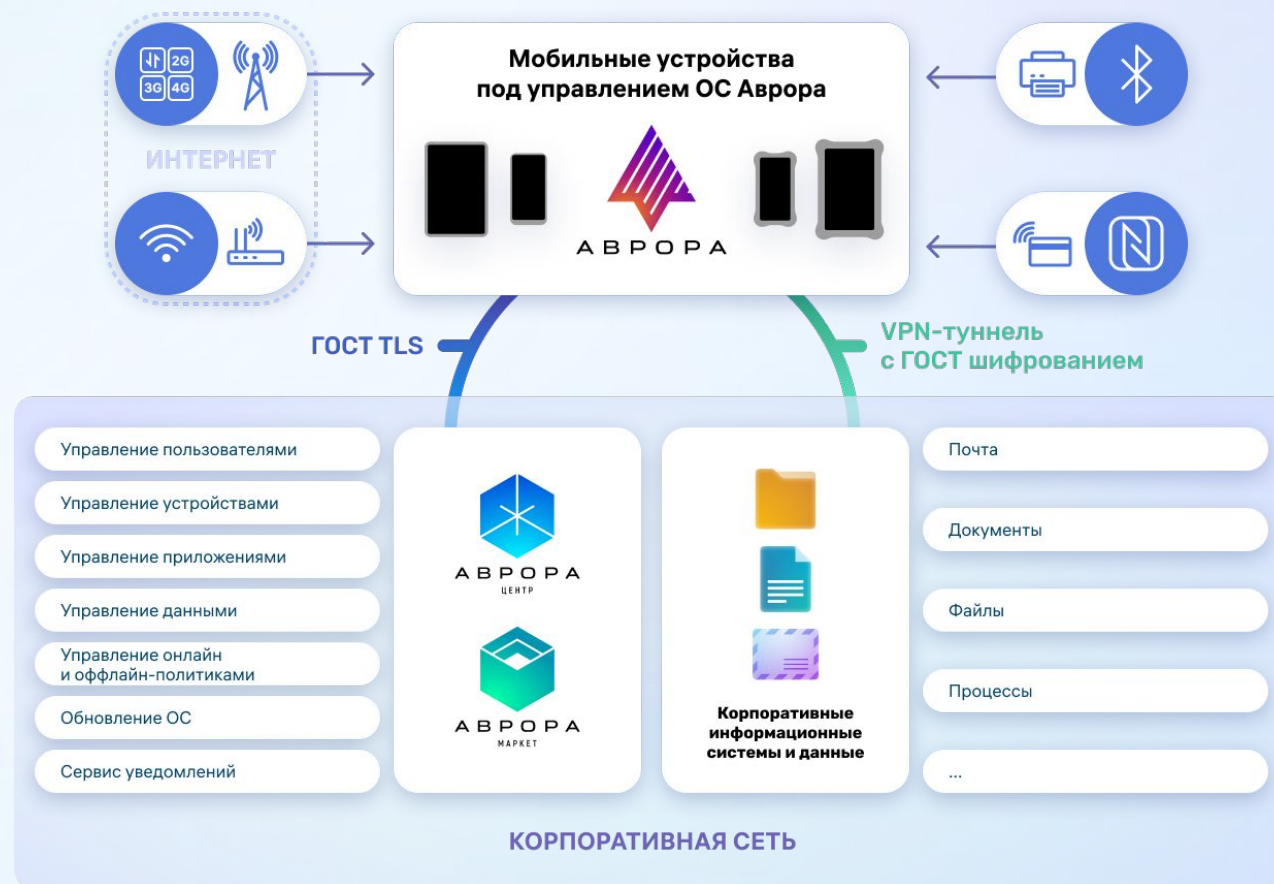
♪ IDE на основе Qt Creator

♪ Библиотеки C++ и JS (QML)

♪ D-Bus

# Аврора — корпоративная ОС

- Доверенные мобильные устройства
- Сертифицированные VPN каналы
- Платформа управления Аврора Центр
  - Контроль функций устройств
  - Политики безопасности
  - Управление приложениями и обновлениями
- PUSH-сервис



# Устройства

## Устройства:

Смартфоны: Qtech, TrustPhone, ...

Планшеты: Aquarius, БайтЭрг, ...

