



EMBEDDED  
LINUX  
CONFERENCE

# APERTIS - A Collaborative Industrial Grade Linux Construction Kit

Philipp Ahmann, Robert Bosch GmbH  
Supported by the great APERTIS team from Collabora



#EmbeddedOSSummit



# whoami - Philipp Ahmann



Product Manager for Linux & Embedded Open Source



Chair of the Technical Steering Committee  
Lead of the Systems Working Group



Member of the Inaugural Advisory Board



OSS enthusiast and promoter



# Content

## Setting the Scene

- Why was Apertis started?
- What is Apertis?

## Ecosystem

- What is inside?
- How does the work flow and infrastructure look like?

## Experience

- Example systems in production.
- Try it out.

## What else to say?

- CIP Project - a related project example
- Discussion about embedded system in production.

# Setting the Scene

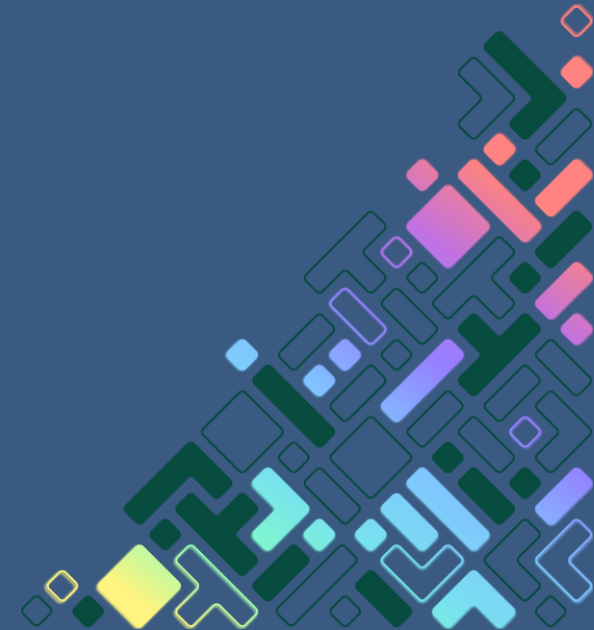




Photo by Josue Isai Ramos Figueroa on Unsplash

Hardware costs, SoP, differentiators, flexibility, risks

Build

# Extended Focus



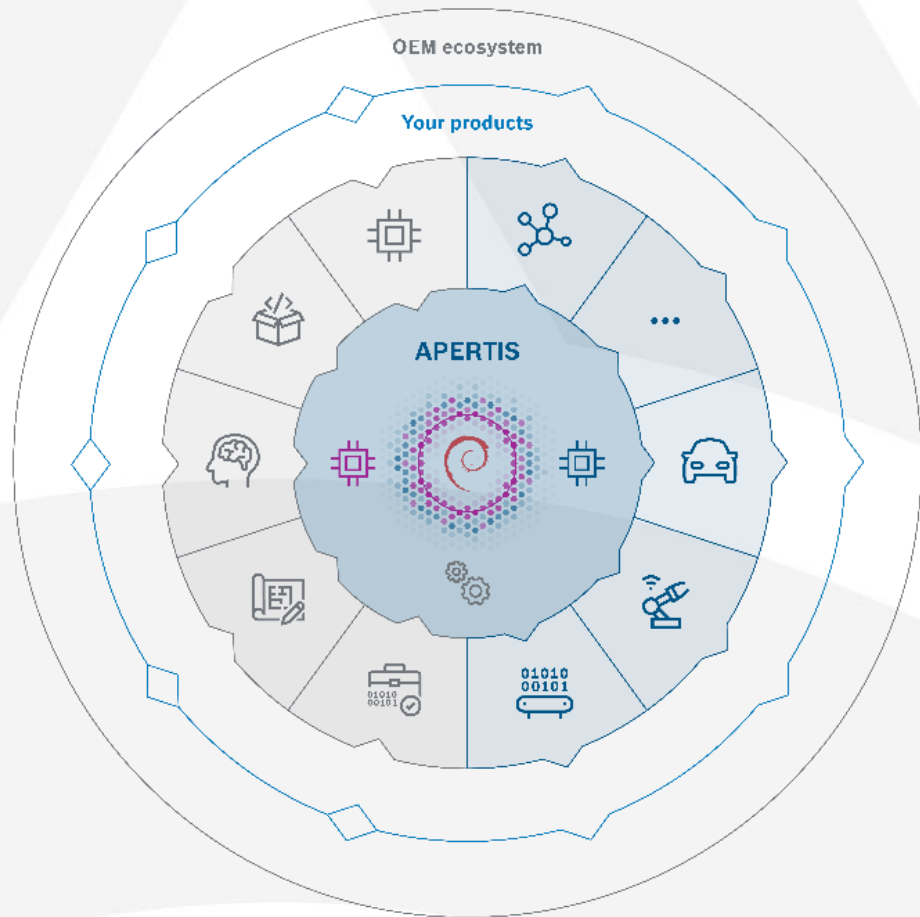
Maintenance costs, re-use, upstream

Maintain



Photo by Taylor Simpson on Unsplash

# What is APERTIS?



... Free and open source, GNU/Linux-based distribution (originally) for infotainment in automotive vehicles, with focus on security and modularity.



- **debian** derivative tailored for industrial needs
- Fit for a wide variety of electronic devices
- Product-specific images for ARM and Intel x86
- Beyond operating system, it offers frameworks, new APIs, cloud services, SDK, ...

# APERTIS use ... not only for Automotive



Apertis can be used in various application fields.



We evaluated Apertis for its fit into a wider AIoT software service eco system

# <https://www.apertis.org/> The place to start your journey



The screenshot shows the Apertis website homepage. At the top left is a navigation menu with a close button (X) and the Apertis logo and name. The menu items are: ARCHITECTURE, CONCEPT DESIGNS, GUIDES, POLICIES, QUALITY ASSURANCE, REFERENCE HARDWARE, and RELEASES. At the top right are links for SERVICE STATUS, DOWNLOAD, and GITLAB. The main content area has a dark blue background with the Apertis logo and the text 'Collaborative OS platform for products'. Below this is a paragraph of text describing Apertis as a versatile open source infrastructure. At the bottom, there are three feature cards: 'Production Friendly' (with an image of a factory), 'Frequent Releases' (with an image of hands on a keyboard), and 'Developer Ready' (with an image of a person at a computer).

