Functional Safety and Linux

Maxime Ripard
Senior Principal Software Engineer

Red Hat

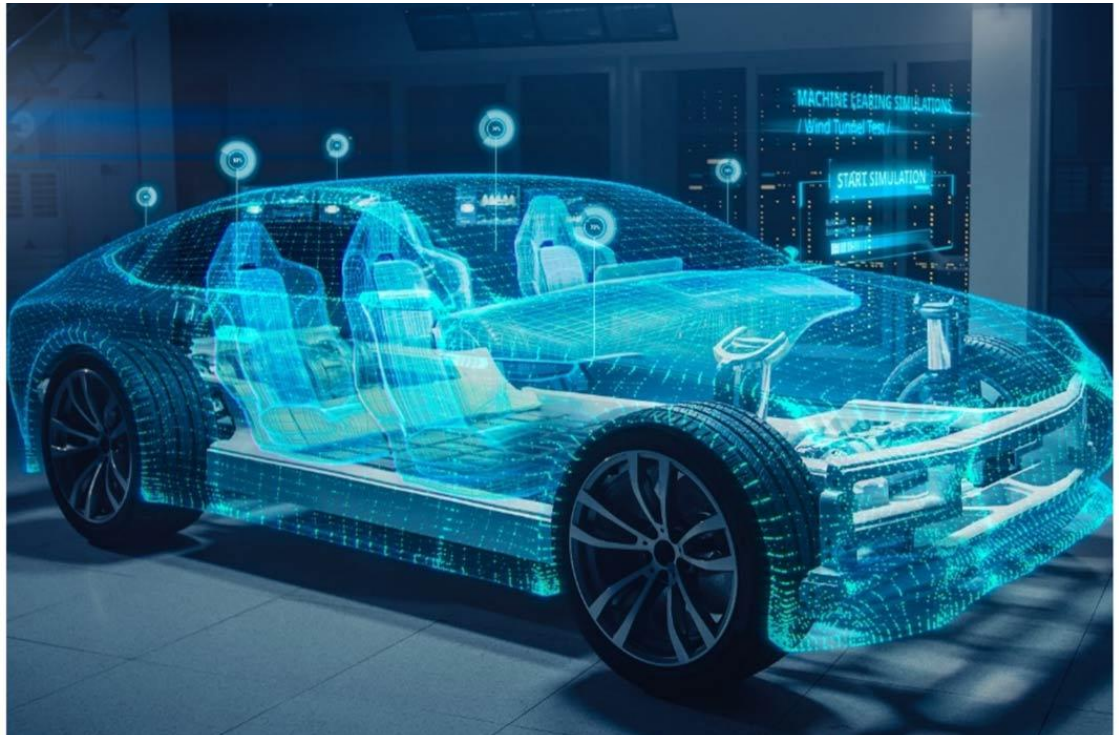
The Automotive Software Revolution

A Car in the 80s

- No ABS, no Electronic Stability Program (ESP), no active suspension
- Thermal Engine, no injection, no turbo, 60bhp on a good day.
- No Airbags
- No Advanced Driver Assistance Systems (ADAS)
- No In-Vehicle Infotainment (IVI) (radio and cassette player in option)
- Basically no software

Source:
caradisiac.fr

A Car Now (Allegedly)



- Hybrid or EV
- ADAS: ESP, lane keeping, adaptive cruise control, park assist, autonomous driving, etc.
- Airbags, Emergency Braking, etc.
- Can run Doom (or navigation, whatever)
- Software is basically everywhere

Source:
cadence.com

(Mixed) Criticality

- Not all features are equally critical
- Some are life-critical (braking), some are just about comfort (radio)
- Industry shift from multiple processing components to a central one
  - COVID Chip Shortages
  - User now expects the system to be updated regularly
  - Margins!
- That component will have to handle various criticality levels

ISO 26262 Functional Safety For Road vehicles

Enter ISO 26262

- Ratified in 2011, revised in 2018
- Apply to all road vehicles but mopeds
- Considered an industry standard, but not mandatory
- Only deals with functional safety, ie. making sure that electronics behaves as it should
- Does so by introducing risk levels and associated requirements
- Classification based on the severity of the consequences of a defect, the probability of it occurring, and the probability of the driver or a passenger mitigating it.

