ELISA
Enabling Linux in Safety Applications

WORKSHOP

ELISA Workshop
Munich, Germany

November 18-20, 2025
Co-hosted with Red Hat

Beyond the OS: What else is required for *safe* automotive applications?

Agenda

- About Elektrobit
- The Safe OS
- The Safe Application
- Automotive Use-Cases

Our software moves the world

More than **600 million vehicles** with over **5 billion embedded** devices

The safe OS

Recap: making the OS „safe“

- There are different approaches.
- Argue that the OS kernel (trapeze artist) is safe.
- Prevent the OS kernel from causing harm by using a “safety net.” (ASIL Decomposition)

Achieved guarantees

Spatial freedom from interference

The Linux kernel and other applications cannot corrupt the memory of a safety-related application

Temporal freedom from interference

The kernel cannot block applications indefinitely

Communication integrity

Shared memory communication is protected

Startup integrity

Safety-related applications are checked for proper startup

The Open Source Advantage (1/2)

Start now in the cloud

- ✓ *Shift left:* Begin software development **6–12 months earlier**, even before hardware arrives
- ✓ *Scale globally:* Enable teams to work in parallel, **independent of hardware constraints**

Launch new projects faster

- ✓ *Accelerate onboarding:* Get developers, integrators, and testers **productive from day one**
- ✓ *Skip long ramp-ups:* Avoid the **6+ month learning curve** of proprietary systems

If you need it – build it

- ✓ Stay in control: Don’t depend on someone else’s roadmap or business case
- ✓ Leverage the community: Share development and maintenance effort
- ✓ Cut costs: Pay only a fraction for new features or components

The Open Source Advantage (2/2)

Innovate faster

- ✓ Deliver continuously: Ship new features and improvements at the pace your customers expect
- ✓ Build on shared progress: Leverage enhancements from a global open-source community

Use what works best

- ✓ Choose freely: Run the most performant, stable, and secure software for your needs
- ✓ Tailor with precision: Include only what you need — avoid bloat and maximize hardware efficiency

Deploy on the hardware you want

- ✓ Stay flexible: Don’t let software vendors dictate your hardware options
- ✓ Use open standards: Leverage Linux VIRTIO support to stay independent of vendor lifecycles

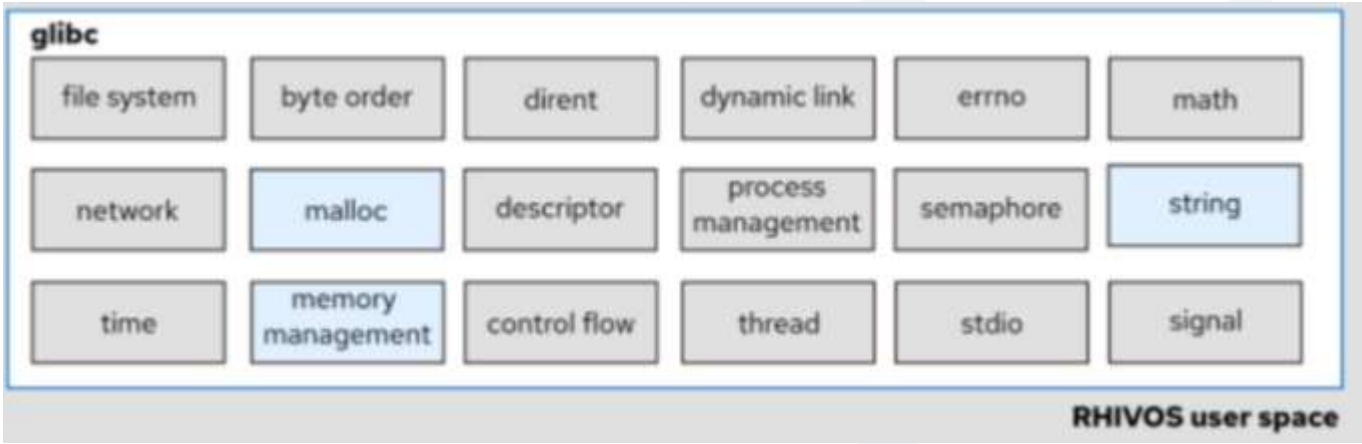
And It Can Run Doom!