APERTIS

SERVICE STATUS DOWNLOAD GITLAB

ARCHITECTURE ▲  
CONCEPT DESIGNS ▲  
GUIDES ▲  
POLICIES ▲  
QUALITY ASSURANCE ▲  
REFERENCE HARDWARE ▲  
RELEASES ▲

APERTIS

Collaborative OS platform for products

Apertis is a versatile **open source infrastructure**, fit for a wide variety of electronic devices, with a history within the automotive industry. Security and modularity are two of its primary strengths. Apertis provides a feature-rich framework for add-on software and resilient upgrade capabilities. Beyond an operating system, it offers tools and cloud services to optimise development and increase reliability.

**Production Friendly**

Apertis provides suitably licensed, tested releases.

**Frequent Releases**

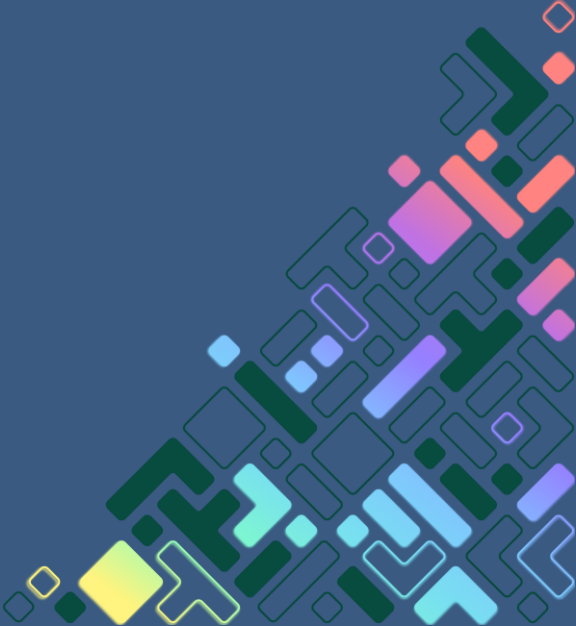
A fresh stable version of Apertis is released yearly.

**Developer Ready**

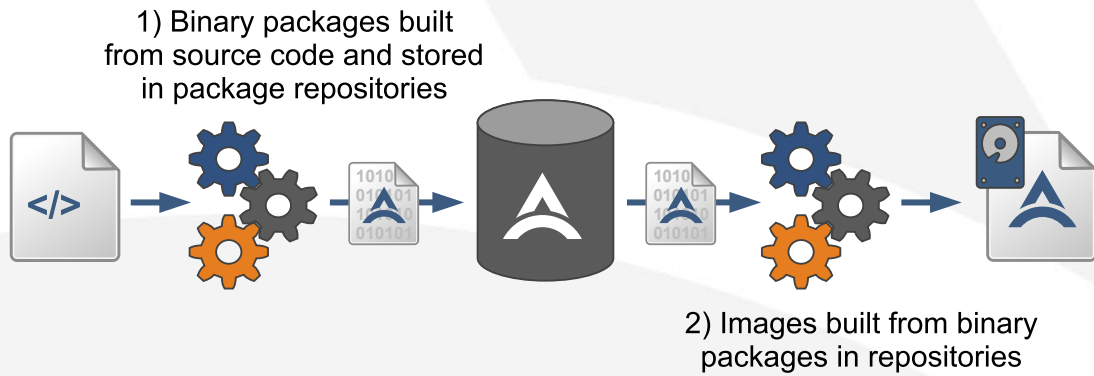
Ready to use infrastructure, automation and images.



# Ecosystem



# Package Centric Approach



- With Apertis, developers can focus on developing the components that provide their unique experience and
  - rely on the shared core components
  - infrastructure for everything else.
- The package centric approach ensures reproducibility, traceability and consistency during the whole product life cycle.

# It is all about packages [gitlab.apertis.org/pkg](https://gitlab.apertis.org/pkg)



🚩 ● v2025dev1 entered Hard Code Freeze, changes will not be accepted • [schedule](#)

**P** pkg 🌐

Subgroups and projects Shared projects Archived projects

🔍 Search Updated ▾ ⌵

📁 <b>A</b> apertis-tests 🌐	LAVA test definitions and implementations Repository imported from <a href="https://git.apertis.org/git/apertis-tests.git/">https://git.apertis.org/git/apertis-tests.git/</a>	★ 0	25 minutes ago
📁 <b>Q</b> qemu 🌐		★ 0	37 minutes ago
📁 <b>L</b> linux 🌐		★ 0	2 hours ago
📁 <b>L</b> libgpod 🌐		★ 0	3 hours ago
📁 <b>D</b> Debmake 🌐		★ 0	3 hours ago
📁 <b>P</b> pkgconf 🌐		★ 0	3 hours ago
📁 <b>A</b> atf 🌐		★ 0	5 hours ago
📁 <b>K</b> kyua 🌐		★ 0	5 hours ago
📁 <b>L</b> lutok 🌐		★ 0	22 hours ago