Продажа мобильных устройств осуществляется только корпоративным и государственным заказчикам

# Архитектура ОС Аврора

## Базовый уровень

♪ Ядро

♪ Аппаратно-  
зависимые  
компоненты

♪ Board Support Package  
(BSP)

♪ Hardware Abstraction  
Layer (HAL)

♪ Прошивка

## Уровень связующего ПО

♪ Mer

♪ Окружение Qt

♪ Firejail

♪ Integrityd

♪ Другие связующие  
библиотеки  
и сервисы

## Уровень приложений и UI

♪ Подсистема Lipstick  
и управление UI

♪ Домашний экран, сетка  
приложений, лента  
событий, блокировка

♪ Навигация свайпами

♪ Уведомления

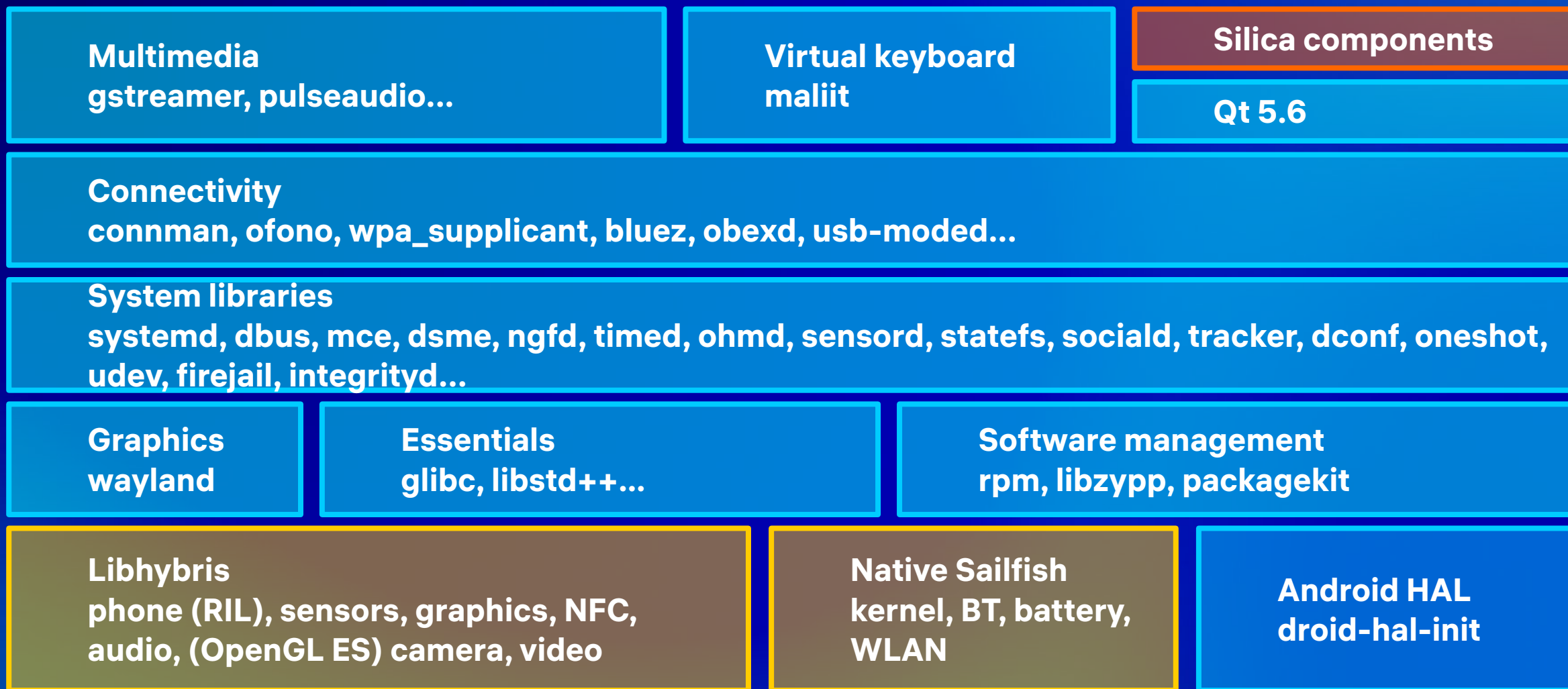
♪ Атмосферы

♪ Окна приложений

♪ Приложения



