Vehicle Domain Interaction Workshop

Safety Domain interaction, including Hypervisors

October 11, 2017

Gunnar Andersson
Development Lead
GENIVI Alliance

Ralph Sasse
Open Synergy

Background – why are we here?

Solve challenging problems in highly integrated vehicle systems across domains

How?

- **Open code and standards**
- **Interface specifications**
- **Technology and Knowledge exchange**

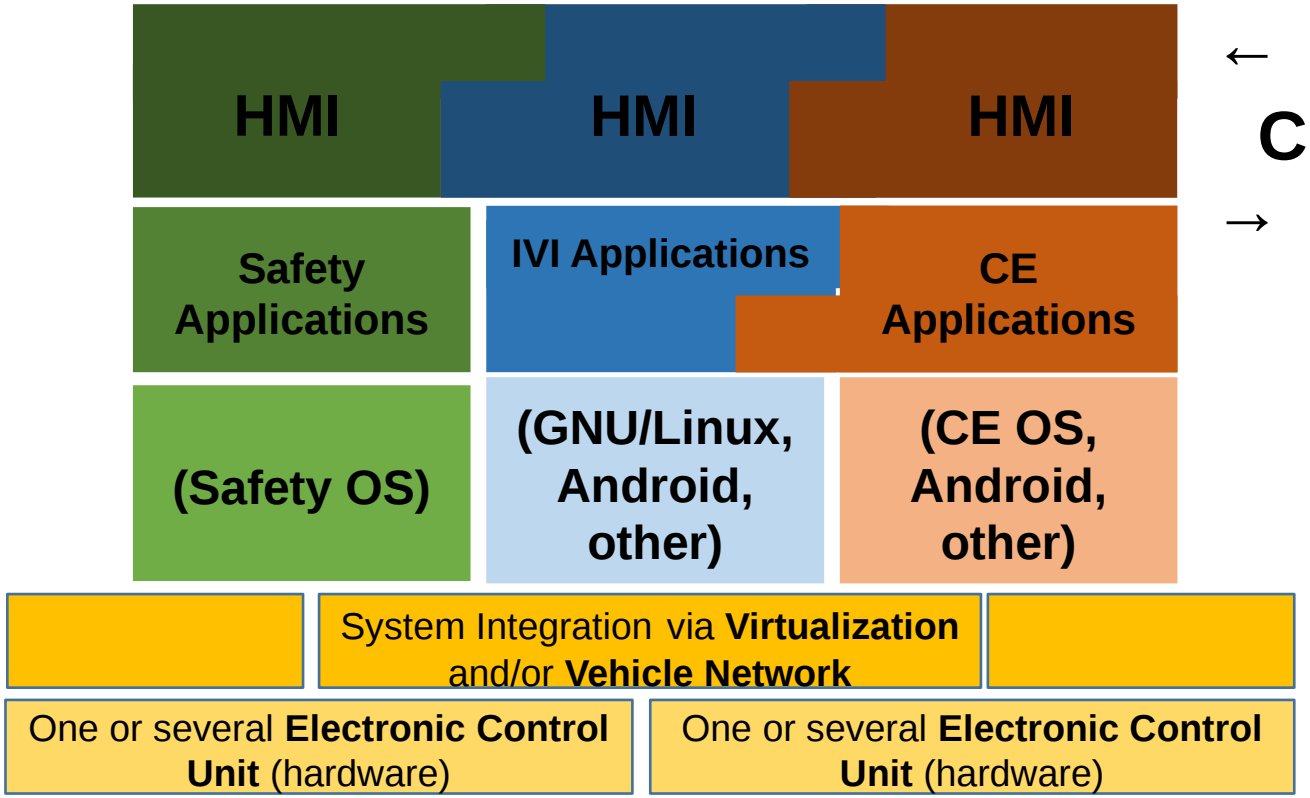
GENIVI®

Real challenge

Complex software integration in the entire vehicle, and the environment around it

GENIVI®

Trend: High integration and Consolidation... with more diversity



← **Connectivity** →

Cloud computing

"Internet Of Cars"

Driving & Safety ↔ Infotainment ↔ Consumer ↔ Connectivity everywhere ↔

GENIVI®

TODAY’s AGENDA:

- 1. Purpose and topic introduction (Gunnar Andersson, GENIVI)
- 2. Quick Walkthrough of **whole slide deck** (to understand scope)
- 3. Hypervisor introduction (Ralph Sasse, OpenSynergy)
  - Typical Hypervisor system architectures
  - Examples with RTOS, Linux and other
  - Designing Mixed-criticality systems
  - The **Functional Safety Challenge** and how Hypervisors help
  - **Security** Challenge & design (intro – not main focus today)
  - Detailed: Mechanisms for device sharing – virtualization approaches for hardware

  4. Discussion

Remember this is a: **WORKSHOP – INTERACTIVE – DISCUSSION**

Ask A Question: How to

NOW

- Quick run through the whole slide deck

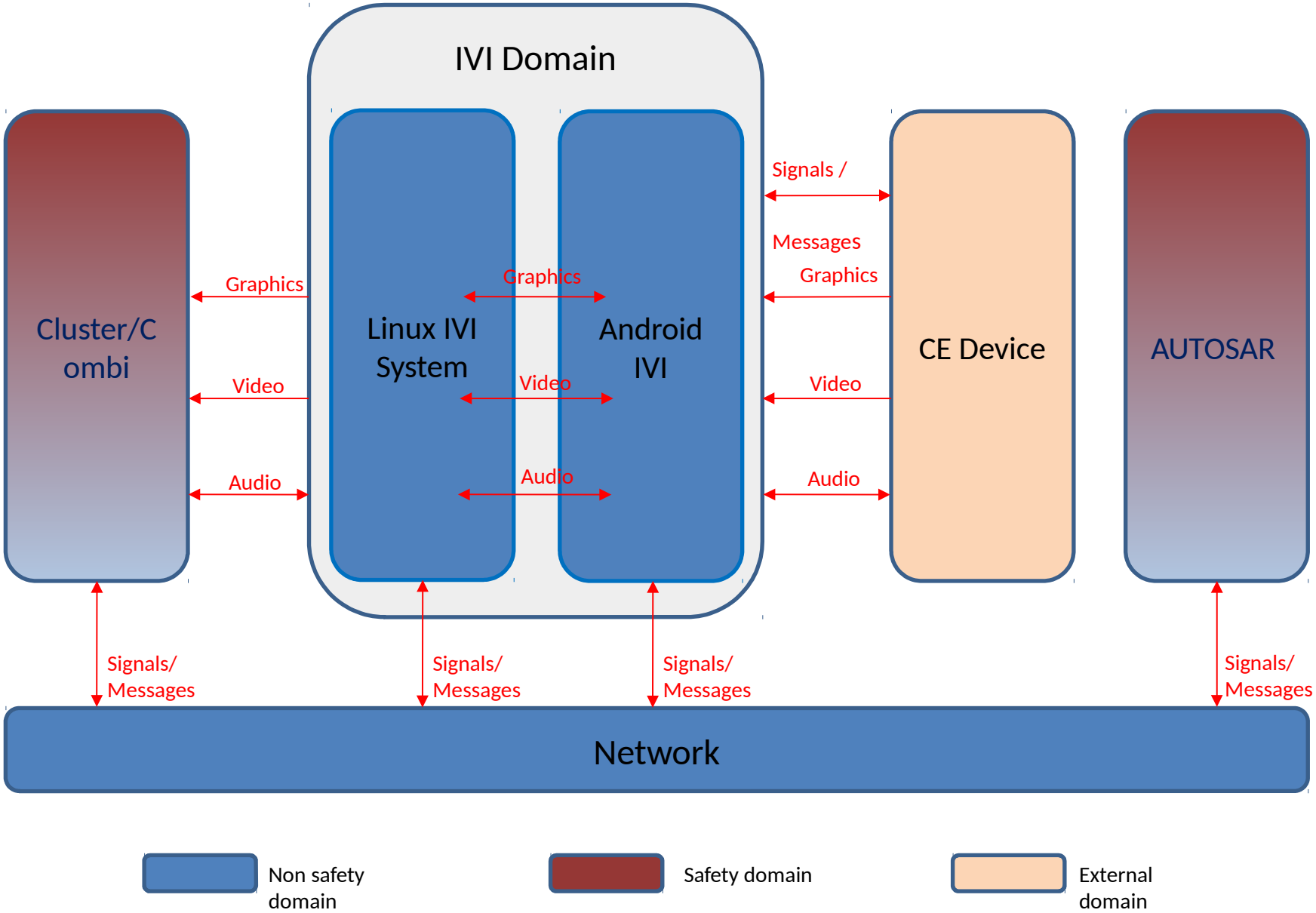
- Remember - we are not discussing the details uet
- Only for preparation

- Please use this as a “trigger” for input questions during workshop – remember which topic you found most interesting.