~7000 packages

# Packages need to be released

## [apertis.org/release/](https://apertis.org/release/)

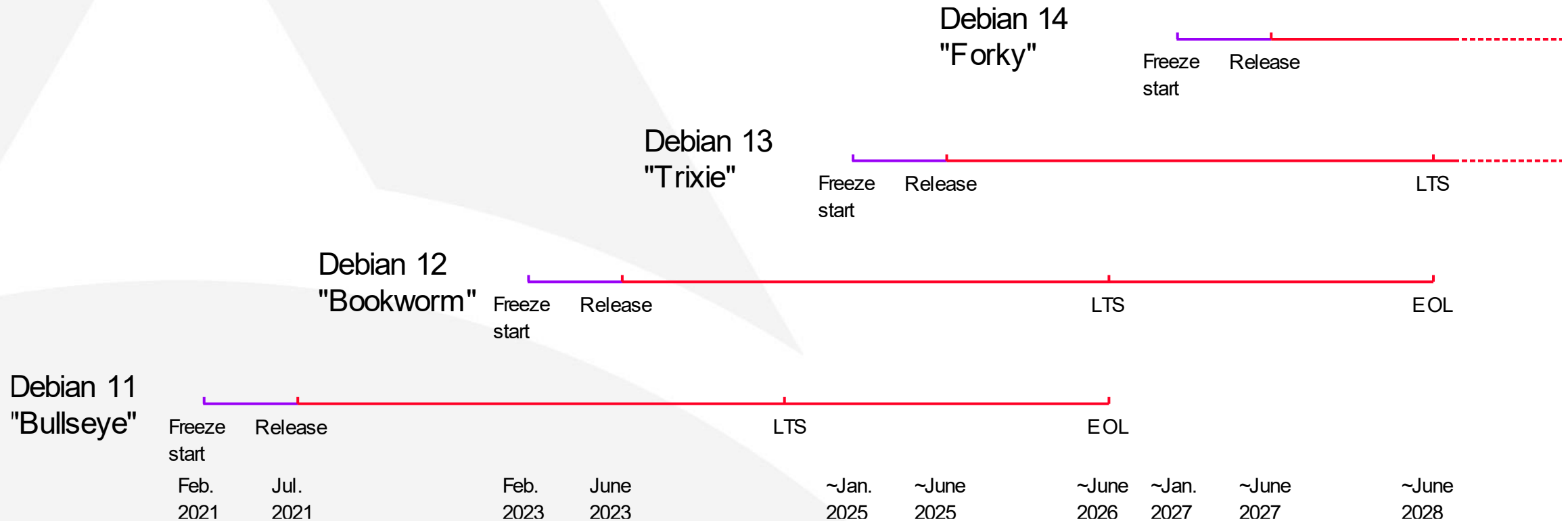


Recent release cycle from April:

<b>Milestone</b>	<b>2023.5</b>	<b>2024.1</b>	<b>V2025dev2</b>
Start of Release Cycle	2024-04-01	2024-04-01	2024-04-01
Soft Feature Freeze	2024-05-01	2024-05-08	2024-05-15
Hard Feature Freeze / Soft Code Freeze	2024-05-08	2024-05-15	2024-05-22
Release Candidate (RC1) / Hard Code Freeze	2024-05-15	2024-05-22	2024-05-29
Release	2024-05-30	2024-06-06	2024-06-13

# Packages come from Debian releases

## [policies/release-flow/#debian-release-processes](#)

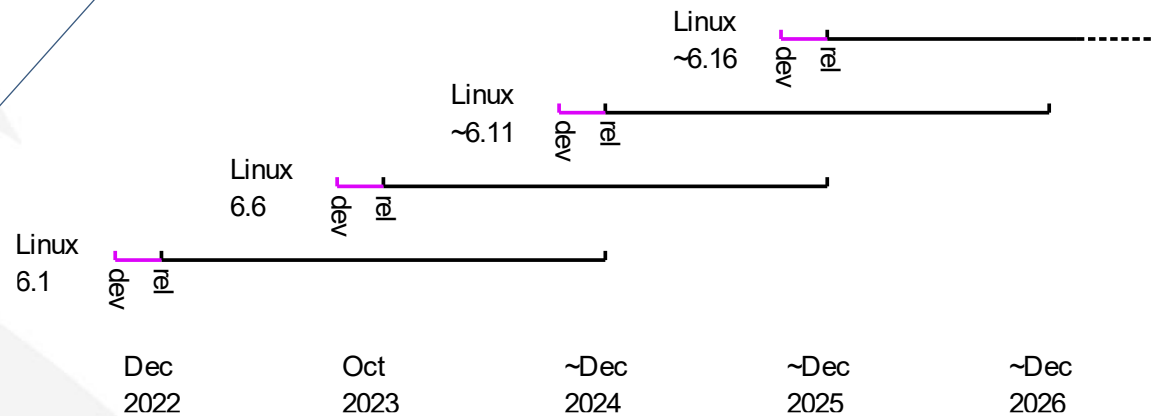


# Kernel release flow

## [policies/release-flow/#linux-kernel-release-flow](#)



- Apertis is following the Linux kernel LTS releases to ensure it includes modern features and support for recent hardware.
- More frequent kernel updates are typically not requested by customers
- Latest Apertis releases run on 6.6 kernel