Automotive Safety Integrity Levels (ASIL) Criterias

- Severity (S0 to S3)
  - The severity of injury a defect could cause, from no injuries (S0) to life-threatening or fatal injuries (S3)
- Exposure (E0 to E4)
  - The expected frequency of an injury, from incredibly unlikely (E0) to high (E4)
- Controllability (C0 to C3)
  - The likelihood of the driver preventing the injury from controllable (C0) to difficult to control or uncontrollable (C3)

ASIL

- ASIL-D: Potentially Fatal (S3), High Probability of Injury (E4), Uncontrollable (C3)
- Every reduction of any criteria brings the level down by one, down to ASIL-A
- Below ASIL-A is Quality Managed (QM)
- QM means that all risks are tolerable from a safety perspective. Standard development practices are sufficient.

ASIL (cont.)

- ASIL-D: Total loss of braking
- ASIL-C: Cruise Control, Loss of rear braking
- ASIL-B: Head Lights, Brake Lights
- ASIL-A: Tail Lights
- QM: The Weather widget on the dashboard
- ASIL-C and -D highly recommend formal methods, and require verification and validation.
- Anything below is less constrained

Freedom From Interference (FFI)

- The “absence of cascading failures between components that could lead to the violation of [some] safety requirement.”
- Spatial Interference: one task affects the memory of another
- Temporal Interference: one task affects the execution of another
- Resource Interference: one task affects a resource shared with another task, or its access to it

