Finding the Path from Embedded to Edge using Product Lines

Dr. Steffen Evers, Bosch.IO
Philipp Ahmann, Robert Bosch GmbH
Bringing the IoT to life

Sources: https://www.youtube.com/watch?v=YfLiwpwEqtU, https://www.youtube.com/watch?v=uaeADiepFXk
>50 Mio. expected Linux based devices produced by Bosch in 2025 p.a.
Build

Maintenance costs, re-use, upstream

Focus

Hardware costs, SoP, differentiators, flexibility, risks

Maintain
Drowning in issues alone

Missing opportunity to share burden
Finding the Path from Embedded to Edge using Product Lines
System Evolution Level

Standalone

Connected

IoT

Container based

just Linux custom build (1 of n)

Layer/Packages @ SDK

eyo3

eyo2

eyo1

dev

dev

dev

edge computing
Finding the Path from Embedded to Edge using Product Lines

Our Embedded IoT Linux Ecosystem

Current focus areas

- Container
- SW Management (OSS compliance)
- SoC mainling
- Security
- Virtualization
Reference System Elements

Container OSS Compliance & APERTIS
Finding the Path from Embedded to Edge using Product Lines

Container Frameworks – Benchmarks

Goals

• Strengths and weaknesses
• Identify critical properties
• Build up knowledge base and experience

Measurements and observations (so far)

• All frameworks show reasonable performance.
• Individual properties differ.

“Best” container solution depends on use case demands
### Finding the Path from Embedded to Edge using Product Lines

**Container Frameworks – Benchmarks**

<table>
<thead>
<tr>
<th></th>
<th>LXC</th>
<th>FLATPAK</th>
<th>docker</th>
<th>podman</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Startup time [sec]</strong></td>
<td>~0.85</td>
<td>~0.68</td>
<td>~3.02</td>
<td>~2.85</td>
</tr>
<tr>
<td><strong>Network latency (TCP)</strong></td>
<td>~8.1</td>
<td>~2.6</td>
<td>~4.4</td>
<td>~2.6</td>
</tr>
<tr>
<td><strong>Host to x [ms]</strong></td>
<td>~17</td>
<td>~102</td>
<td>~293</td>
<td>~179</td>
</tr>
<tr>
<td><strong>Storage installation [MB]</strong></td>
<td>~127</td>
<td>~54</td>
<td>~120</td>
<td>~125</td>
</tr>
</tbody>
</table>

- **CPU and memory usage:** testing method created comparable results using stress-ng's VM stressor to create memory and CPU load while using top and SMEM tool or `/proc/meminfo`, we observed similar results, as the sys calls use the same kernel and runtime.
- **Disk I/O performance:** depends more on your disk, than your container framework
Embedded Container Orchestration
The next step

- Reliability
- Bandwidth
- Autonomy
- Resource constraints
- Heterogenous devices
Automated Open Compliance

Finding the Path from Embedded to Edge using Product Lines
Finding the Path from Embedded to Edge using Product Lines
Automated Open Compliance

More details see: https://github.com/oss-review-toolkit/ort
Finding the Path from Embedded to Edge using Product Lines
All details available at: www.APERTIS.org

... Conceived for Automotive infotainment use cases. Now deployed across various industries and products. Focus is on Security and Modularity.

- Debian derivative, using OStree, latest Linux LTS and more
- All packages available in source and binary form
- Pre-built images (aarch64, amd64, lxc, ostree, ...)
- Extensive SDK, documentation, and infrastructure
Finding the Path from Embedded to Edge using Product Lines
APERTIS ... not only for Automotive

APERTIS can be used in various application fields.

Currently APERTIS is enhanced to fit a wider AIoT software service eco system
# Embedded Debian: Increase efficiency with additional (shared) layers

<table>
<thead>
<tr>
<th>Hardware (Support)</th>
<th>Software (Base System)</th>
<th>Software (Add-On)</th>
<th>Runtime Services</th>
<th>Tooling (Incl. Services)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Upstream projects</td>
<td>Embedded Debian</td>
<td>Apertis</td>
<td>Embedded Debian</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Aligned generic</td>
<td>Apertis Pro (Technology)</td>
<td>Apertis Pro (Product)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>elements ready to use inside Bosch</td>
<td>Generic elements adjusted to customer needs</td>
<td>Customize to your specific needs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Firmware &amp; Updates</td>
<td>Add-On Software</td>
<td>Backend Clients</td>
</tr>
</tbody>
</table>

**Upstream projects**

- Additional (shared) layers for efficiency

**Embedded Debian**

- Aligned generic elements ready to use inside Bosch

**Apertis**

- Technology

**Apertis Pro** (Product)

- Generic elements adjusted to customer needs

**Embedded Software**

- Project

**Device Software**

- Firmware & Updates

**Add-On Software**

- Backend Clients
Finding the Path from Embedded to Edge using Product Lines
APERTIS from source to update

- CI/CT toolchain & SDK for faster product ramp up
- Cross compilation
- Over the air update (package manager for development mainly)
- Integration of customer specific software, FOSS packages, Binaries
Handling GPL-3

- bash fully replaced by dash
- Original coreutils replaced by uutils/coreutils (Rust, MIT)
- Old GnuPG GPL-2.0 version still maintained (options: rPGP and/or Sequoia)

More details see:
https://www.apertis.org/concepts/gpl3_free_deltas/
https://www.apertis.org/concepts/gnupg-replacement/
https://www.apertis.org/concepts/coreutils-replacement/
Upstreaming of hardware support for inhouse devices

- SoC reasonably supported
- Board and schematics available
- IP modules which are not upstream are major tasks
- Some IP may also not be owned by silicon vendor

Go outside together with the people rather than keeping code inside. Be part of the community.
The journey goes on...

**Friday, June 24 • 2:00pm - 2:40pm**

**BOF: Corporate Use of Embedded Linux - Tim Bird, Sony Corporation**

*Photo by Patrick Tomasso on Unsplash*
THANK YOU

Dr. Steffen Evers
Director Open Source
steffen.evers@bosch.io
Bosch.IO GmbH, Berlin

Philipp Ahmann
Product Management - Open Source
philipp.ahmann@de.bosch.com
Robert Bosch GmbH, Hildesheim