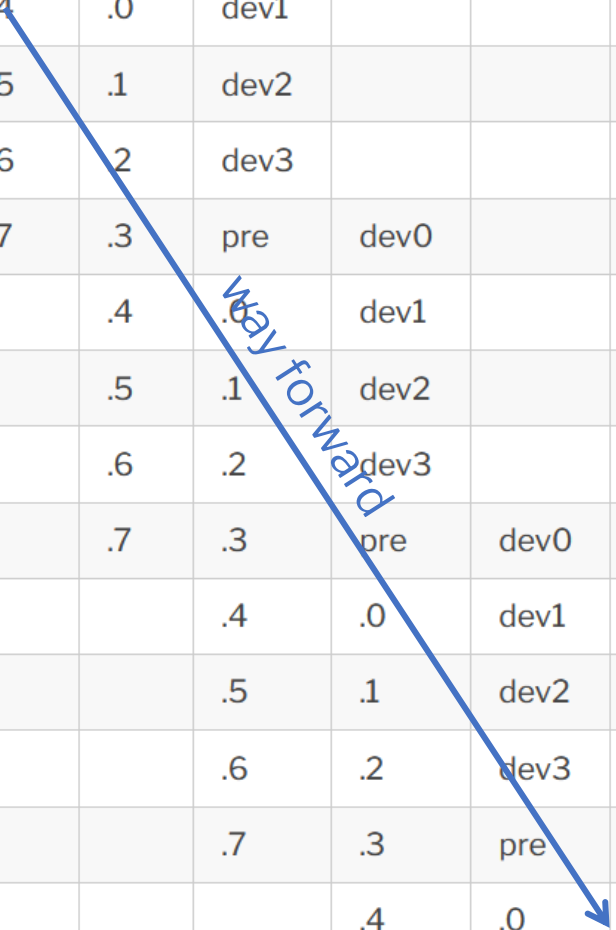


# Version scheme

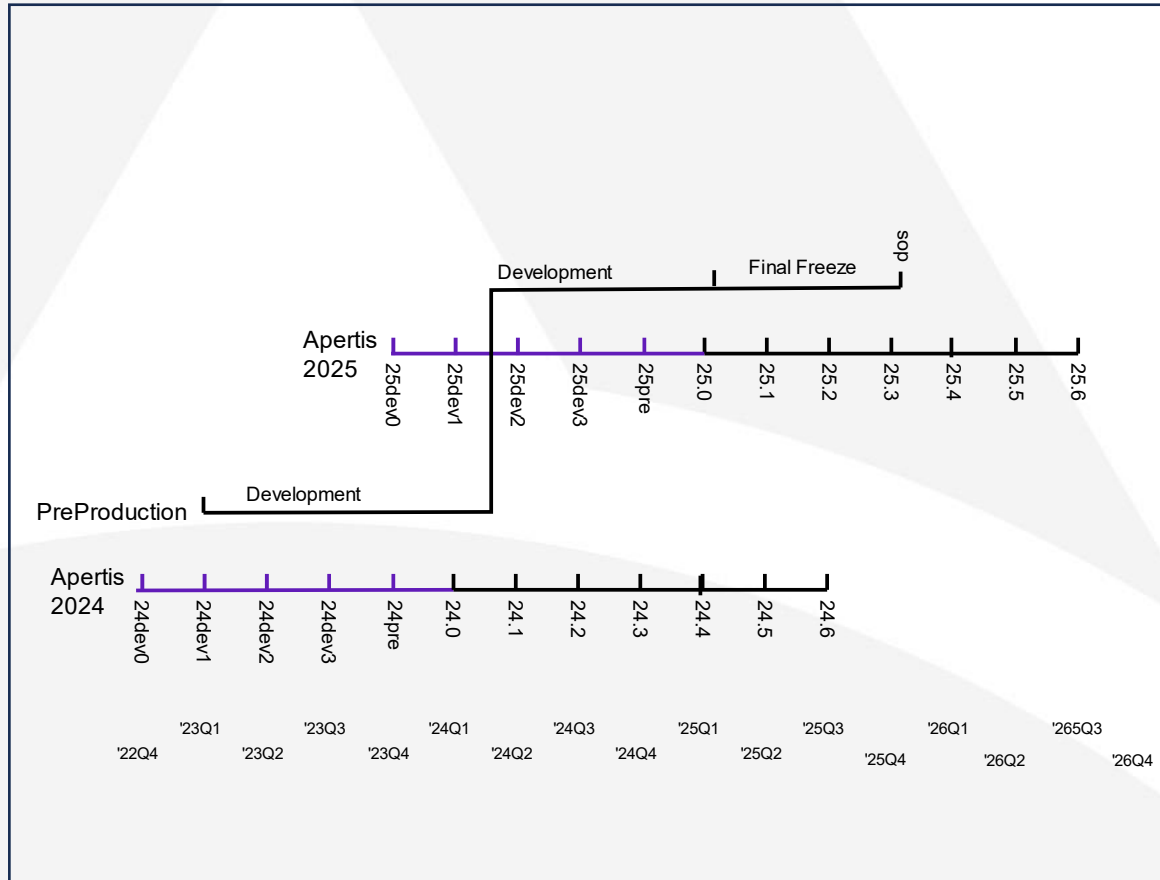
## policies/release-flow/#apertis-release-flow

Quarter	N-2	N-1	N	N+1	N+2	N+3	v2023	v2024	v2025	v2026
Q1	.4	.0	dev1				v2023.0	v2024.dev1		
Q2	.5	.1	dev2				v2023.1	v2024.dev2		
Q3	.6	.2	dev3				v2023.2	v2024.dev3		
Q4	.7	.3	pre	dev0			v2023.3	v2024.pre	v2025.dev0	
Q1		.4	.0	dev1			v2023.4	v2024.0	v2025.dev1	
Q2		.5	.1	dev2			v2023.5	v2024.1	v2025.dev2	
Q3		.6	.2	dev3			v2023.6	v2024.2	v2025.dev3	
Q4		.7	.3	pre	dev0		v2023.7	v2024.3	v2025.pre	v2026.dev0
Q1			.4	.0	dev1			v2024.4	v2025.0	v2026.dev1
Q2			.5	.1	dev2			v2024.5	v2025.1	v2026.dev2
Q3			.6	.2	dev3			v2024.6	v2025.2	v2026.dev3
Q4			.7	.3	pre	dev0		v2024.7	v2025.3	v2026.pre
Q1				.4	.0	dev1			v2025.4	v2026.0

