



Linux
Plumbers
Conference | Richmond, VA | Nov. 13-15, 2023

Pixel 6 support on android-mainline

William McVicker (Google)
Peter Griffin (Linaro)





Linux
Plumbers
Conference | Richmond, VA | Nov. 13-15, 2023

Pixel 6 has been supported on
android-mainline for 2 years.





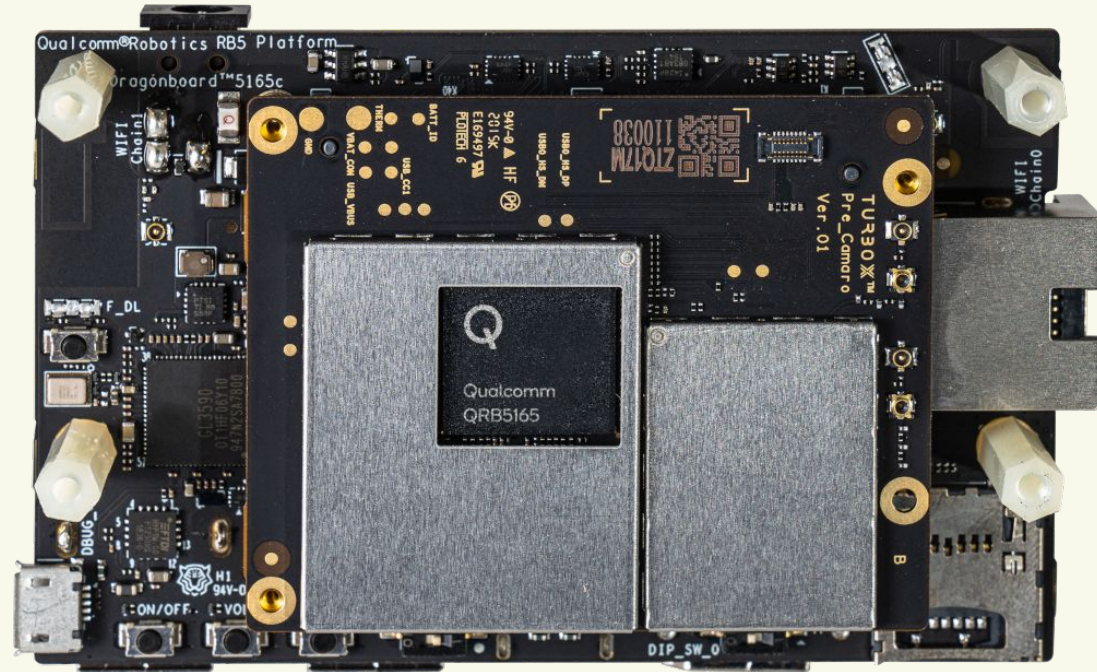
Overview of Pixel 6 support on android-mainline

- Goal to add testing of the GKI kernel with a real in-market Android device.
 - Catch regressions from upstream merges into android-mainline before they are propagated to Android vendors.
 - Increase Android specific testing on real in-market hardware.
 - **All** android-mainline kernel changes are tested on Pixel 6 before submission (in presubmit and postsubmit).
- Allows for kernel developers to test upstream patches from the list to evaluate performance impacts
 - This helps Android developers be more involved in upstream discussions.
- Allow for easier in-market kernel upgrades
 - It is easier to maintain functionality and performance incrementally for each Linux release, than to upgrade from one LTS release to the next.
 - Quicker bring-up of new devices when based off of android-mainline drivers vs launch kernel on older LTS branch.



Example upstream features developed and/or tested on Pixel 6 with android-mainline

- 16k support for f2fs
 - <https://lore.kernel.org/all/20231002230935.169229-1-drosen@google.com/>
- VM CPUFreq
 - <https://lore.kernel.org/all/20230731174613.4133167-1-davidai@google.com/>
- Proxy Execution
 - <https://lore.kernel.org/all/20230819060915.3001568-1-jstultz@google.com/>
- pKVM
- userfaultd move operation
 - <https://lore.kernel.org/all/20231009064230.2952396-1-surenb@google.com/>
- Binder development and testing
 - Rust implementation: <https://lore.kernel.org/all/ZUKaSD4sPtHzlqfT@google.com/>
- Multi-gen LRU (feature tested on android14-6.1)
 - <https://lore.kernel.org/all/20230413214326.2147568-1-kaleshsingh@google.com/>



Enable Android vendors to do kernel development on Pixel 6

- AOSP development boards often lack certain hardware found on real phones.
- Partners like Linaro often want to run latest AOSP with the latest Linux kernels.

