The case for moving to Debian stretch or Ubuntu 18.04
Contents

1 Why was Apertis based on the Debian/Ubuntu ecosystem 2
2 Why was Ubuntu taken as the direct upstream rather than Debian 2
3 What has changed 3
   Debian Stretch ........................................................................ 3
   Ubuntu 18.04 ........................................................................ 4
4 Impact of move 4
   Will the rebase process take longer if we move to Debian instead of
   the next Ubuntu LTS release? ................................................... 4
   Ubuntu does validation, this would be missing if we move to Debian? 5
   Do we lose anything by moving to Debian? ............................... 5
5 Recommendations 5

This document provides the analysis and rationale for migrating to Debian as the projects upstream distribution. This change was completed in late-2018 and as such all releases since v2019 have been based on Debian.

Why was Apertis based on the Debian/Ubuntu ecosystem

At the beginning of Apertis, a few platforms were considered for the base of
Apertis: MeeGo, Tizen, OpenEmbedded Core, Debian and Ubuntu. A choice
of Debian/Ubuntu ecosystem was based on Debian being ‘one of the oldest and
largest (most inclusive of OSS packages), and one of the first Linux distribu-
tions to feature an ARM port’, providing ‘a very solid distribution baseline’
and ‘a high degree of robustness against the involvement or not of individual
contributing companies’, while Ubuntu bases on Debian but adds value important
for Apertis (see below). Another point against the other alternatives (e.g.
OpenEmbedded Core) was that Collabora and Bosch have already invested into
Open Build System infrastructure, while Yocto/OpenEmbedded has its own
build infrastructure and tools not compatible with OBS.

Another important point was that Collabora employed and continues to em-
ploy many Debian package maintainers, who contribute to key OSS middleware
packages within both the Debian and Ubuntu projects directly, which presented
a serious benefit over other alternatives.
Why was Ubuntu taken as the direct upstream rather than Debian

When the decision to use Ubuntu was taken, Ubuntu had several benefits over Debian. Especially taking into account the initial goal or having an update cycle of around 6 month of the baseline platform.

Debian only releases once every 2 to 2.5 years, while Ubuntu does release every 6 months with every 4th of those being a long-term support release. This means that the only way of doing a refresh every 6 months based directly on Debian would mean creating a snapshots of Debian testing, stabilising that and providing security support for it. Doing that purely for Apertis would of course require a significant amount of resources, but more importantly, it is essentially what Ubuntu is already doing. This made Ubuntu more suitable as a baseline for a 6 month update cycle.

Furthermore, the Linaro initiative used Ubuntu as a reference distribution for all of their validation of hardware enablement. Linaro and Canonical engineers actively integrated the latest work from Linaro and SoC vendors, including Freescale, into Ubuntu. By using Ubuntu as a base Apertis could benefit from and build on this work.

At the time, Ubuntu was the de facto upstream of AppArmor, this included patched kernels to enable latest features (D-Bus mediation, socket mediation, ptrace mediation, etc.) as well as changes to individual packages to improve their apparmor profiles.

What has changed

While doing two base platform refreshes every year has been successful, the users of Apertis weren’t actually set up to follow in such a fast cycle. On top of that the non-LTS releases of Ubuntu limited their security support cycle from 18 months after release to only 9 months after release. In other words the upgrade window since the start of the Apertis project went from around one year after a platform refresh to only 3 months after each platform refresh before the upstream security support end. Such a short time-frame is not achievable with the required updates and validation that are required before a major product rollout.

Due to the policy changes, it was decided to base the Apertis platform on LTS versions rather than refreshing on each version, utilising the longer security support period on these LTS releases. Apertis was last rebased onto the Ubuntu 16.04 LTS release (codenamed “Xenial Xerus”).

Ubuntu and Linaro are no longer collaborating together as they were. Linaro are now supporting various boards using a Debian based release, directly contributing to Debian and no longer supporting Ubuntu.
The infrastructure required by Apparmor has matured to the point where the features used by Apertis have been upstreamed and as such Apertis is no longer tied to Ubuntu in this regard.