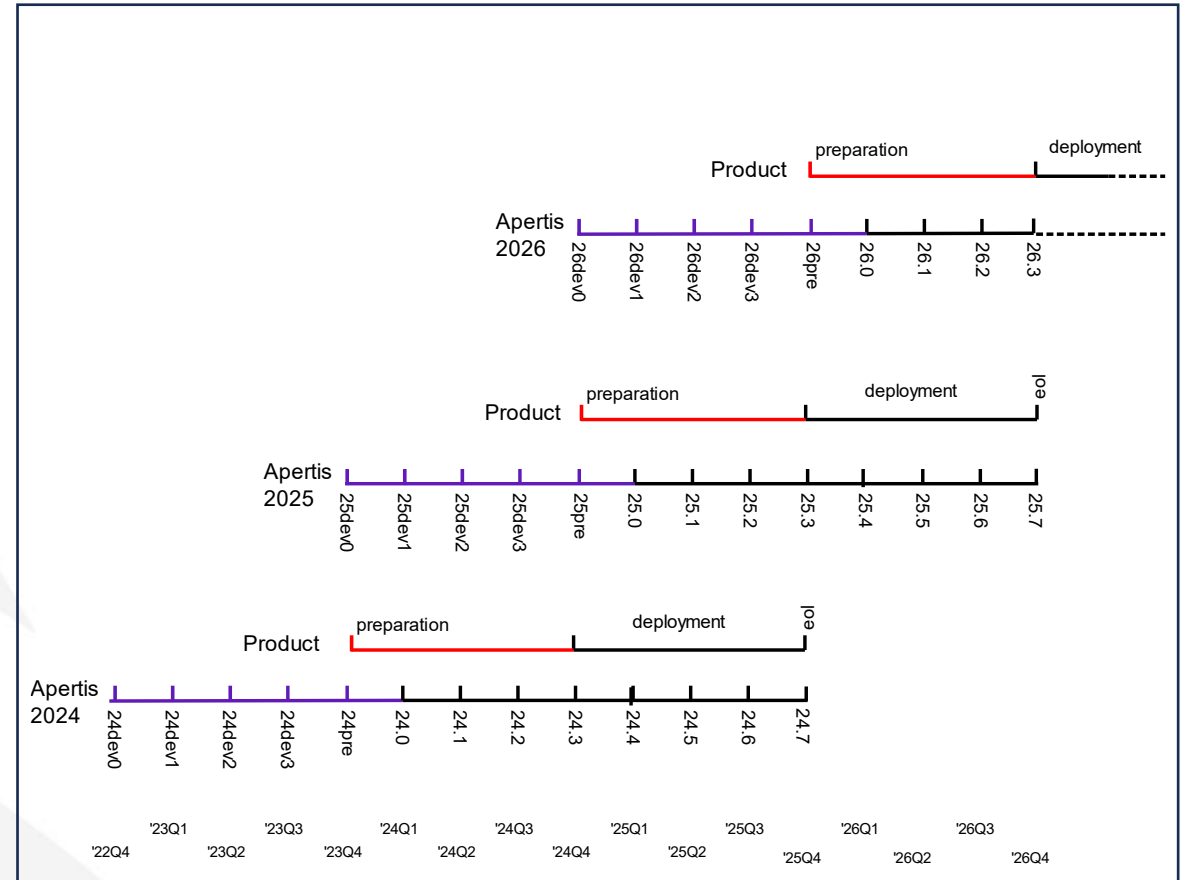
Way forward



# Product update cycle



pre-production phase



production phase



# Releases need to be tested.

<https://qa.apertis.org/>

Test Suites and Reports List Login

Test Suites v2025dev2 v2025dev1 v2025dev0 v2024 v2024pre v2024dev3 v2024dev2 v2023 v2022 v2021

Test Reports Show all

**Releases**  
v2025dev1.0rc1 v2025dev1 / 20240228.0017

	executed	pass	skip	fail	incomplete
apt (a)	243	231	0	10	2
apt (m)	147	145	0	2	0
lxc (a)	36	33	0	3	0
nfs (a)	3	3	0	0	0
ostree (a)	173	165	0	4	4
ostree (m)	84	76	0	8	0

**Weeklies**  
v2025dev2 / 20240306.0316

	executed	pass	skip	fail	incomplete
apt (a)	243	210	0	12	21
lxc (a)	36	33	0	3	0
nfs (a)	3	3	0	0	0
ostree (a)	173	163	0	4	6

**Dailies**  
v2025dev2 / 20240313.1154

	executed	pass	skip	fail	incomplete
apt (a)	179	170	0	5	4
ostree (a)	129	117	0	0	12

**v2024.0**  
v2024 / 20240221.0115

	executed	pass	skip	fail	incomplete
apt (a)	215	203	0	8	4
apt (m)	147	145	0	2	0
lxc (a)	36	31	0	5	0
nfs (a)	3	3	0	0	0
ostree (a)	152	148	0	3	1
ostree (m)	84	79	0	5	0

**Weeklies**  
v2023 / 20240306.0217

	executed	pass	skip	fail	incomplete
apt (a)	201	172	0	2	27
lxc (a)	36	28	0	2	6
nfs (a)	3	3	0	0	0
ostree (a)	146	145	0	0	1

**v2025dev2 / 20240313.0315**

	executed	pass	skip	fail	incomplete
apt (a)	228	215	0	7	6
lxc (a)	36	33	0	3	0
nfs (a)	3	3	0	0	0
ostree (a)	162	154	0	1	7

- LAVA testing
  - (a) = automatic testing
  - (m) = manual → executed for Releases
- Unit testing on package base during CI
  - not part of formal test reports,
  - unit tests come with the packages
- Image variants (apt & ostree) considered
- Reference hardware with several architectures
  - arm64, arm32, amd64
  - [https://www.apertis.org/reference\\_hardware/](https://www.apertis.org/reference_hardware/)
  - TI SK-AM62 available from v2025 and onwards
- SDK tests on qemu

# Collabora Lava test farm



# Strong copyleft, a challenge for commercial products - [concepts/gpl3\\_free\\_deltas/](https://www.apertis.org/concepts/gpl3_free_deltas/)

Examples:

- bash fully replaced by dash
- original coreutils replaced by utils/coreutils (Rust, MIT)
- GnuPG GPL-2.0 version replaced by Sequoia (in latest v2023)
- Tar GPL-2.0 replaced by libarchive incl. bsdtar (BSD-2-clause)
  
- More details see:  
[https://www.apertis.org/concepts/gpl3\\_free\\_deltas/](https://www.apertis.org/concepts/gpl3_free_deltas/)  
<https://www.apertis.org/concepts/gnupg-replacement/>  
<https://www.apertis.org/concepts/coreutils-replacement/>