Open Synergy introduction

- **Come back here after quick slide deck overview.**

Example Architecture – #1 Combined IVI system

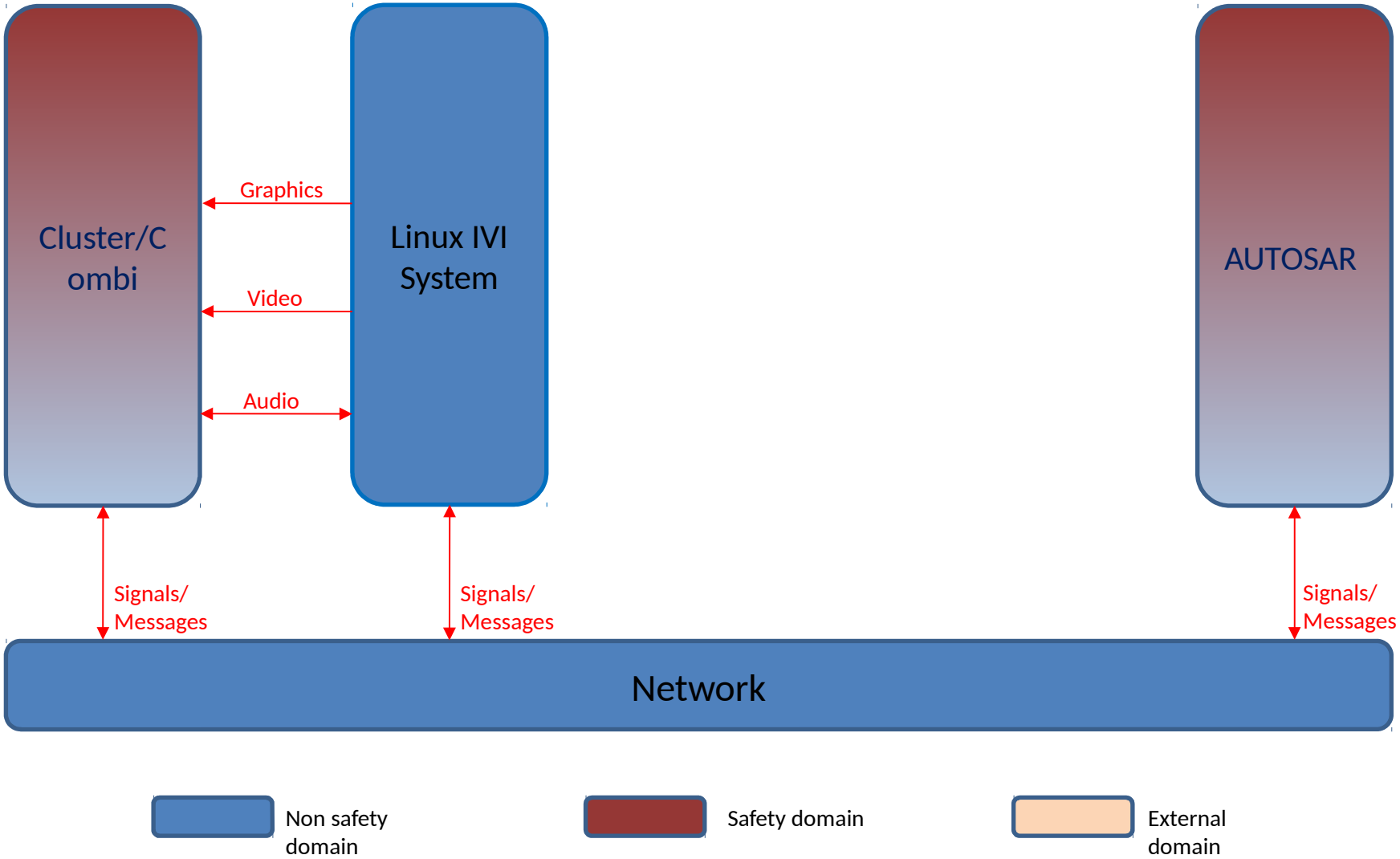


Definitions

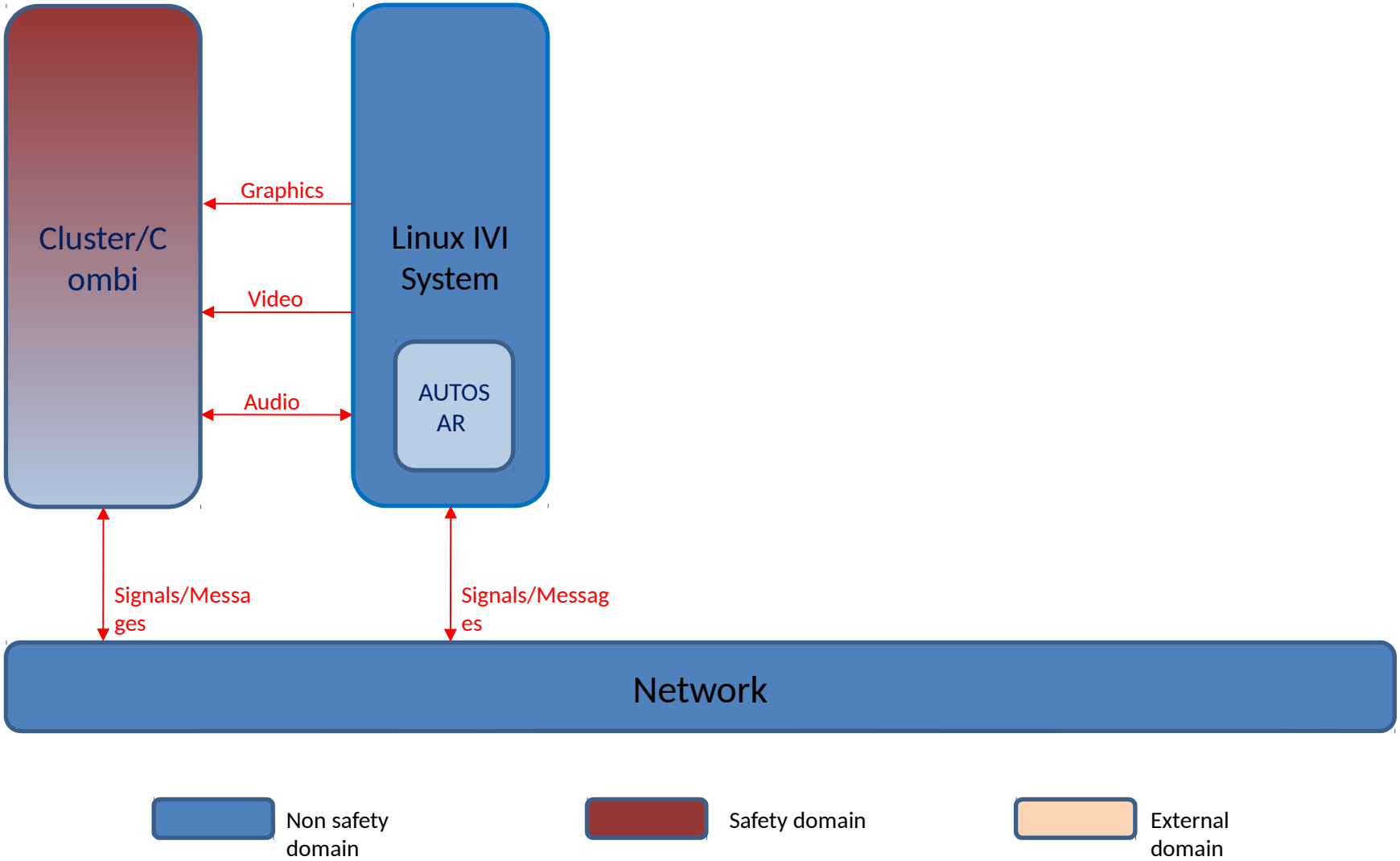
- Graphics: Pictures, drawing commands, OpenGL, Rendering needed
- Video: Streaming Video, Rendering needed
- Audio: Digital stream (PCM, MP3/AAC, …)
- Signal: A named observable data item (property) with defined data type
- Message: Well defined, self-contained, single data transfer with a clear purpose