With the Help of the Community

The safe application

What about the application?

- If you don‘t want to write your application in assembler you are going to need a **runtime and compiler** for a high-level language.
- Here is what Red Hat has done for **C**, but what about **C++ and Rust**?



Safe

Unsafe

Useful stuff also assessed by Red Hat

- **systemd**: system and service management
- **podman:** container management tool
- **dbus-broker:** interprocess communication
- **mixed criticality:** allow non-safe and safe applications to run side by side within the same operating system

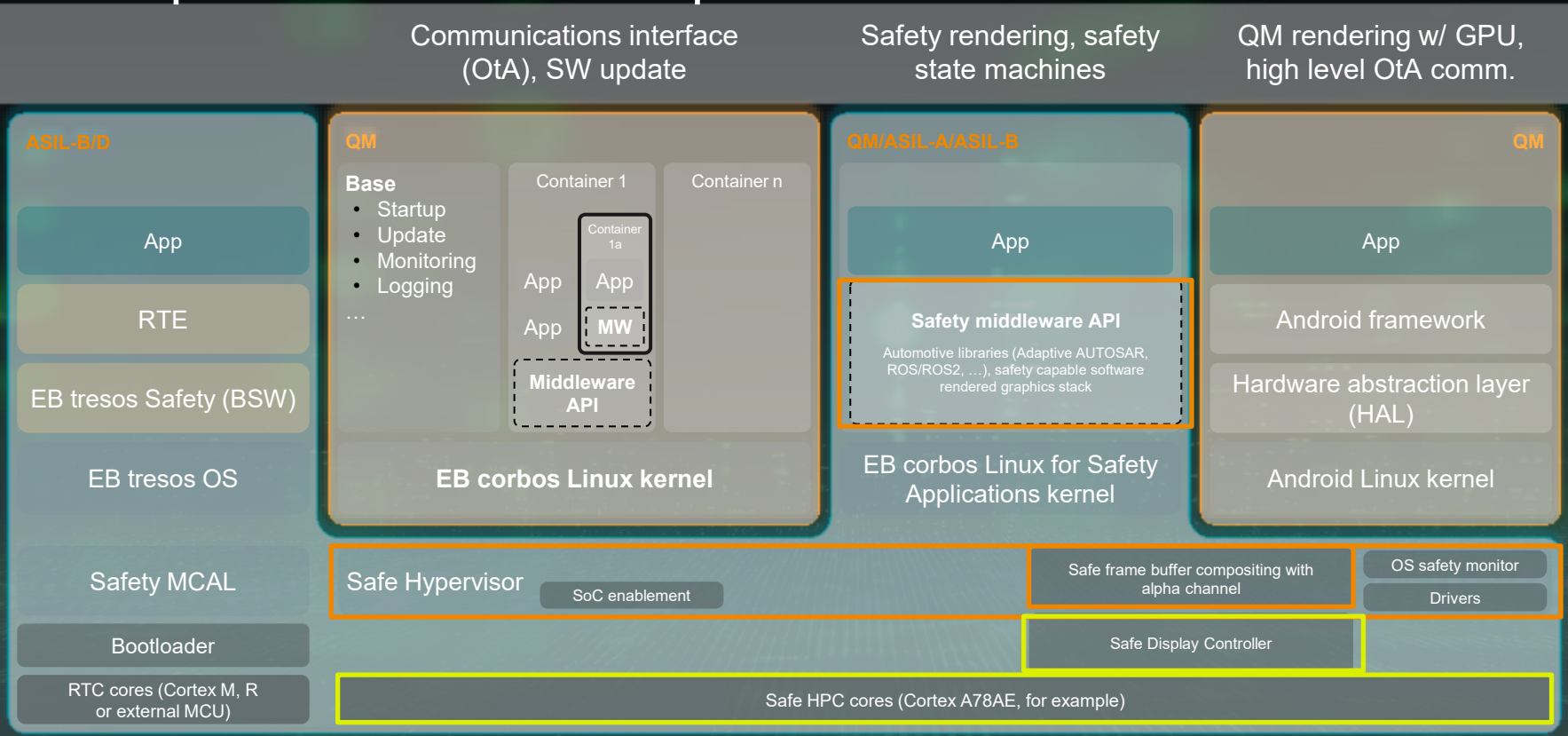
Safe automotive use-cases

Intrument Cluster Tell-Tales ASIL-B

Use-Case

- **Function:** Notify the driver of important events or vehicle malfunctions.
- **Safety Requirement**: The driver must be informed if the system is incorrectly displaying tell-tales.

Example architecture cockpit/IVI for SDV



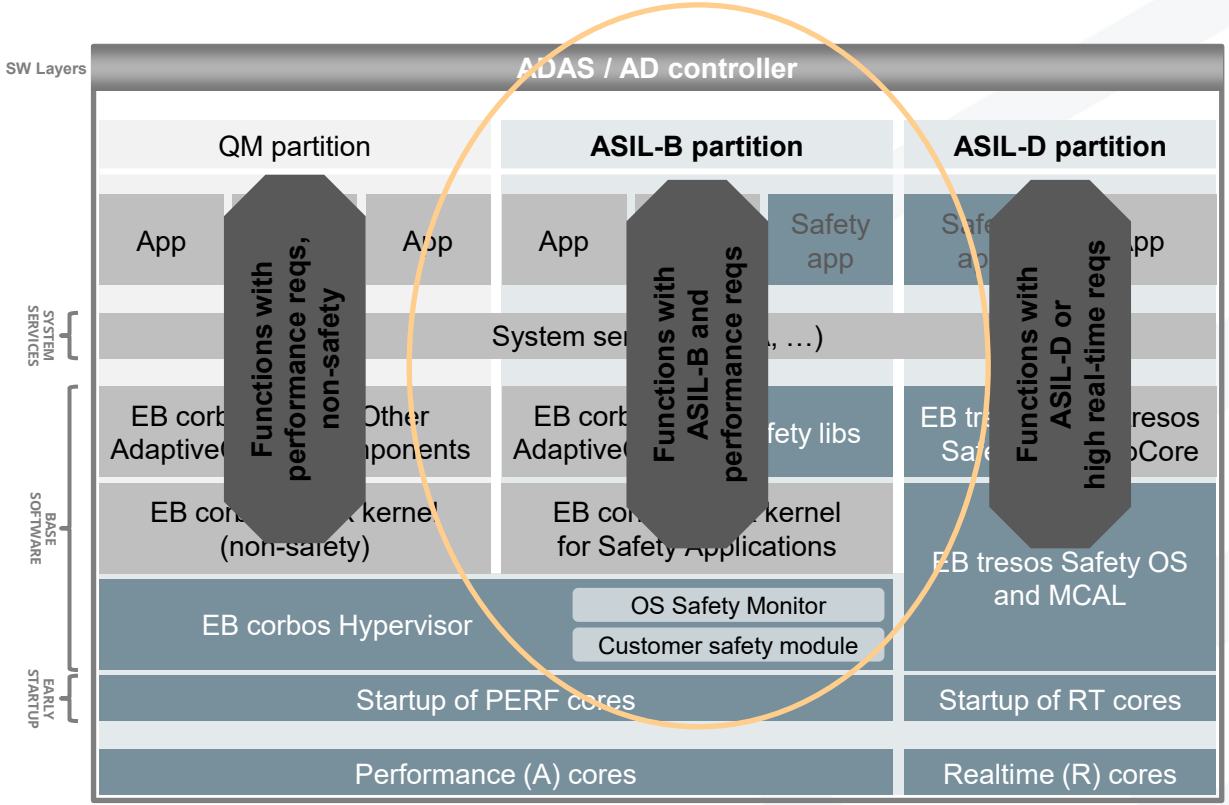
Collision warning; ASIL-B

Use case

- **Function:** Detect a potential collision and lightly apply the brakes. The driver then has the choice to either continue braking or override the system.

