

Konquering the Droids

Porting KDE apps to Android



4 - 11 September 2020

The year of the Linux Desktop

KDE's Mission



4 - 11 September 2020

“A world in which everyone has control over their digital life and enjoys freedom and privacy.”

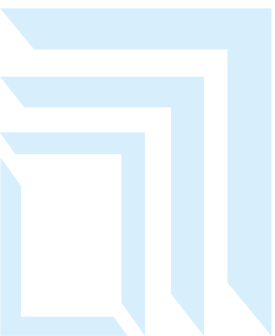
- Our digital lives increasingly take place on mobile devices
- If we don't provide solutions there we are failing our mission
- Plasma Mobile is our vision
- Creating solutions for Android is a stepping stone

Agenda

- Why do we care about Android?
- Where are we now?
- How hard is it?
- How is it done?
- How can we improve it?



4 - 11 September 2020



Why do we care about Android

- More than one billion users
- Including a lot of us
- Enable eat-your-own-dogfood testing
- Grow the number of potential contributors



4 - 11 September 2020



Where are we now?

- 30 apps on <https://binary-factory.kde.org/view/Android>
- 7 apps on Google Play
- 3 apps on F-Droid



4 - 11 September 2020



How hard is it?

- It depends ;-)
- Things that make it hard
 - Mobile-unfriendly design
 - QtWidgets
 - DBus
 - glibc-isms
 - Exotic build systems
 - Multi-process architectures



4 - 11 September 2020

How is it done?

- Just call cmake with the right arguments ;-)
- Build steps
 - Get Android SDK and NDK
 - Get Qt for Android
 - Build all dependencies for Android
 - Build your app for Android
 - Build APK using androiddeployqt



4 - 11 September 2020

How does KDE help with that?



4 - 11 September 2020

- ECM provides an Android toolchain
`cmake -DCMAKE_TOOLCHAIN_FILE=/path/to/ECM/toolchain/Android.cmake -DCMAKE_PREFIX_PATH=/path/to/qt/android`
- Wraps `androiddeployqt` calls into 'make create-apk'
- Ready-made docker container
https://community.kde.org/Android/Environment_via_Container
- KDE Frameworks abstract things
- Check out Volker's talk: <https://www.youtube.com/watch?v=bleX5HzF11E>

Steps for new apps

- Add AndroidManifest.xml
 - Copy from some project
 - Replace names
 - android.app.lib_name must match executable name
 - Usually in android/ subfolder
- Export main() method: `Q_DECL_EXPORT int main()`
- Add app icon in android/res/drawable



4 - 11 September 2020

Where are my icons?



4 - 11 September 2020

- To load icons from the icon theme we need to bundle them
- Done with `kirigami_package_breeze_icons` cmake macro

```
kirigami_package_breeze_icons(ICONS
    document-open
    ...
    help-about
)
```

- `target_link_libraries(yourapp Qt5::Svg)` to bundle it

What else needs to be done?



4 - 11 September 2020

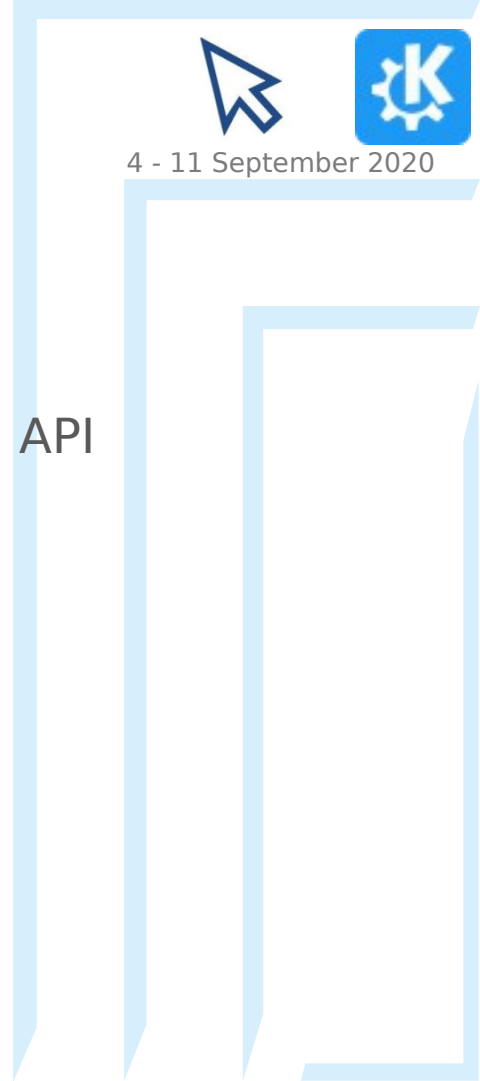
- Plugin systems need special care
 - Needs libname-android-dependencies.xml file
 - <https://invent.kde.org/frameworks/knotifications/-/blob/master/src/CMakeLists.txt#L243>
 - <https://invent.kde.org/frameworks/kpeople/-/commit/2ce4d61647972c61100ae0dbd06c328d9d9901da>
- Check your file paths!
 - Consider bundling things in qrc
 - `QStandardPaths::GenericDataLocation` is not accessible without permission

What about Android-specific code?

- Use `if(ANDROID)` or `if(NOT ANDROID)` in `cmake`
- Use `#if Q_OS_ANDROID` in C++ code
- Use classes from `QtAndroidExtras` where possible
- Use JNI wrappers from `QtAndroidExtras` to call Android API
- Consider upstreaming solutions to KF5



4 - 11 September 2020



Where can I learn more?

- The code is documentation enough ;-)

- Study

<https://invent.kde.org/sysadmin/ci-tooling/-/tree/master/system-images/android/sdk>

- Consult Qt documentation

- <https://community.kde.org/Android>

- #kde-android:kde.org on Matrix



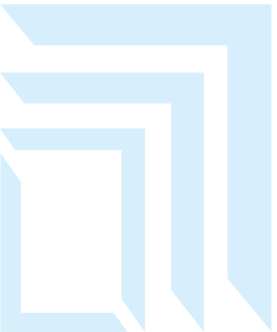
4 - 11 September 2020

Can we improve this?

- The whole process can be tricky
- We need better documentation and tooling
- Come to the BoF



4 - 11 September 2020





4 - 11 September 2020

Thank you for your attention!