ELISA
Enabling Linux in
Safety Applications

WORKSHOP

ELISA Workshop
Munich, Germany

Aqib Javaid
System Architect, Elektrobit

November 18-20, 2025

HYPERVISORS
ARE BAD
spooky

Hypervisors are scary, So why use them for enabling Linux for Safety Applications!!!

What exactly are our needs?

An Operating System which is:

1) Safe (Functional Safety)
2) Secure (Cybersecurity)
3) Open Source (no vendor locking)
4) Standard toolchain and Interface Support (Scalable)
5) Widely used:
   I. There is large pool of developers
   II. Many Hardware BSPs are already available
   III. Hardware vendors port their hardware devices drivers as a sample on it like GPU driver
   IV. OEMs are already using it to write their Applications
6) Regular updates from open-source community without going to Safety/Security Assessment

Why Hypervisors are Scary?

- They are slow?
    - Even with hardware virtualization support?
- Closed source. And Open-Source variants are not Safety Certified or Security Certified?
- They don’t support running Applications directly on top without a VM?
- They don’t support Standard interfaces like Posix?
- We don‘t like the term/name “Hypervisor”?

Hypervisors are Slow

- Hardware Virtualization Support
  - Separate Execution Layer
  - Separate set of System Registers
  - 2-Stage Memory Management Unit
  - Virtualized Interrupt Controller
  - Virtualized Devices

Open-Source Hypervisors are not Safe and Secure

- Common misconception is that only closed source Hypervisors are Safe
- EB corbos Hypervisor which is based on open source L4Re Hypervisor is already Safety Certified according to ISO26262 and BSI grants German GEHEIM approval to L4Re Hypervisor as well.
- Xem Hypervisor which is also open source is in process of getting Safety Certification

Always requires a VM setup for Applications

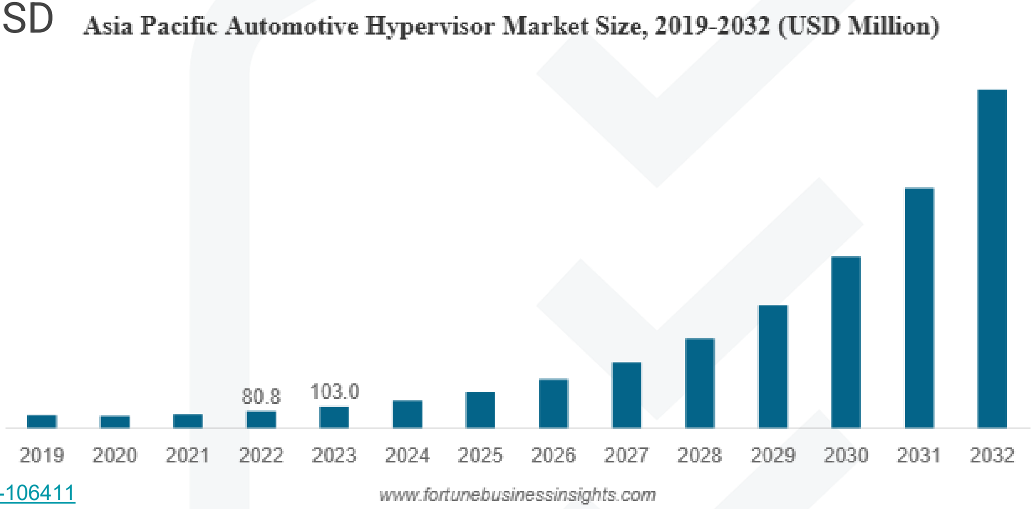
- Another misconception is that Hypervisors always needs a VM to even run small Applications.

- Well, that could be true for some Hypervisors but not for all!
- Microkernel based Hypervisors can run directly Applications without any need of a VM with a small runtime environment.