# Составляющие архитектуры





# Состав Аврора SDK

✧ IDE на основе Qt Creator

✧ Система сборки с кроссплатформенными инструментами

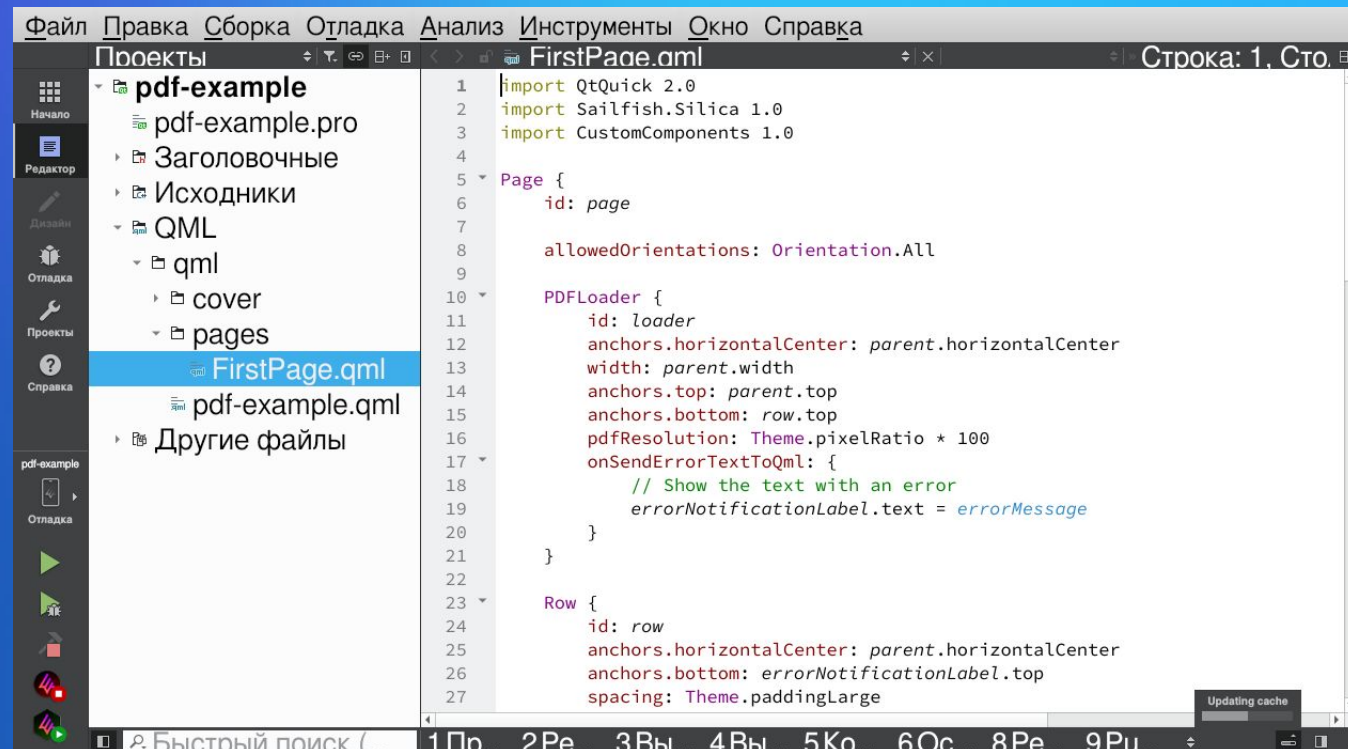
✧ Эмулятор с образом ОС

✧ Подключение  
к внешним устройствам

✧ Валидатор установочных  
пакетов

✧ Инструменты  
для подписи пакетов

✧ Интегрированная  
документация и примеры



# Режимы IDE

𐄂 Начало: недавние проекты, примеры, учебники

𐄂 Редактор: работа с исходным кодом

𐄂 Отладка: дополнительные панели

    𐄂 Текущий шаг

    𐄂 Значения переменных

𐄂 Проекты: настройки сборки и запуска

𐄂 Справка: документация



# Начало работы с проектом

Файлы проекта должны быть в домашней директории или директории, указанной при установке SDK