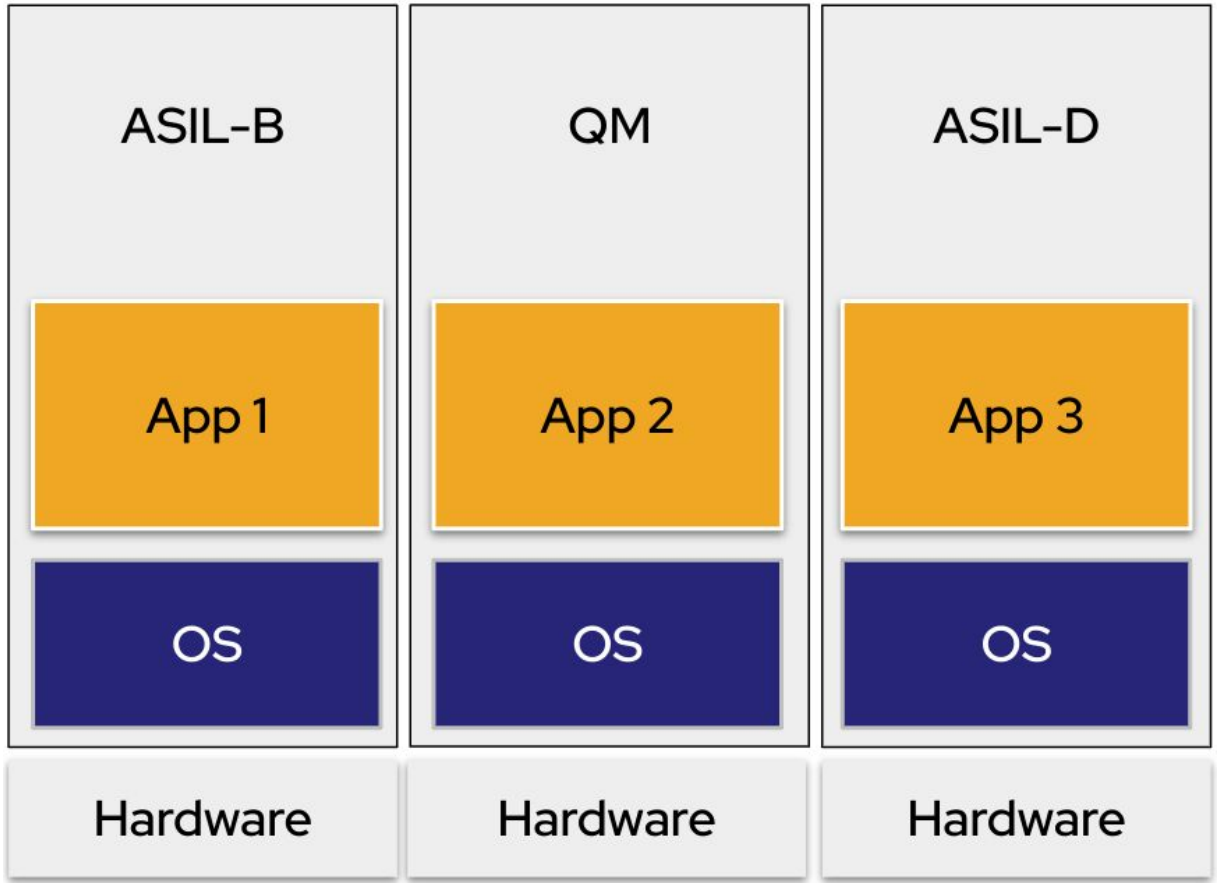

WHEN YOU REALIZE FFI
IS JUST GOOD OS DESIGN
imgflip.com

ISO 26262 Implementation

General architectures

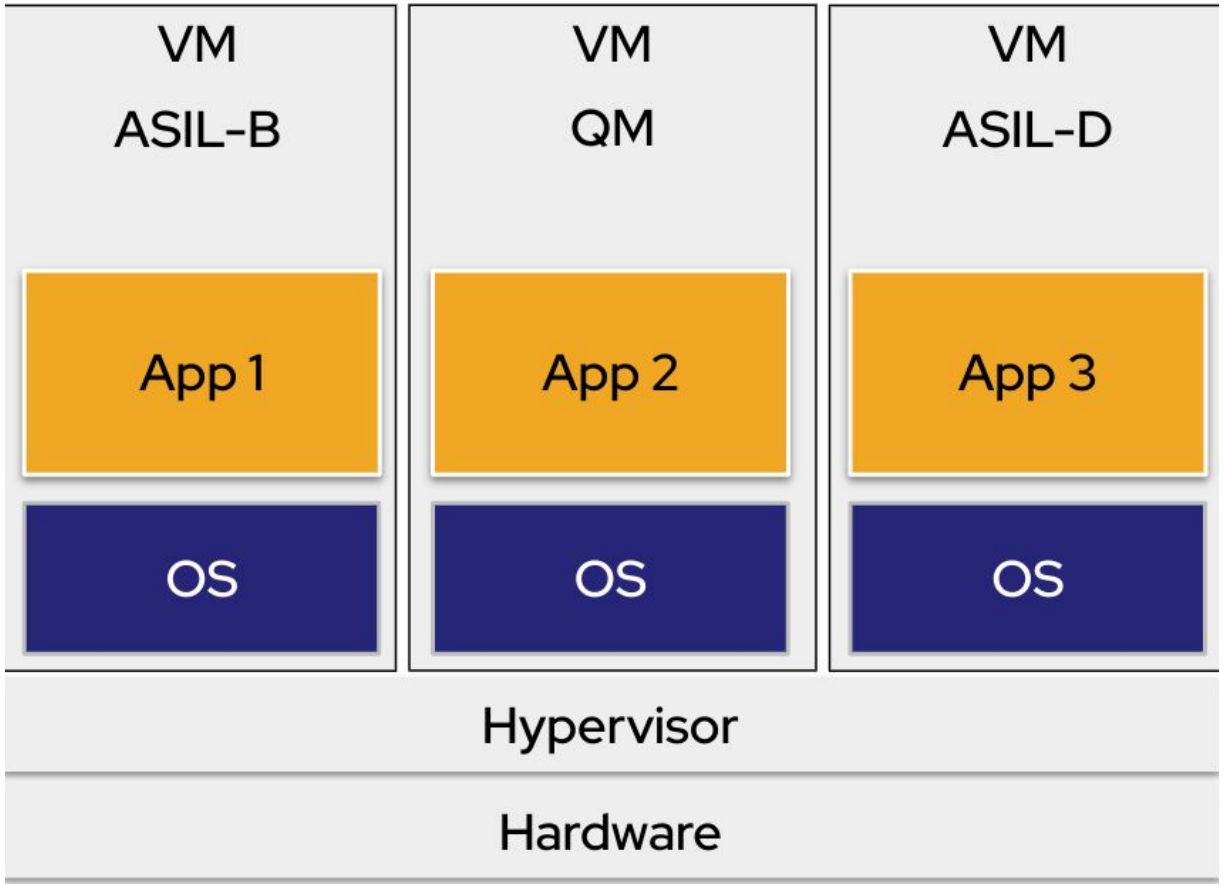
- The FFI concept falls nicely into the age-old concept of CPU and memory isolation
- Different takes on it:
  - Discrete Physical Devices
  - Heterogeneous Systems
  - VMs
  - Containers
  - Process sandboxing