Hypervisor usage in Automotive Industry

- The global automotive hypervisor market size was valued at USD 314.5 million in 2024

- The market is projected to grow to USD 3,863.6 million by 2032



Source: https://www.fortunebusinessinsights.com/automotive-hypervisor-market-106411

ELISA Enabling Linux in Safety Applications

WORKSHOP

Hypervisor usage in Automotive Industry

- Already where in Automotive Industry?
  - Modern automotive high-performance computing (HPC) platforms are very powerful; they have a lot of physical cores to run things in parallel
  - Major Car companies utilize Automotive Android (AAOS) in their In-Vehicle Infotainment (IVI) system and want to run separate operating system in parallel
  - In Gateway controller there is need of having role specific separate Operating Systems (Sometimes from different Vendors)
  - Hypervisor is already used in Automotive Industry to provide VM(s) level Isolation for fulfilling the needs of running AAOS and other role specific Operating Systems at same time on a single HPC platform
- Why?
  - To save cost by sharing hardware resources

Hypervisor usage in Automotive Industry

- Future of Hypervisor in Automotive Industry?
    - Huge demand of running ADAS (Advanced Driver Assistance System) Applications which have safety requirements in an operating system as a separate VM in parallel to Android Automotive on same HPC platform
    - Another example: Display Controller for Instrument Cluster is hosted by Android OS and Telltales as an overlay on this Display Controller controlled by a Safe Application

Hypervisor usage in Automotive Industry

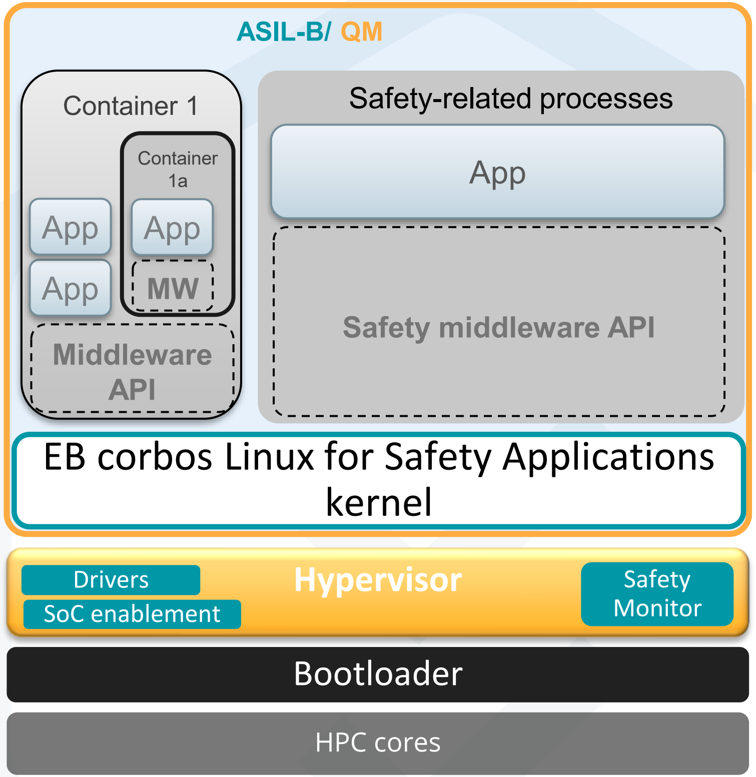
- Now we have established that Hypervisor(s) are (going to be) essential part of Automotive Software in HPC Domain

- Then why not use it (or extend its usage for Linux for Safety Applications)?

Approach for qualifying Linux for Safety Application(s)

The solution consists of two main software elements:

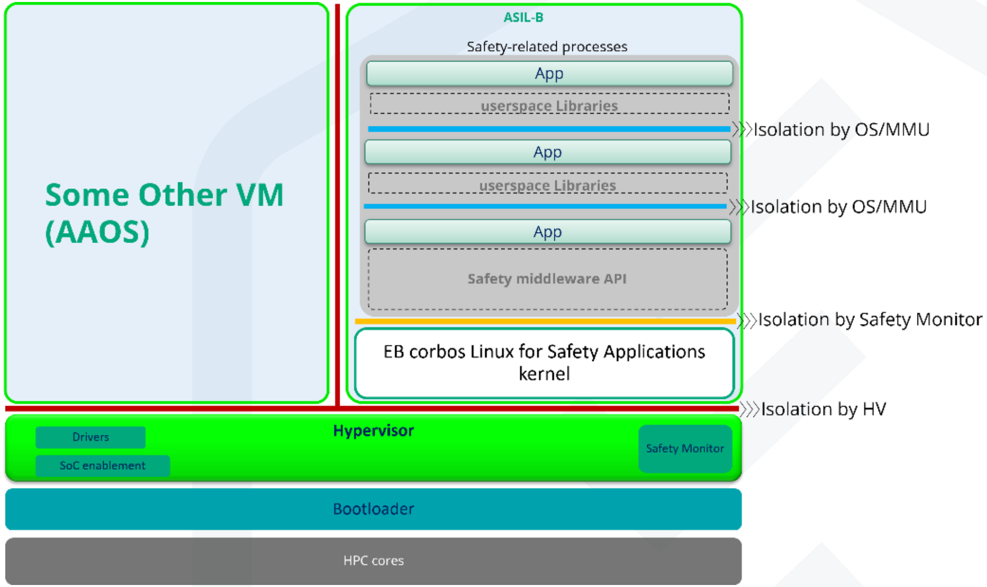
- A hypervisor that provides virtualized memory and computation resources, giving it full control over access to these resources by Linux. It duplicates address spaces and data objects for the VMs and ensures correct loading and initialization of memory segments for Safety Apps.
- A supervisor software layer that monitors the interface between safety applications and the Linux kernel, detecting any attempts by Linux kernel to access memory or computation resources that could adversely affect the dependability of the system's functions.