- **Safety Requirement:** The driver must be informed if the system is not functioning correctly or bring the vehicle into the safe state; The most likely cause is a sensor issue (camera, LiDAR, etc.)

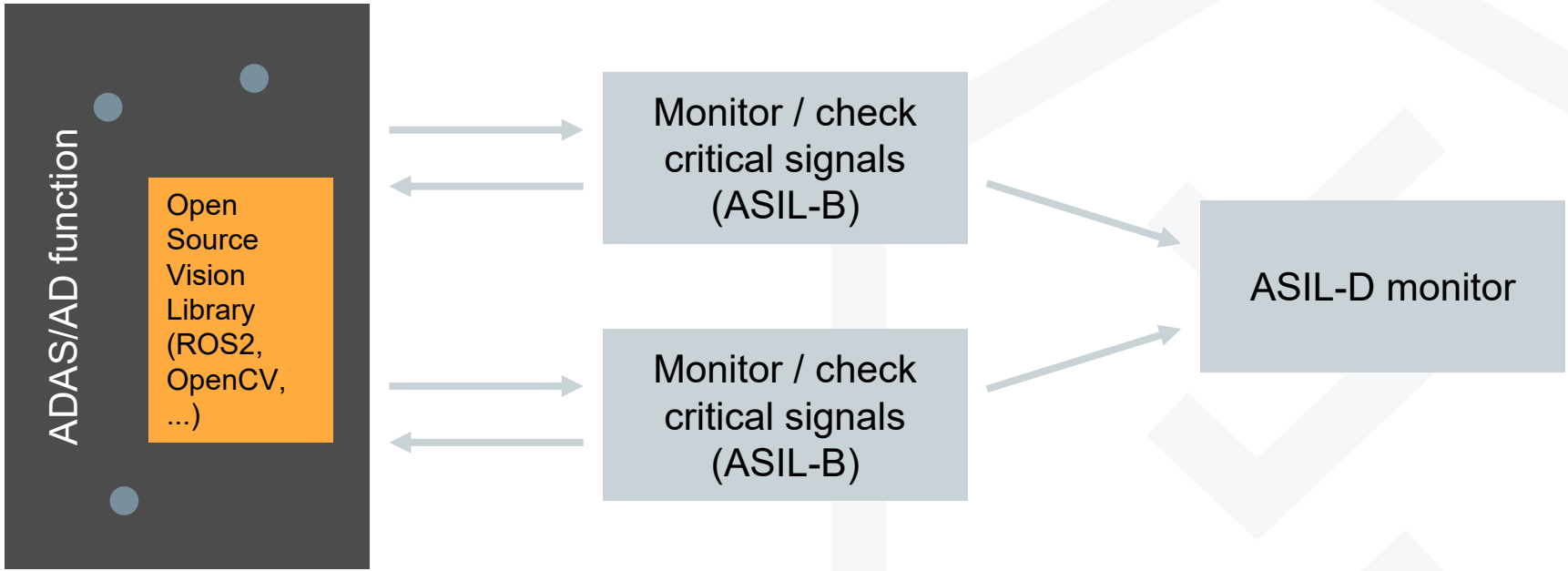
ADAS / AD controller architecture



ADAS Applications: Make it Safe

You need safe versions of big, FOSS libraries!

ADAS Applications: ASIL Decomposition



Add only few components that just ensure you can prove you fulfil your functional safety requirements.

But now you need performant and Safe Inter-process Communication!

Conclusion

- Linux has many advantages over proprietary, safe operating systems BUT
- Potential customers already have solutions that they know work! Switching to Linux is a risk. Just being safe isn‘t enough.
- The more problems the ELISA community can solve up-front, the faster the adoption of Linux for safe applications will be.
- Simply proving the suitability and compatibility of existing solutions and storing in a knowledge base will often be enough.

ELISA
Enabling Linux in Safety Applications

WORKSHOP

Thank you!

Isaac Trefz

Elektrobit – Our Software Moves the World
Isaac.trefz@elektrobit.com