Photo by Szylemon Fikcyjny on Unsplash

# Wide use of existing open source (infrastructure) projects.



Flatpak

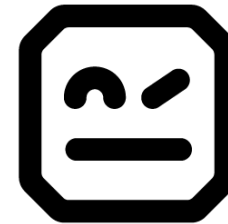
OSTree



podman

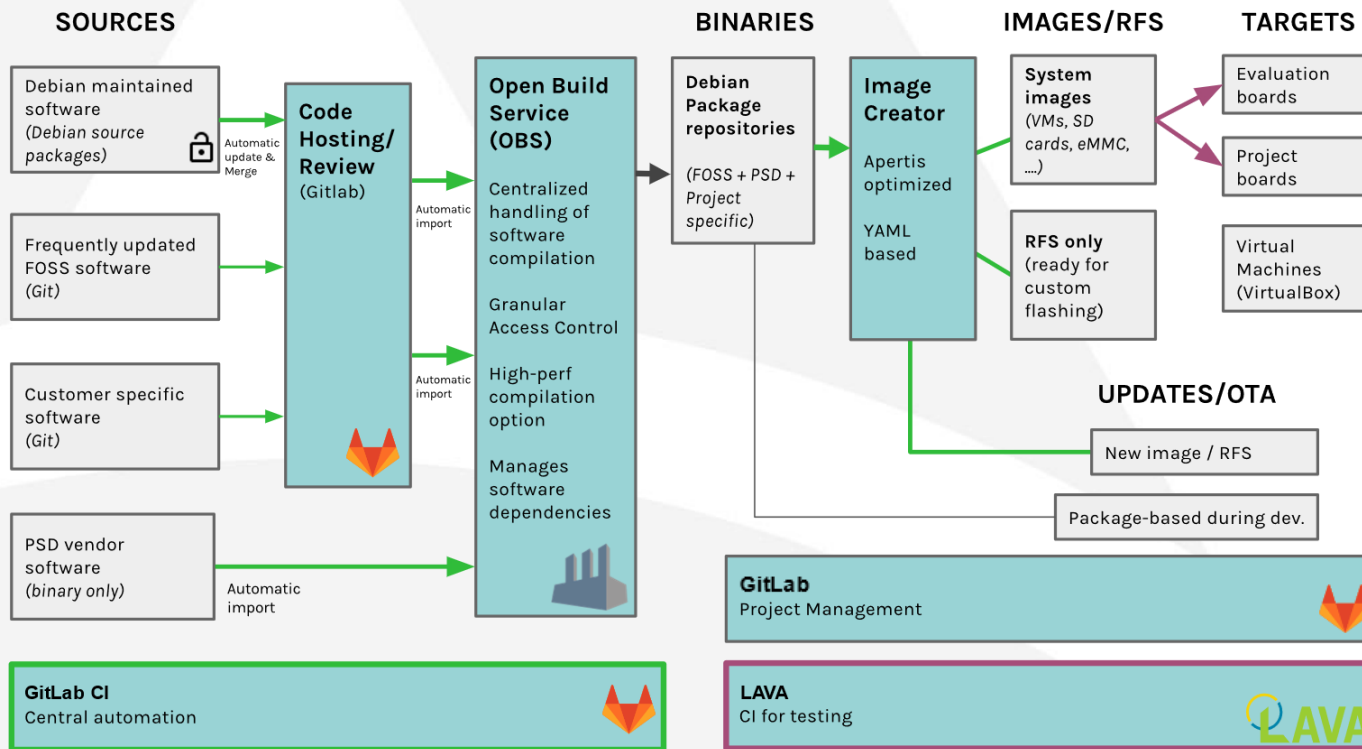


GitLab



debian

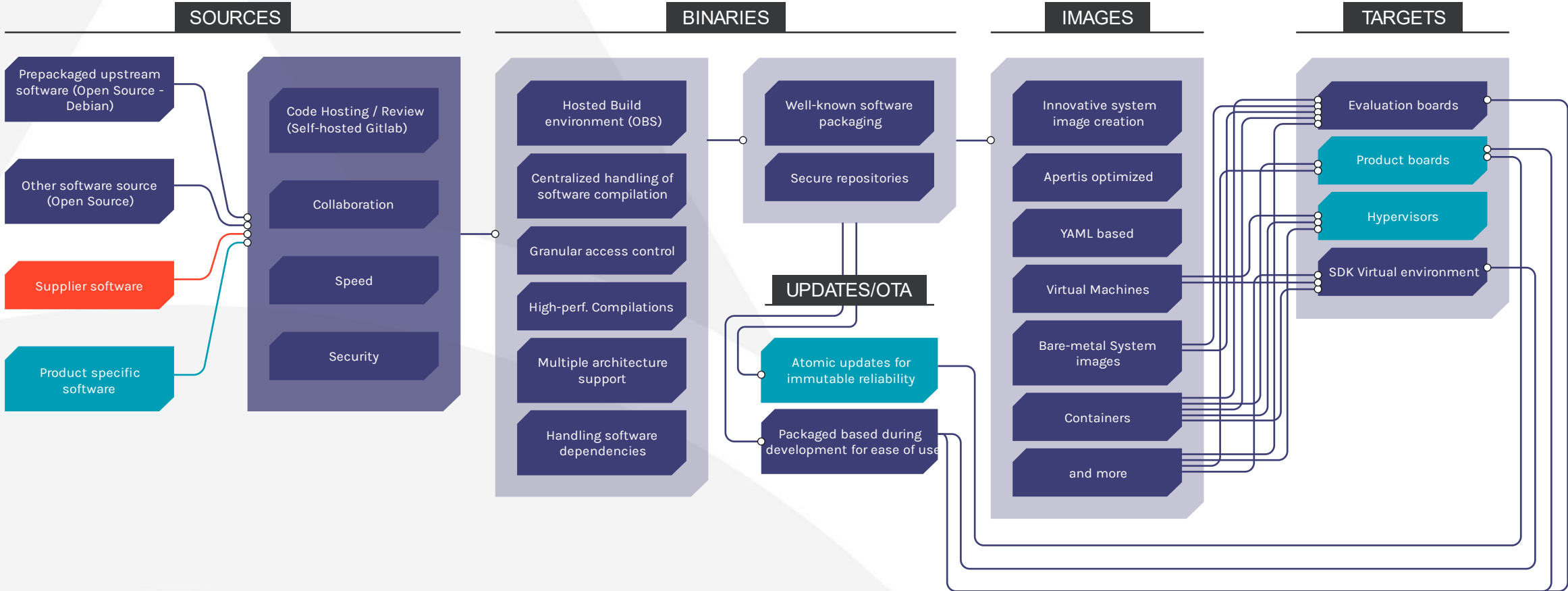
# Architectural infrastructure elements architecture/workflow-guide/



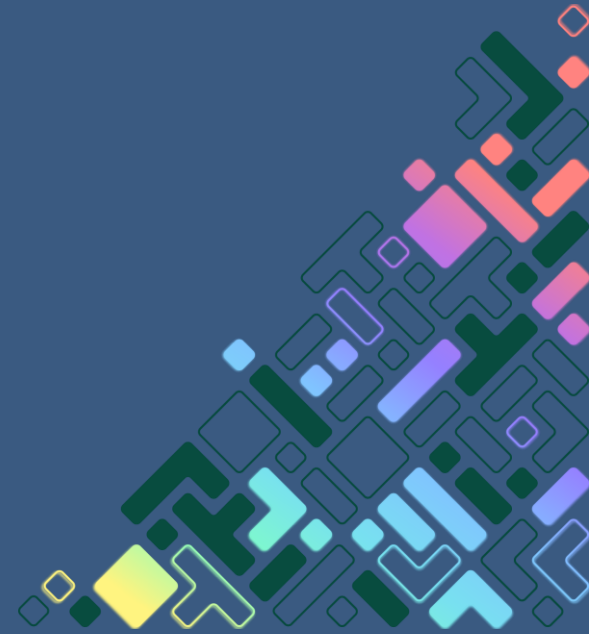
- CI/CT toolchain & SDK for faster product ramp up
- Cross compilation
- Over the air update (package manager mainly for development)
- Integration of customer specific software, FOSS packages, Binaries