Approach for qualifying Linux for Safety Application(s)

Create another Isolation layer between OS Kernel and userspace Safety Application(s)

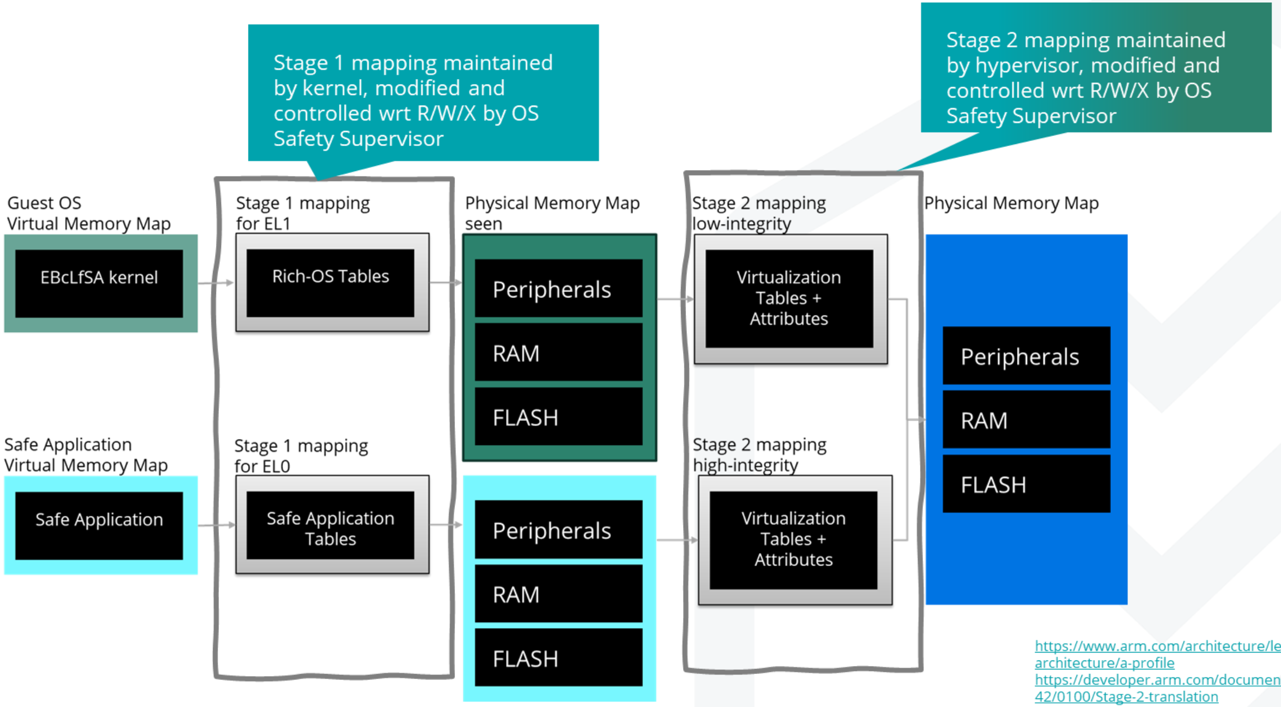
- Supervise special cases of access in safety monitor
- Detect Access violations