- Network: Logical network (virtual or physical) inside the vehicle

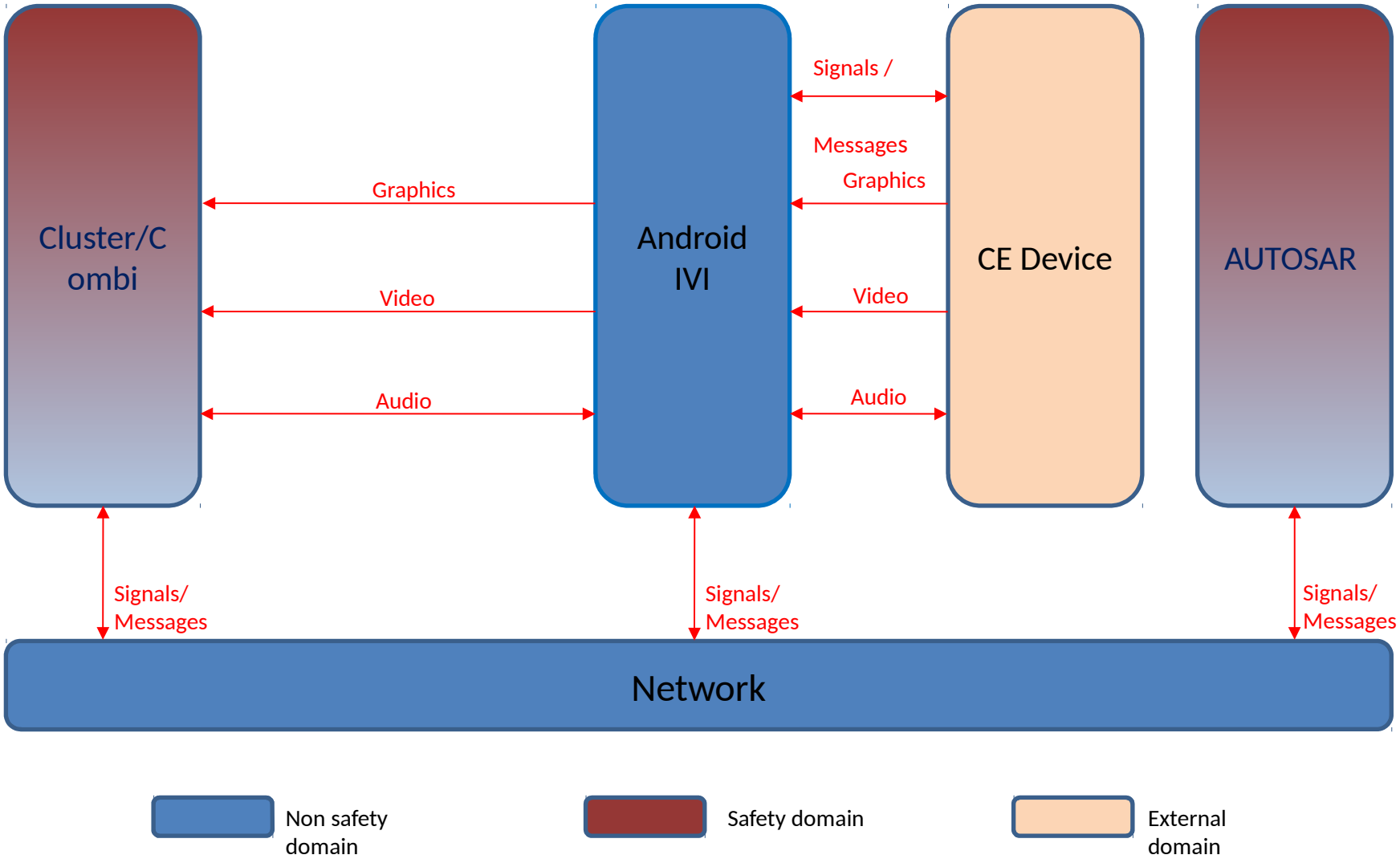
Example Architecture – #2 Linux-only IVI system