👤 Приложение из базового шаблона

Начало → Проекты → Новый проект

👤 Открытие существующего проекта

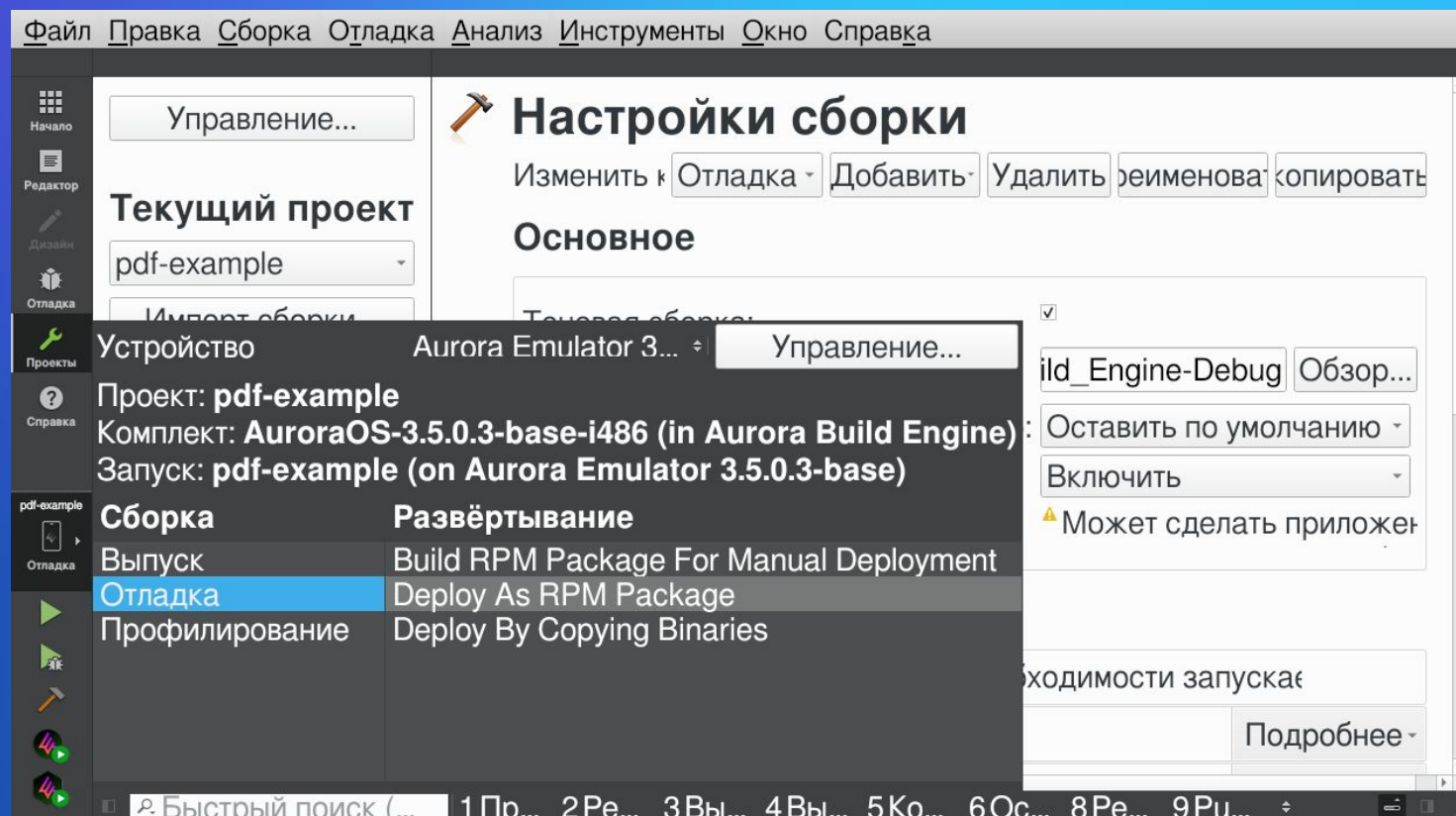
Меню → Файл → Открыть файл или проект

# Создание приложения

- 1 «Файл» → «Создать файл или проект...».
- 2 В окне «Новый файл или проект» выбрать «Проекты» → «Приложение» и отметить «ОС Аврора SDK Qt Quick Application»
- 3 Указать название организации, имя проекта, директорию  
✧ проект должен находиться или в домашней директории пользователя, или в альтернативной директории, указанной при установке Аврора SDK
- 4 Указать разрешения, которые необходимы приложению
- 5 В качестве системы сборки можно оставить qmake
- 6 В окне «Выбор комплекта» указать комплекты для сборки  
✧ arm7hl используется для мобильных устройств, i486 — для эмулятора
- 7 В окне «Управление проектом» выбрать систему контроля версий и нажать кнопку «Завершить»