ELISA Enabling Linux in Safety Applications

WORKSHOP

Approach for qualifying Linux for Safety Application(s)



Stage 1 mapping maintained by kernel, modified and controlled wrt R/W/X by OS Safety Supervisor

Stage 2 mapping maintained by hypervisor, modified and controlled wrt R/W/X by OS Safety Supervisor

Guest OS Virtual Memory Map

EBcLfSA kernel

Stage 1 mapping for EL1

Rich-OS Tables

Physical Memory Map seen

Peripherals

RAM

FLASH

Stage 2 mapping low-integrity

Virtualization Tables + Attributes

Physical Memory Map

Peripherals

RAM

FLASH

Safe Application Virtual Memory Map

Safe Application

Stage 1 mapping for EL0

Safe Application Tables

Peripherals

RAM

FLASH

Stage 2 mapping high-integrity

Virtualization Tables + Attributes

https://www.arm.com/architecture/learn-the-architecture/a-profile
https://developer.arm.com/documentation/102142/0100/Stage-2-translation

■ SOC

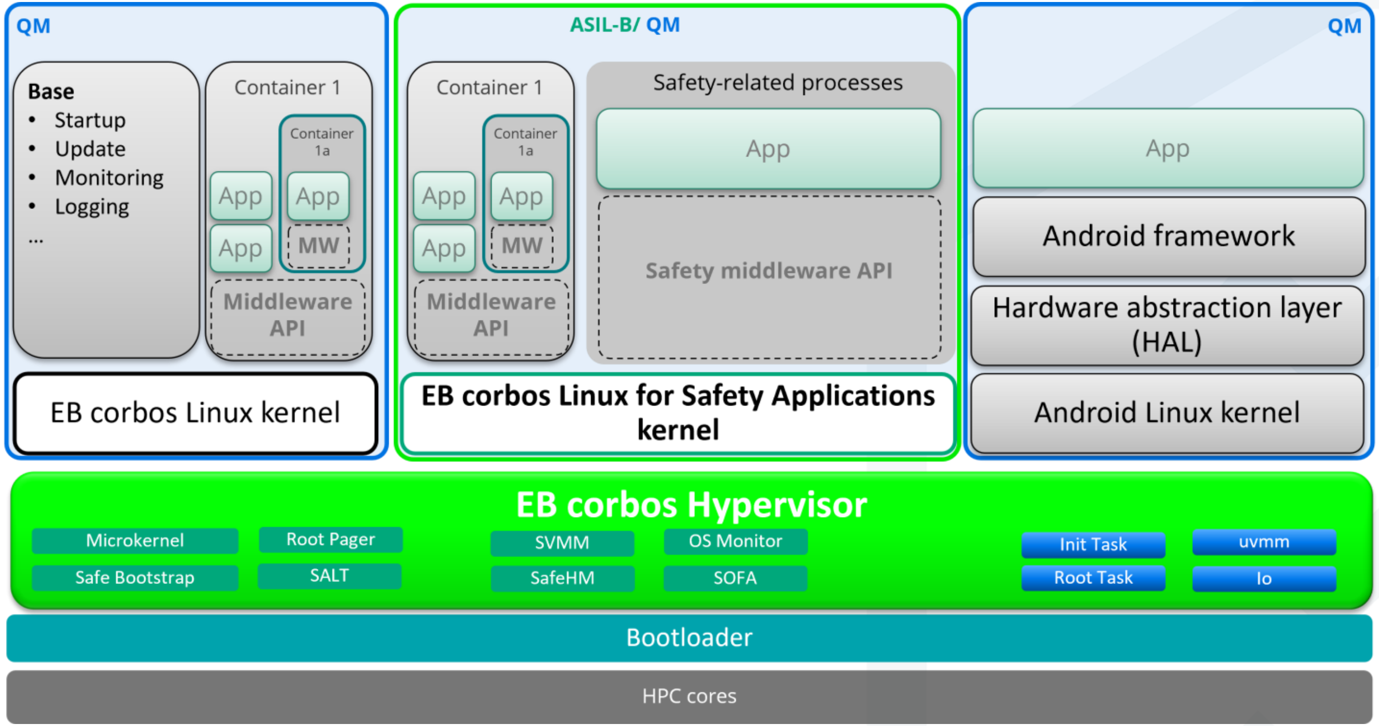
■ Safe App

■ Hypervisor

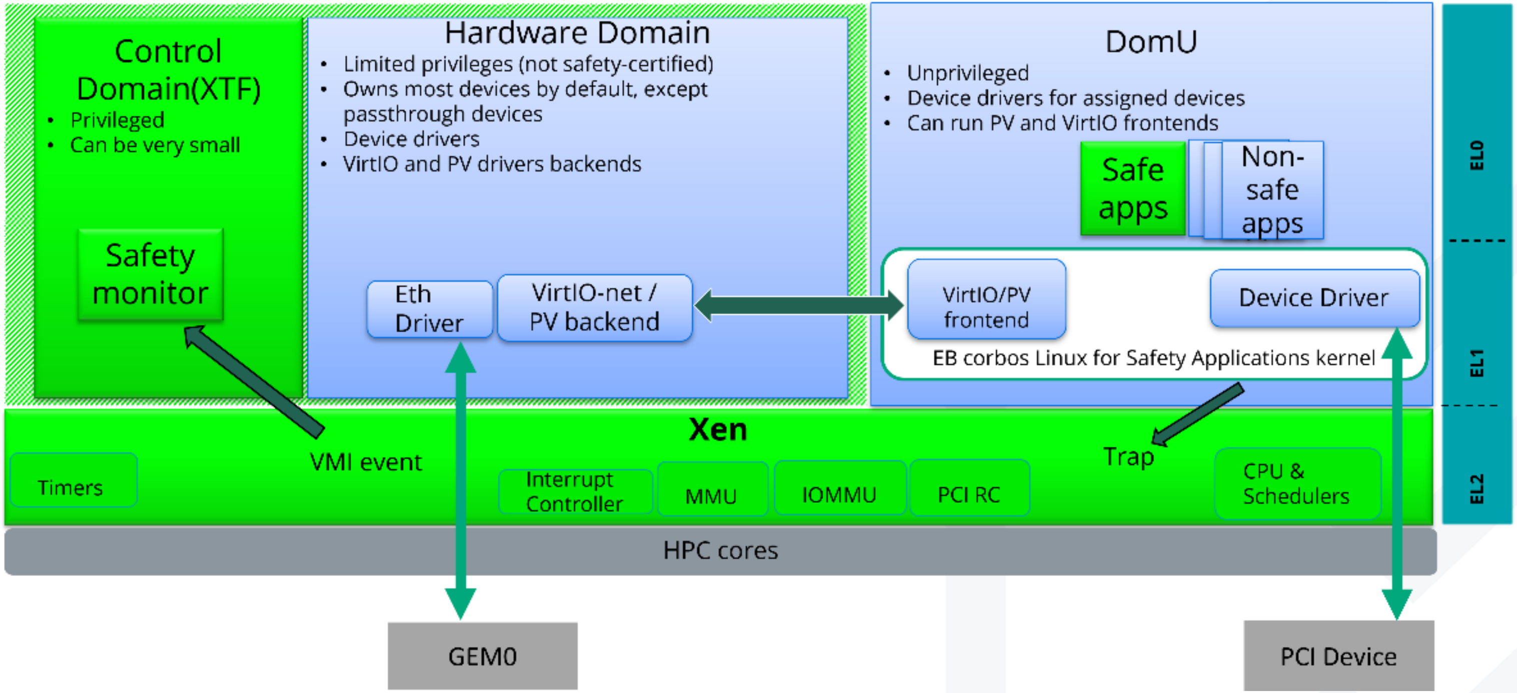
■ EB corbos Linux for Safety Applications kernel

■ OS Safety Supervisor

EB corbos Linux for Safety Applications



EB corbos Linux for Safety Applications



Work in Progress - License: CC-BY-4.0

Requirements on Hypervisor for Linux for Safety Application(s) Solution

- Ability to have duplication of VM Address spaces on the level of stage-2 mapping
- Ability to have duplication of data-objects for vcpus

- Able to host OS safety Monitor (either as a Hypervisor App or as part of its kernel)
- Provide interface and allow OS safety Monitor to update the stage-2 VM Address spaces mappings
- Allow OS safety Monitor to access some HW system registers
- Forward required VM exceptions to OS safety Monitor

- Provide a Safe Health Monitoring Mechanism for safety applications (e.g., Health Monitor)

Questions?