# Architectural infrastructure elements

## [architecture/workflow-guide/](#)



# Experience



# Bosch Power Tools run APERTIS on the D-TECT 200



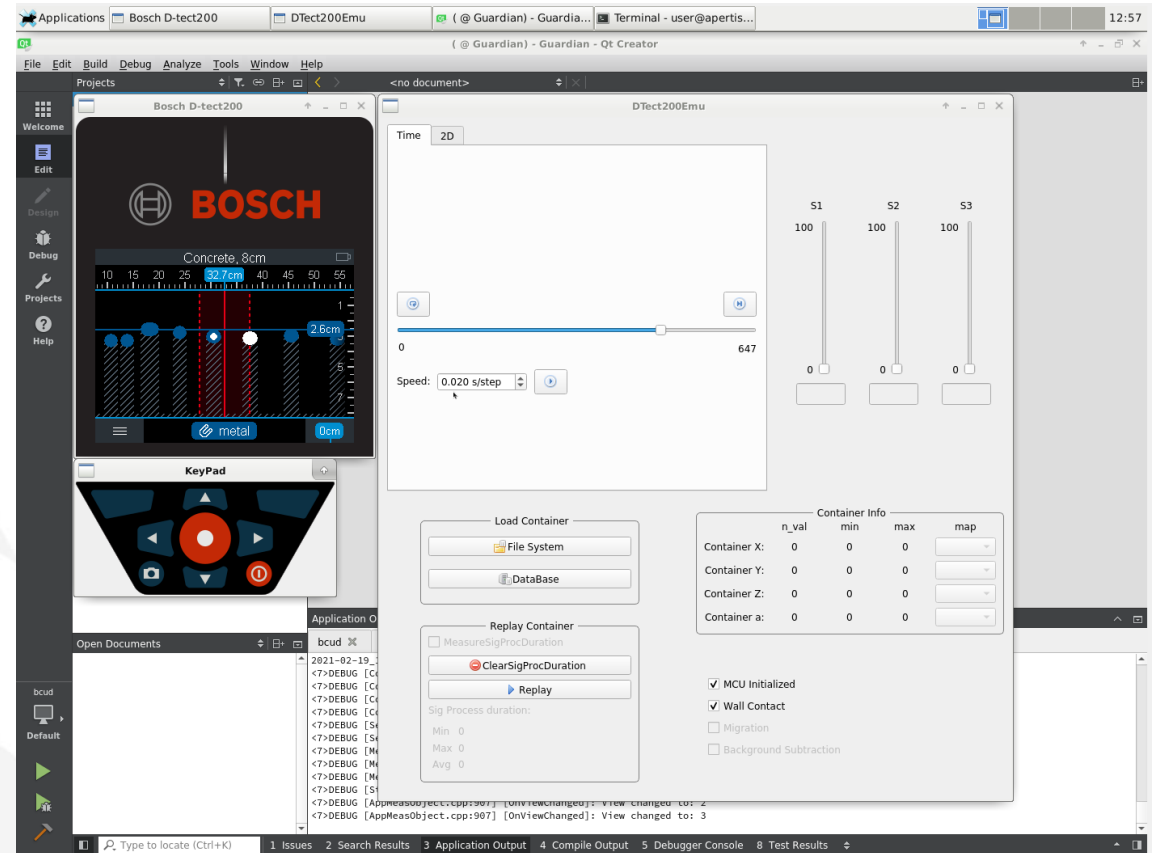
- Wall scanner product for worldwide market
- From Automotive to Measurement Equipment
- Power Tools make heavy use of vBox images for their development (e.g. for deeply embedded developers)
- Good fit for smaller team. No need for maintaining own CI.





# Speed up development with virtual images

- QT based visual representation of the wallscanner
- No need to have physical device for development due to pre-recorded wall (raw) data
- Fast “feel home” environment.
- vBox packages are matching device packages



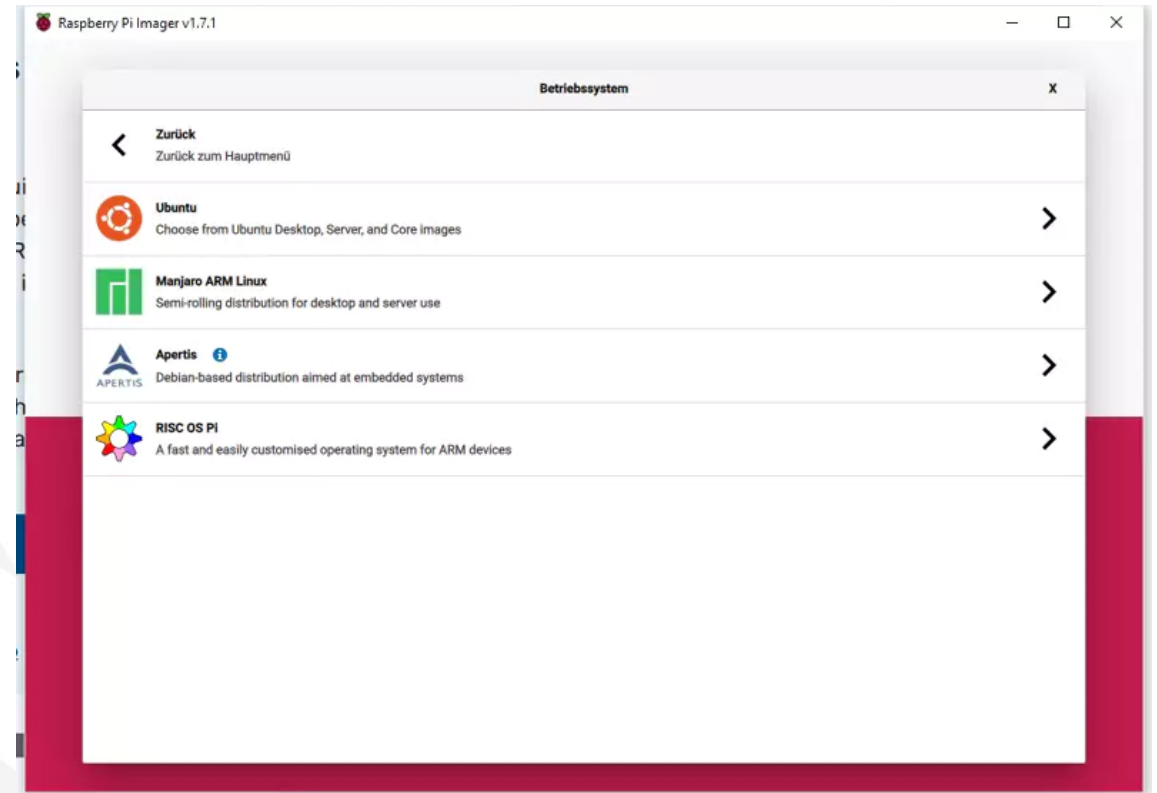
# Or just play with APERTIS...

