Android Layers in Sailfish OS

This page is mostly based on DrYak's explanations in the forum.

Sailfish OS can be seen in multiple layers of software:

- at the bottom, the manufacturer of the chipset (e.g. Qualcomm) usually only produces drivers that are designed to run in Android (including some of the unusual libraries and interfaces that Android runs unlike other Linux-based OS)
- on top of the hardware, SailfishOS is a relatively standard GNU/Linux distribution (audio handled with PulseAudio, graphics with Waydroid, etc. has a standard C library)
- Jolla uses a special compatibility layer (libhybris) that makes the Android-only driver usable in the classic Linux distro such as SailfishOS.
- on top of the OS, you can run native Linux apps, mostly written using Qt and QML graphics.

To run native applications, no Android is required.

To run Android apps

- Jolla provides a compatibility layer called Android App Support. It's an LXC container (mostly like Docker containers) that you can optionally download and install from Jolla's application store.
- Once the container is started (either by autostart or manually), AOSP runs inside it (only Open-Source Android, not the proprietary parts including Google Play).
  - this makes some Android apps possible to run. You'll most probably rely on F-Droid to download (open source) Android apps.
- optionally you can install microG, it's an opensouce re-implementation of â€œGoogle Play Serviceâ€ the API that are normally provided by the proprietary Google Bit.
  - this makes a lot more apps possible to run. You can use the open-source Aurora (installable from F-Droid) to download and install apps from Google's own Store.
  - some apps might still refuse to run due to some specific things that the original proprietary Google Play Service provides but which are missing in microG (e.g., some Banking apps might complain about a rooted device because they want a full Google SafetyNet remote attestation).
- optionally, you can also install it instead of the original Google Play Service.
  - this kind of defies your de-googling attempts.
  - but this makes the stubborn e-banking app workable
  - also, you don't need to constantly run the container. You can keep the container not running, and only run the container for when you need *that one* Android app you can't find a replacement for

A big difference between a stock Android OS, and Android App Support is that you get more options to control what is running in the phone (even more if you opt to use microG instead of Google's own shit).
e.g.:

- you can manually start Android App Support yourself (and only run native SailfishOS apps the rest of the time).
- you can configure Android App Support to not start applications in the background on startup.
- you can configure microG to not automatically start an application when that gets a message/alert from the cloud servers.

Also:

- the AOSP running inside the container cannot see and use the hardware of the phone. For technical reasons, it needs special supports that expose stuff to Android.
- e.g., you can plug in a USB keyboard or pair a Bluetooth keyboard, and Android will see an available â€œinput deviceâ€.
- e.g.: you can pair Bluetooth speakers, and Android will see an available â€œaudio outâ€
- However, there is currently no support for Bluetooth itself, so you cannot use Android apps that need to directly talk to a Bluetooth device (you cannot use the speaker manufacturer's app to flash a firmware update to the Bluetooth speaker).

On smartphones not officially supported by Jolla, there is the open-source Waydroid solution, which is pretty similar to Android App Support.

- The main differences are in what adapters each has (e.g.: Waydroid currently cannot open multiple windows on SailfishOS)

So TL;DR: You get a whole range of options between:

- only the hardware drivers are written by Qualcomm
- more or less support for apps with some containers and optional service replacement (and you can prevent apps from running in the background).
- full(-er -ish) compatibility by replicating a full Android, but at least you get to shut it down when not in use.

Revision #3
Created 28 September 2024 16:52:50 by sebix
Updated 12 November 2024 15:34:41 by sebix