The Pixel 6 android-mainline project is creating various public resources

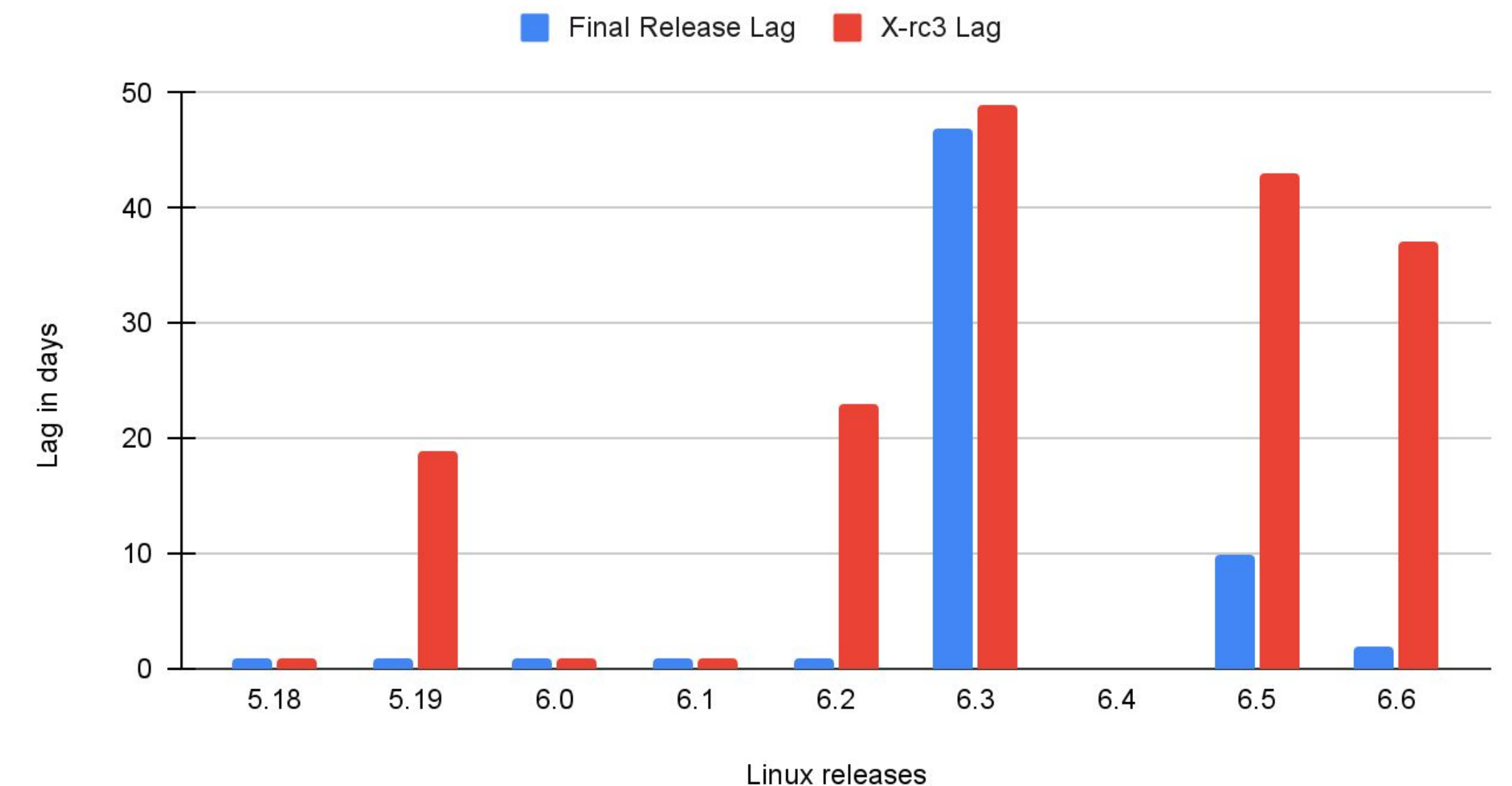
- Public documentation for building and flashing Pixel kernels.
 - <https://source.android.com/docs/setup/build/building-pixel-kernels>
- Periodically update the AOSP mainline Pixel 6 kernel repo.
 - <https://android.googlesource.com/kernel/manifest/+refs/heads/gs-android-gs-raviole-mainline>



android-mainline versus Linux release lag

- Often android-mainline releases merge within 1 day of Linus' Linux release.
- Looking at RC3 dates, the lag is much longer due to:
 - Forward porting Pixel 6 drivers.
 - Resolving conflicts with GKI technical debt.
- 16 upstream bugs found by testing Pixel 6 on android-mainline in CI!
- However often testing starts late in the RC stages. If Pixel 6 was upstream we could test and find regressions much earlier!

android-mainline Lag





Why upstream Pixel 6?

Pixel 6 on android-mainline has many benefits, but upstreaming the SoC and phone unlocks even more, including:

- Testing earlier in the Linux release candidate cycle with more CI systems (lkft, kernelci).
- Adding a phone form factor upstream kernel development platform!
- Reducing GKI technical debt due to an upstream user.

Initial Pixel 6 series under review on LKML:

<https://lore.kernel.org/linux-serial/ZScZPstpJInZxwID@google.com/T/#m8a08b45cbbf317c229982741cc2c5e0e208bf3ab>

Goal is to replace downstream Pixel 6 android-mainline drivers with their upstream counterparts.



DT overlays Problem Statement

Android phone models have many board revisions before mass production, e.g. EVT, DVT and PVT. Typically a 'board-id' is baked into the firmware to identify the particular board model and revision. A common base DTB is shared across all device variants with a board revision specific overlay applied at runtime by the bootloader.

- The 'board-id' overlay scheme is used by multiple Android vendors already, e.g. Qcom, Samsung, Pixel.
 - Android documentation for using multiple device tree overlays is found at <https://source.android.com/docs/core/architecture/dto/multiple>.
- Re-using the same DTB for each board variant saves space on the device and allows you to support more devices with the same dtb and dtbo images.
 - Pixel 6/6Pro/6a saves ~**6.8MB** by sharing the same DTB for each overlay.
$$\text{size_of_dtb} * (\# \text{ of variants} - 1)$$
- Using a numeric ID vs compatible string, reduces string comparison in the kernel
 - Supporting all these board variants via string compatibles would lead to an explosion of board compatibles)
 - For example, Pixel 6/6 Pro/6a and unreleased gs101 bring-up hardware has **29** different 'board-id'/'board-rev' variants.



Existing Implementations

Overlays are applied at build time and each supported board uses the overlayed DTB while DTBOs are reserved for plug-in daughter boards.

Proposal

Extend DT overlay usage upstream to include board variants based on a vendor defined board-id property.

Thoughts?