**Debian Stretch**

- **Benefits**
  - Debian is a community project, with no single company driving its development
  - Maintenance of components we rely on is not tied to Canonical’s commercial strategies
  - Security support for at least 5 years since the initial stretch release via the Debian LTS project
  - More direct contribution path for package changes done for Apertis since they can go directly into the main upstream distribution
  - Debian stable and security updates tend to be more conservative and stable making it easier to track over time
  - Debian provides a backports repository for packages where a version newer than that in the stable release might be of interest
- **Risks**
  - Debian does not use a strict 2-years release cycle. Thus the Apertis platform update cycle also cannot be strictly time-based when using Debian

**Ubuntu 18.04**

- **Benefits**
  - Ubuntu has a strict time-based release cycle of a new LTS every two years
  - Ubuntu also has a 6-months regular release cycle (with very limited support) should the decision to use LTS version be revised
- **Risks**
  - Ubuntu is bound to the health, technical and commercial strategy of Canonical. Canonical has shifted its focus several times in recent years which has resulted in numerous changes not aligned to the goals of Apertis. Canonical has also introduced their own technologies rather than utilising ‘upstream’ technologies a number of times, for example Mir vs. Wayland and Snappy vs. Flatpak. Some of these choices have had an impact when utilising Ubuntu packages in Apertis, requiring extra work to be performed (e.g. disabling Mir).
  - Ubuntu’s stable releases can have more aggressive updates to certain packages, which can destabilise things for Apertis as well as requiring extra support effort
  - Ubuntu’s main support is around a subset of Debian packages available in the Ubuntu’s *main* repository. A more complete set of packages can be found in Ubuntu’s *Universe* repositories, however these
tend to get less attention, and basically only provide as much support
as Debian provides
– On-going support of Ubuntu depends on the commercial success of
Canonical

Impact of move

Will the rebase process take longer if we move to Debian
instead of the next Ubuntu LTS release?

When Apertis tracked non-LTS Ubuntu releases, rebases were performed every
six months following each release. Every rebase took about two months to
complete. As a part of a rebase procedure, the following tasks needed to be
completed:

- Fork Apertis in preparation of a new release
- Set up Merge-our-Misc to track the latest Ubuntu release
- Repeat until there are no build failures:
  – Accept automatic merges produced Merge-our-Misc
  – If there are no automatic merges, process pending manual merges
  – If new packages break the builds, fix them

Since Apertis no longer pulls changes from regular Ubuntu releases, it is quite
behind the future release which is set to be an LTS. The delta between Apertis
and the current Ubuntu is about the same size as between Apertis and Debian,
and will take similar time to process. Regardless of the decision to stay with
Ubuntu or move to Debian, the following work will need to be done:

- Switch to the latest versions of GCC and rebuild all packages with them
- Rebase all packages to their newer versions from either Ubuntu 18.03 or
  Debian stretch, for each component
- Review Apertis changes to the packages updated upstream, potentially
dropping them if they are no longer relevant
- Switch to the latest Java version for the SDK, dropping Apertis patches
  fixing build failures with the older Java version Apertis shipped

According to our estimation, the difference in the amount of time needed to
perform that work is going to be negligible.

Ubuntu does validation, this would be missing if we move
to Debian?

We understand that Ubuntu does some hardware validation testing of standard
Ubuntu configurations (which we do not use) against hardware from their part-
ners who pay for it (https://certification.ubuntu.com/). The vast majority of
the functionality that such tests will focus on are related to kernel functionality.
Since we do not for the most part use the Ubuntu kernel and target different
hardware, these tests do not seem relevant in our use case and thus we do not lose anything by moving to Debian.

**Do we lose anything by moving to Debian?**

We believe that we do not lose anything other than the strict time-based release cycle by moving from Ubuntu to Debian. However, we feel that this is now less important given we are now syncing on just Ubuntu LTS releases (every 2 years), and with Debian release cycle tending to a 2 year release cycle this is not believed to be problematic.

**Recommendations**

Collabora recommends rebasing on Debian Stretch for Apertis 18.06 and onwards. Most of the benefits of basing on Ubuntu have gone away since the original decision was taken in late 2011, while the projects dynamics have also changed to better suit a Debian based distribution. Basing on Debian rather than Ubuntu, would move Apertis closer to it’s ultimate upstream (as Ubuntu is also a downstream of Debian) cutting out a middle-man, which currently brings very little to the table as described above. This also may make the process of upstreaming appropriate package changes more efficient, reducing the maintenance overhead in Apertis.