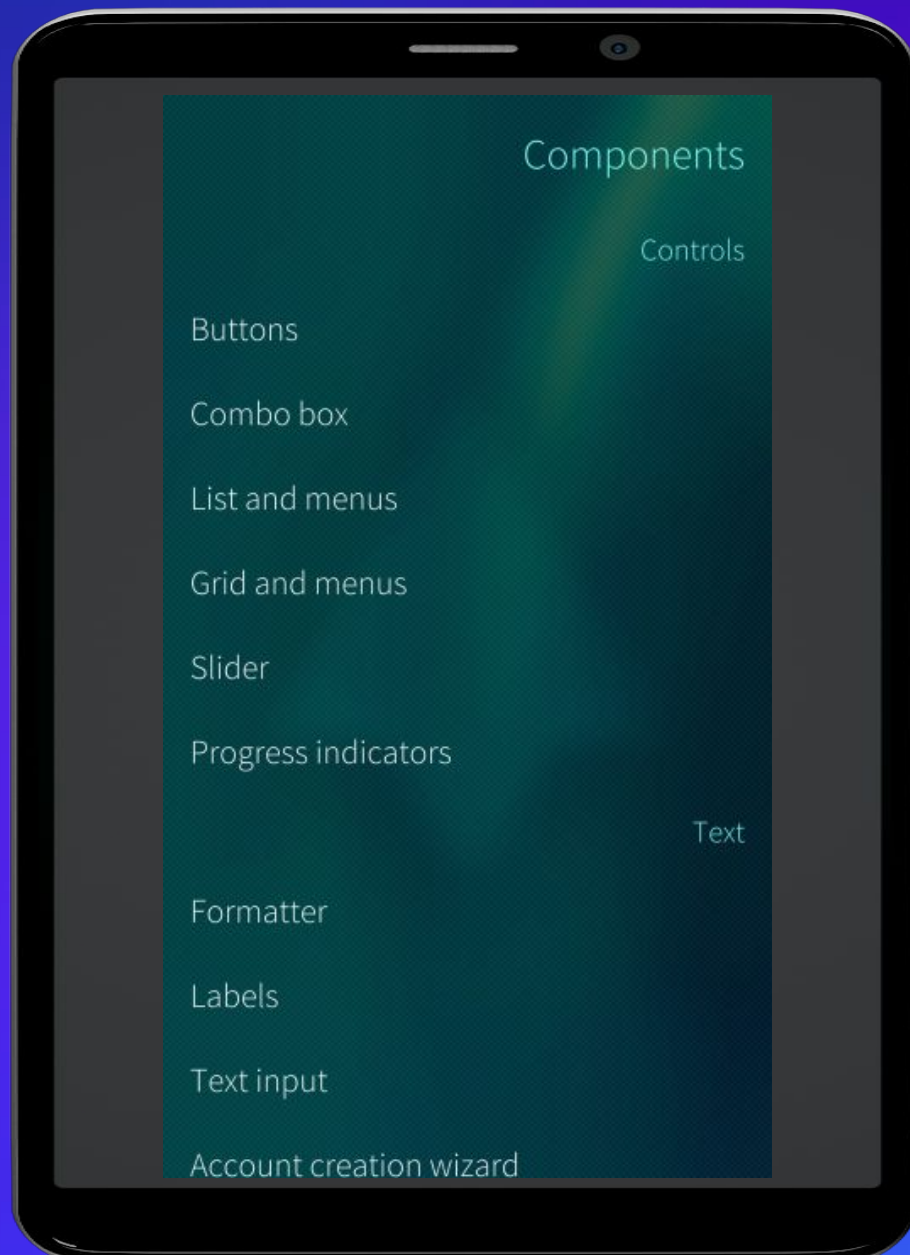
# Сборка проекта

1. Выбрать комплект
  - 妹 Целевая платформа
  - 妹 Целевая архитектура
2. Выбрать тип сборки
  - 妹 Выпуск
  - 妹 Отладка
  - 妹 Профилирование
3. Запустить сборку
  - 妹 Нажать значок сборки  
внизу слева
  - 妹 Ctrl + B



# Запуск приложения

1. Выбрать целевое устройство или эмулятор
2. Выбрать способ установки
  - Собрать grm-пакет
  - Скопировать файлы
3. Настроить этап подписи
4. Собрать и запустить
  - Обычное исполнение: Ctrl + R
  - Режим отладки





# Архитектура GUI-приложений

♪ Декларативное описание UI с помощью QML

♪ Стандартный модуль QtQuick

♪ Специфичный модуль Silica

♪ Доступ к некоторым API

♪ «Лёгкая» логика на «JavaScript»

♪ Запуск с помощью исполняемого кода на C++

♪ Основной файл QML встраивается в QQuickView

♪ Доступ к библиотекам Qt и C++

♪ Низкоуровневый доступ к API устройства

♪ «Тяжёлые» вычисления

# Директория проекта приложений

- 妳 **<имя проекта>.pro** — описание структуры, модули
- 妳 **<имя проекта>.pro.user** — локальные настройки SDK (не для передачи)
- 妳 **src** — исходный код на C++
- 妳 **qml** — исходный код на QML и ресурсы UI
- 妳 **icons** — иконки для значка приложения
- 妳 **rpm/<имя проекта>.spec** — параметры сборки rpm-пакета
- 妳 **rpm/<имя проекта>.yaml** — для автогенерации spec
- 妳 **<имя проекта>.desktop** — параметры запуска приложения
- 妳 **translations** — файлы переводов



# Файл .desktop

♪ Основная структура:

♪ Type=Application

♪ X-Nemo-Application-Type=silica-qt5

♪ Icon=<название\_пакета>

♪ Exec=<название\_пакета>

♪ Name=<название\_приложения\_на\_английском\_языке>

♪ Name[ru]=<название\_приложения\_на\_русском\_языке>

♪ X-Nemo-Application-Type=silica-qt5 — приложение использует Silica

♪ Строка Exec=<название\_пакета>

♪ Name определяет название приложения на экране приложений

♪ Локализация названия: Name[<двухбуквенный\_код\_языка>]

# Управление правами приложения

- ♪ Изоляция — механизм безопасного исполнения приложений в песочнице
- ♪ Система ограничений прав приложений применима только к приложениям, запускаемым в песочнице
  - ♪ В дальнейшем изоляция будет применяться ко всем запускаемым приложениям
- ♪ Разрешения или права, необходимые для работы приложения указываются в desktop-файле
- ♪ В desktop-файле указывается команда запуска приложения
- ♪ Для запуска в песочнице Аврора OS приложение должно соответствовать правилам

# Секция X-Application

- Раздел со списком прав
  - `Permissions` — список идентификаторов требуемых прав
  - `OrganizationName` — название организации-разработчика
  - `ApplicationName` — имя директории, выделенной для хранения данных приложения
- Разрешения предоставляют доступ к директориям, интерфейсам D-Bus, сокетам и приложениям
- Приложения должны определять все необходимые разрешения в desktop-файле
  - Они предоставляются в момент запуска
- `[X-Application]`  
`Permissions=Internet;Pictures`  
`OrganizationName=org.foobar`  
`ApplicationName=myapp`

# Доступные директории

✧ OrganizationName и ApplicationName используются для предоставления приложению прав на запись в директории

✧ \$HOME/.local/share/\<Название организации\>/\<Имя приложения\>

✧ \$HOME/.cache/\<Название организации\>/\<Имя приложения\>

✧ \$HOME/.config/\<Название организации\>/\<Имя приложения\>

✧ Для доступа к каталогам из приложения используйте QStandardPaths

✧ Данные приложения - QStandardPaths::AppDataLocation

✧ Кеш приложения - QStandardPaths::CacheLocation

✧ Конфигурация приложения - QStandardPaths::ConfigLocation

✧ Только эти каталоги могут использоваться для хранения данных



# Основной объект приложения Qt

✧ Измените основной объект приложения Qt, установив организацию и имя приложения

```
✧ QScopedPointer<QGuiApplication>  
    app(Sailfish::createApplication(argc, argv));  
✧ app->setOrganizationName(QStringLiteral("org.foobar"));  
✧ app->setApplicationName(QStringLiteral("myapp"));
```

✧ В будущем это будет делать автоматически библиотека `libsailfishapp`

# Разрешения для приложений

## 👤 Функции мобильного устройства:

👤 Технология Bluetooth: Bluetooth

👤 Получение координат: Location

👤 Технология NFC: NFC

## 👤 Сеть:

👤 Управление сетевыми подключениями: Internet

👤 Отображение WEB-контента: WebView

👤 Передача данных: Internet

## 👤 Мультимедиа:

👤 Проигрывание и запись аудио и видео: Audio, Microphone

👤 Низкоуровневые функции работы аудио: Audio, Microphone

👤 Съемка фотографий и видео: Camera

# Разрешения для приложений

✧ Доступ к системным службам:

✧ MediaIndexing — Получение списка файлов на устройстве

✧ PushNotifications — Push-уведомления

✧ SecureStorage — Безопасное хранение данных

✧ Для профилей Extended, Antivirus и MDM также возможно использование разрешения LogSecurityEvents

# Разрешения для приложений

♪ Доступ к дополнительным директориям:

♪ RemovableMedia

♪ UserDirs даёт доступ ко всем директориям из списка ниже

♪ Documents — Доступ к директории документов

♪ Downloads — Доступ к директории загрузок

♪ Music — Доступ к директории Music, плейлистам и изображениям обложек

♪ Pictures — Доступ к директории изображений и эскизам

♪ PublicDir — Доступ к директории ~/Public

♪ Videos — Доступ к директории видео и эскизам



# Формат метаданных разрешений

♪ Метаданные используются для генерации переводов во время сборки и отображения информации о разрешениях в UI

♪ В комментариях к файлам разрешений есть метаданные

♪ Строка метаданных начинается с символа комментария #

♪ Строка содержит ключ, который начинается с x-sailjail-

♪ За ключом следует '=' и его значение

♪ Ключи:

♪ X-sailjail-translation-catalog — имя каталога переводов Qt

♪ x-sailjail-translation-key-description — ключ перевода для краткого описания

♪ x-sailjail-description — технический английский текст для краткого описания

♪ x-sailjail-translation-key-long-description — ключ перевода для длинного описания