Discrete Devices



Source:
sig.centos.org/automotive

VMs

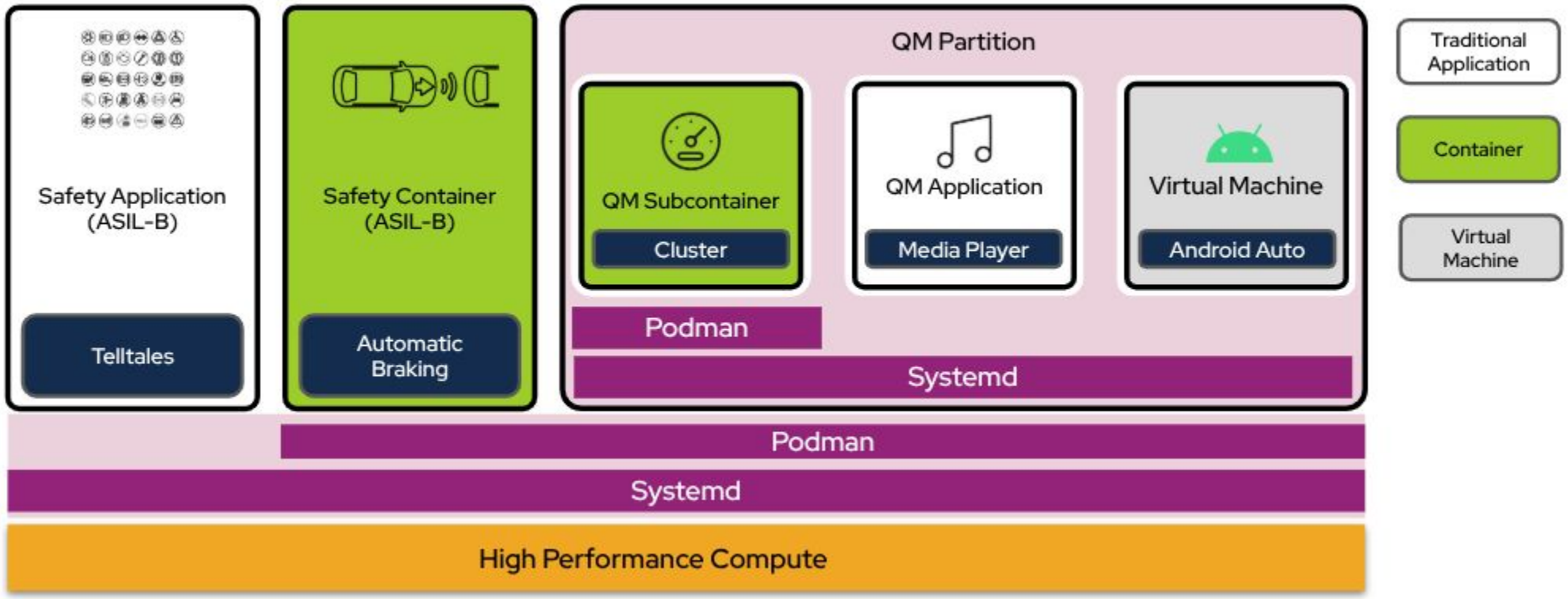


Source:
sig.centos.org/automotive

Doing it with Linux?

- Temporal Interference?
  - Scheduler, PREEMPT_RT, cgroup, etc.
- Spatial Interference?
  - Process Address Space, cgroup, containers, etc.
- Resource Interference?
  - Partitioning, QoS, Arbitration, etc.
- Plus usual issues for embedded devices
  - Software updates, secure boot, boot time, etc.

Doing it right



Source:
sig.centos.org/automotive


ONE DOES NOT SIMPLY
CLAIM THEY ARE ASIL-B
imgflip.com

Getting Certified

- The certification is made by an authority
- Designing a robust system is only the first step
- You also need to show the authority that the design is indeed robust, doesn't have any gap, is reviewed, tested, documented, etc.
- The certification attestation is then published for a given version

Missing Pieces

- There's still some parts of upstream Linux that don't provide FFI
  - Userspace Buffer Allocations APIs
  - GPU scheduling constraints
  - Clock Framework tree rate changes
- Missing/incomplete features
  - OpenGL / Vulkan SC
  - Fault-Tolerant V4L2
  - Virtualized everything
  - Being able to still display something when the compositor crashed

Questions?

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.


linkedin.com/company/red-hat


youtube.com/user/RedHatVideos


facebook.com/redhatinc


x.com/RedHat

Red Hat