- Beside games a lot of fancy stuff is included in the Atari VCS... like Weston, Wayland compositor, Chrome browser, Rust, ...
- The tools and methodologies of the Apertis open source infrastructure were applied during the build of the Atari VCS
- Read the Collabora blog: <https://www.collabora.com/news-and-blog/news-and-events/blast-from-the-past-at-embedded-world-atari-plays-for-linux.html>



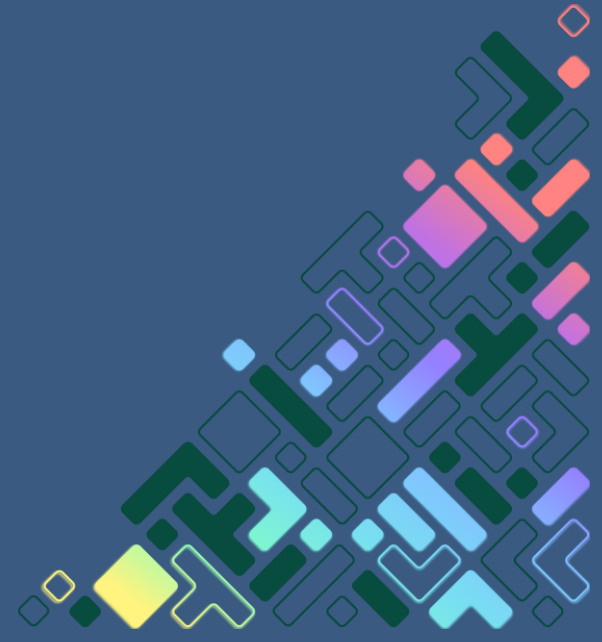
# Reference Hardware: Raspberry Pi4

- Raspberry Pi imager supports Apertis
- Collabora confirmed to continue support for Raspberry Pi imager



<https://www.heise.de/news/Pi-Imager-Neue-Version-mit-neuen-Betriebssystemen-und-Tools-6509481.html>

**What else to say?**



# Related activity spotlight

- Embedded Debian Space: Civil Infrastructure Platform (CIP) project
- Major similarities:
  - Package hardening
  - Debian based
  - Industrial Grade Software
  - Lava testing
- Major differences:
  - Maintenance strategy: Far longer LTS kernels
  - Build tooling incl. yocto parts (see next slide)



<https://cip-project.org/>

# CIP build tooling

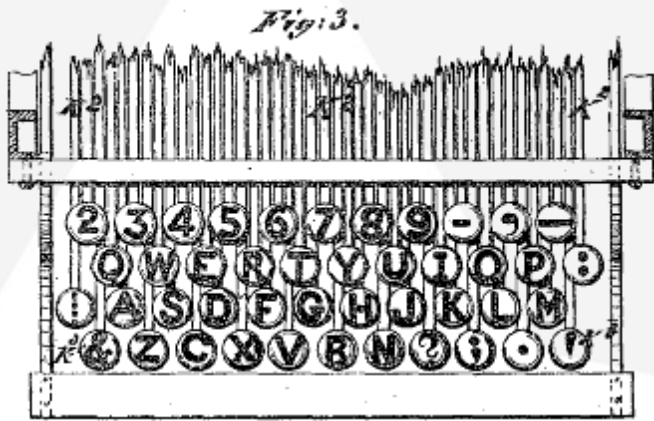
## <https://github.com/ilbers/isar>

- **isar** - Integration System for Automated Root filesystem generation
  - Developed by ilbers GmbH & Siemens AG, Sponsored by Siemens AG
  - Build custom packages from source
  - Bootstrap Debian base system
  - Apply customizations
  - Create complete firmware images
- isar uses **kas**
  - clone and checkout bitbake layers
  - create default bitbake settings (machine, arch, ...)
  - launch minimal build environment, reducing risk of host contamination
  - initiate bitbake build process
  - See the [kas documentation](#) for further details.



<https://youtu.be/GMz3Gyrj0So?feature=shared>

# Also established solutions may have room for improvement



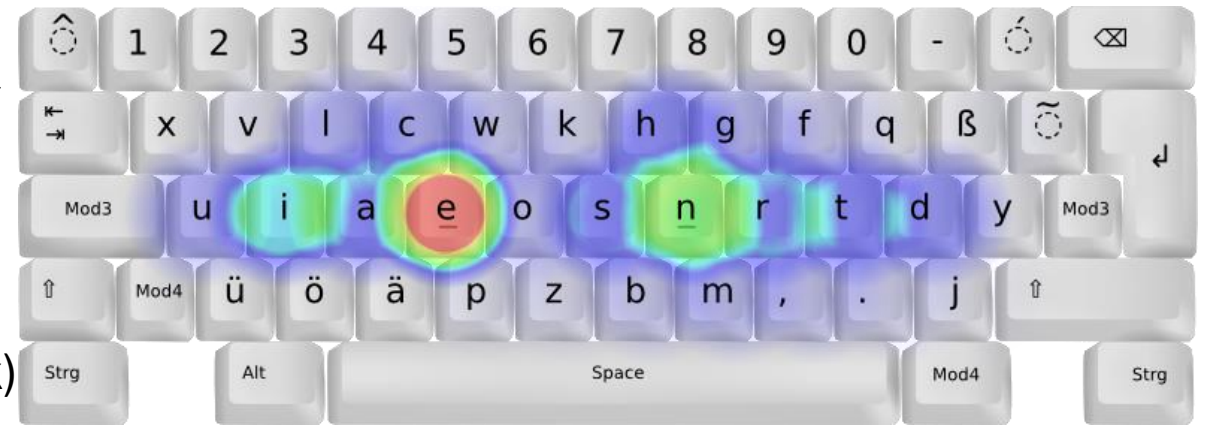
Von C.L. Sholes - U.S. Patent No. 207,559,  
<https://commons.wikimedia.org/w/index.php?curid=4432202>



QWERTY

vs.

Neo (or Dvorak)



From Pudelgedudel - CC BY-SA 3.0,  
<https://commons.wikimedia.org/w/index.php?curid=33506834>



Typical (known) keyboard layout is optimized for mechanical typewriter.  
So, many of us still type modern SW, based on 1868 technology/standard.

# How do you master product challenges?

- Long term support for 10+ years with mandatory updates
- Variant handling for 100+ SoPs and several customers
- Common infrastructure across projects and products
- Avoid last mile improvements during integration
- ...



# Want to solve these challenges together?



Photo by [Hannah Busing](#) on [Unsplash](#)

With more supporters we can grow Apertis (and put it under a foundation stewardship)



Photo by [Kevin Butz](#) on [Unsplash](#)



**EMBEDDED  
LINUX  
CONFERENCE**