♪ x-sailjail-long-description — технический английский текст для подробного описания

# Установочные пакеты

♪ ПО распространяется в виде rpm-файлов

♪ Состав rpm-пакета

♪ Исполняемый файл приложения

♪ Иконки и desktop-файлы

♪ Подключаемые библиотеки

♪ Ресурсы

♪ Зависимости

♪ Скрипты установки и удаления

♪ Подписи

# Сборка и запуск проекта

- 1 Конфигурация проекта в pro-файле
- 2 Настройка проекта в yamI-файле
- 3 Генерация spres-файла на основе yamI-файла
- 4 Сборка grm на основе spres-файла
- 5 Подпись grm-пакета
- 6 Валидация grm-пакета
- 7 Установка grm-пакета



# Файл проекта для ОС Аврора

- **{имя проекта}.pro** — описание проекта для qmake
- **TARGET** — название исполняемого файла и директоии
- **CONFIG** — дополнительные настройки сборки
  - sailfishapp — для Аврора ОС
  - sailfishapp\_i18n — активация переводов
- **TRANSLATIONS** — файлы переводов
- **DISTFILES** — QML и другие файлы для проекта
- **SAILFISHAPP\_ICONS** — разрешения иконок приложения



# Тип проекта и библиотеки

- **TEMPLATE**
  - app (по умолчанию) — приложение
  - lib — библиотека
  - aux — ничего не строить
  - subdirs — собрать несколько проектов (указываются в **SUBDIRS**)
- Использование библиотеки
  - **LIBS** += -L/usr/local/lib -lmath
- **INCLUDEPATH** — дополнительные пути для заголовков

# Дополнительные возможности qmake

- Определения

```
DEFINES += Q_OS_SAILFISH
```

```
contains(DEFINES, Q_OS_SAILFISH):message(sailfish os)
```

- Параметры сборки

```
CONFIG(release, debug|release):message(release build)
```

```
CONFIG(debug, debug|release):message(debug build)
```

- Дополнительные файлы для установки

```
documentation.files = docs/*
```

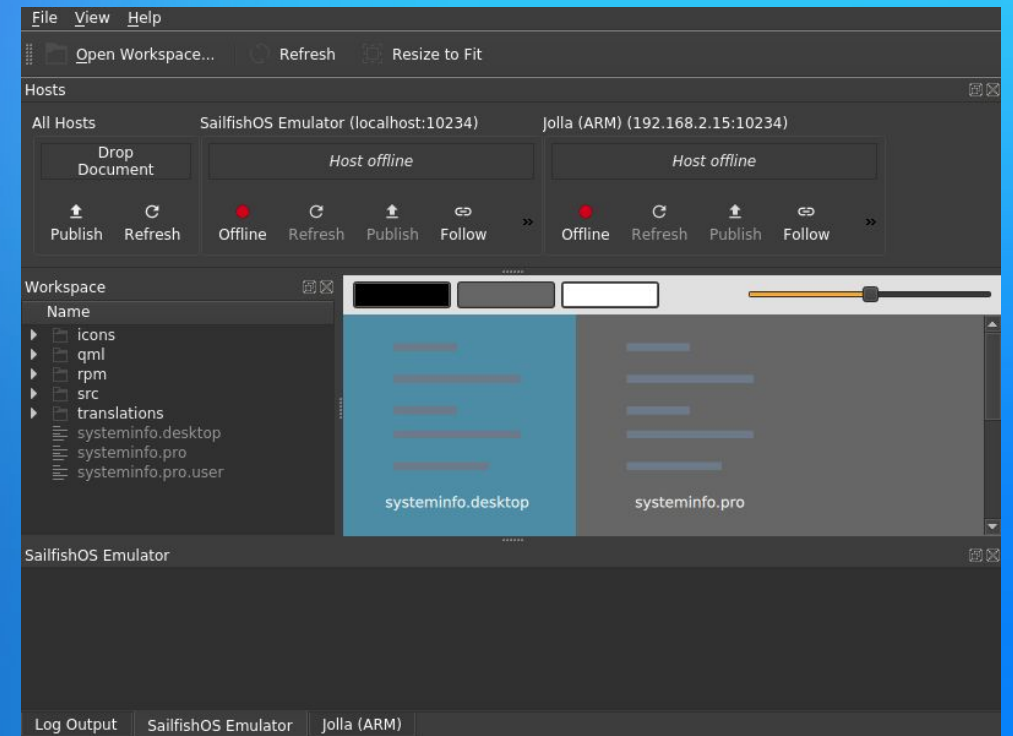
```
documentation.path = /usr/local/program/doc
```

```
INSTALLS += documentation
```

# Qt QmlLive

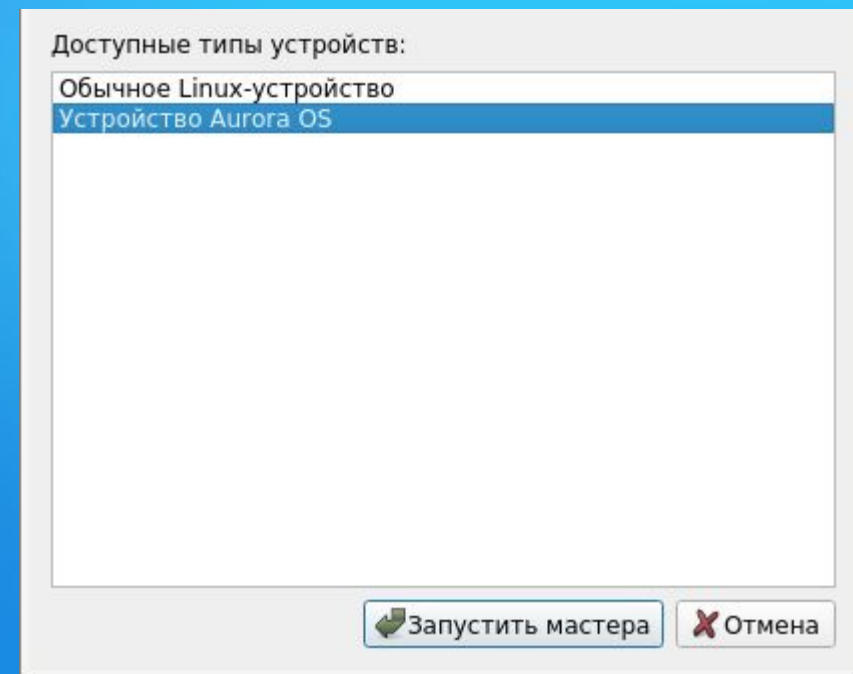
- 1 Активировать Qt QmlLive для проекта  
**Проекты → <цель> → Запуск →  
Enable receiving updates with Qt  
QmlLive**
- 2 Запустить приложение
- 3 Сохранить изменения QML-файла

При использовании на устройстве  
нужен пакет qmlive-sailfish



# Подключение внешнего устройства

- 1 **Инструменты → Параметры → Устройства → Добавить Устройство Аврора ОС**
- 2 Пароль задаётся на устройстве  
**Настройки → Средства разработчика**
- 3 И с эмулятором, и с устройствами взаимодействие происходит по SSH





# Работа со средой сборки из консоли

```
ssh -p 2222 -i ~/AuroraOS/vmshare/ssh/private_keys/  
engine/mersdk mersdk@localhost
```

妹 Доступ к директориям ПК

妹 Домашняя директория: /home/mersdk/share

妹 Директория, указанная при установке: /home/src1

妹 Установка пакета

```
sb2 -t <target> -m sdk-install -R rpm -i <rpm-файл>
```

妹 Сборка пакета

```
mb2 -t <target> build
```

**<target>**: AuroraOS-armv7hl или AuroraOS-i486



# Работа с эмулятором из консоли

- 1 Скопировать пакет:

```
scp -P 2223 -i
```

```
~/AuroraOS/vmshare/ssh/private_keys/AuroraOS_Emulator/<user>  
./RPMS/<package>.rpm <user>@localhost:
```

- 2 Авторизоваться:

```
ssh defaultuser@device
```

- 3 Установить пакет:

```
rpm -i *.i486.rpm
```

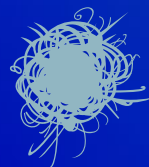


# Ресурсы в интернете

👩 Портал сообщества разработчиков  
[community.omprussia.ru](https://community.omprussia.ru)

👩 Материалы для разработчиков  
[auroraos.ru/developer](https://auroraos.ru/developer)

👩 Сервисная поддержка  
разработчиков партнёров  
[dev-support@omp.ru](mailto:dev-support@omp.ru)



[habr.com/ru/company/omprussia](https://habr.com/ru/company/omprussia)



[stepik.org/users79671378](https://stepik.org/users79671378)



[www.youtube.com/c/AuroraOSCommunity](https://www.youtube.com/c/AuroraOSCommunity)



[www.youtube.com/cAuroraOS](https://www.youtube.com/cAuroraOS)



[t.me/omp\\_ru](https://t.me/omp_ru)



[vk.com/omp\\_ru](https://vk.com/omp_ru)

edu@omp.ru