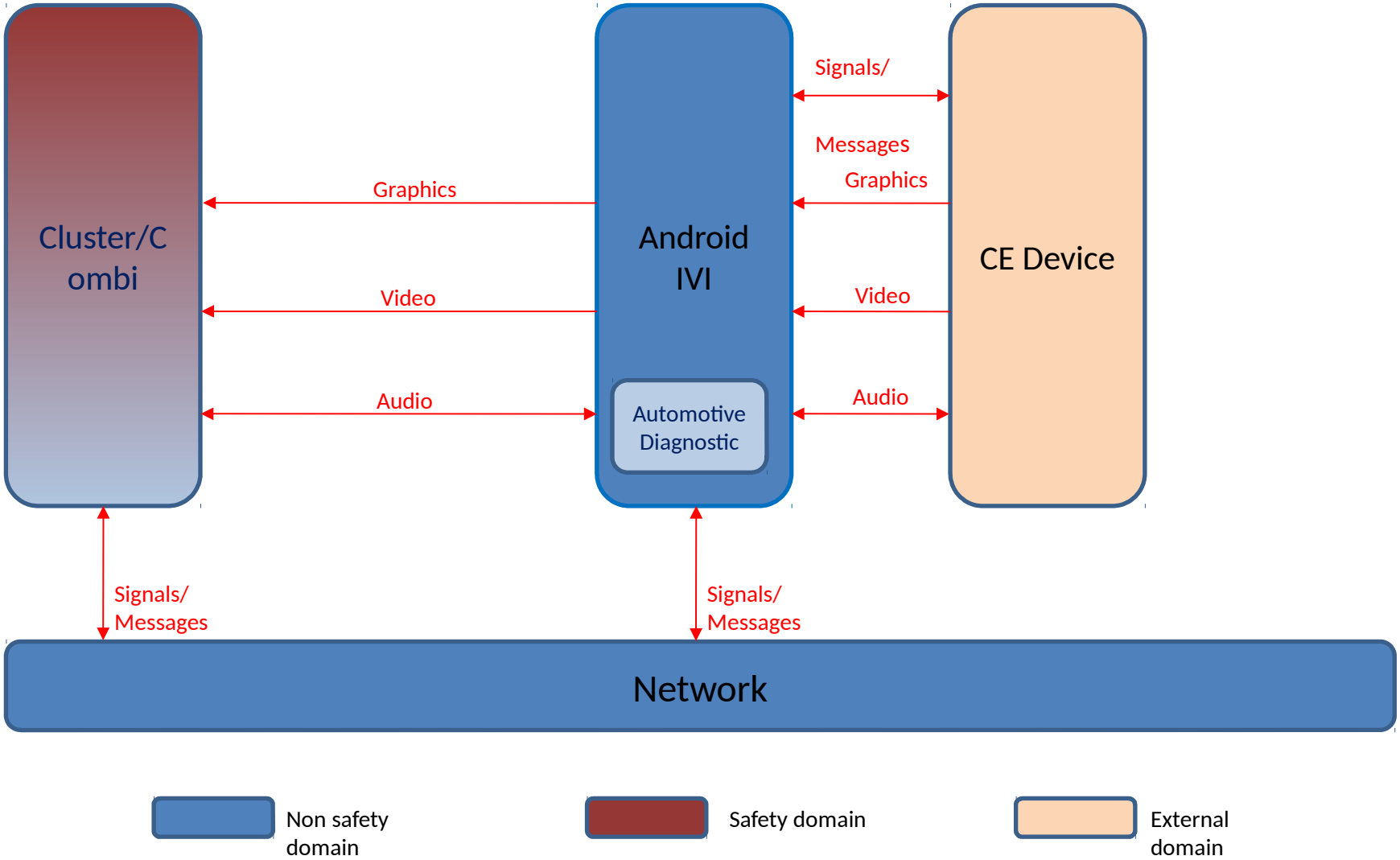
Example Architecture – #3 Embedded AUTOSAR



Example Architecture – #4 Android IVI system



Example Architecture – #5 Android + AUTOSAR



Audio Management:

- Separate Audio Management needed in Safe domain? Conflicts with GENIVI audio manager approach
- Media Apps running in Android require Audio Management.
- We see two different Audio Managers solutions in GENIVI and Android, conflicting with each other. E.g. pausing the Android audio stream needs to result in pausing the App from playing.

Video Management:

- Will become Video management the same complexity as Audio Management, with multiple screens and multiple sources
- Resource Management of Bandwidth and Decoders

Cross domain App management:

- Running state, synchronization. Application management master to control apps across domains

Cross domain data exchange

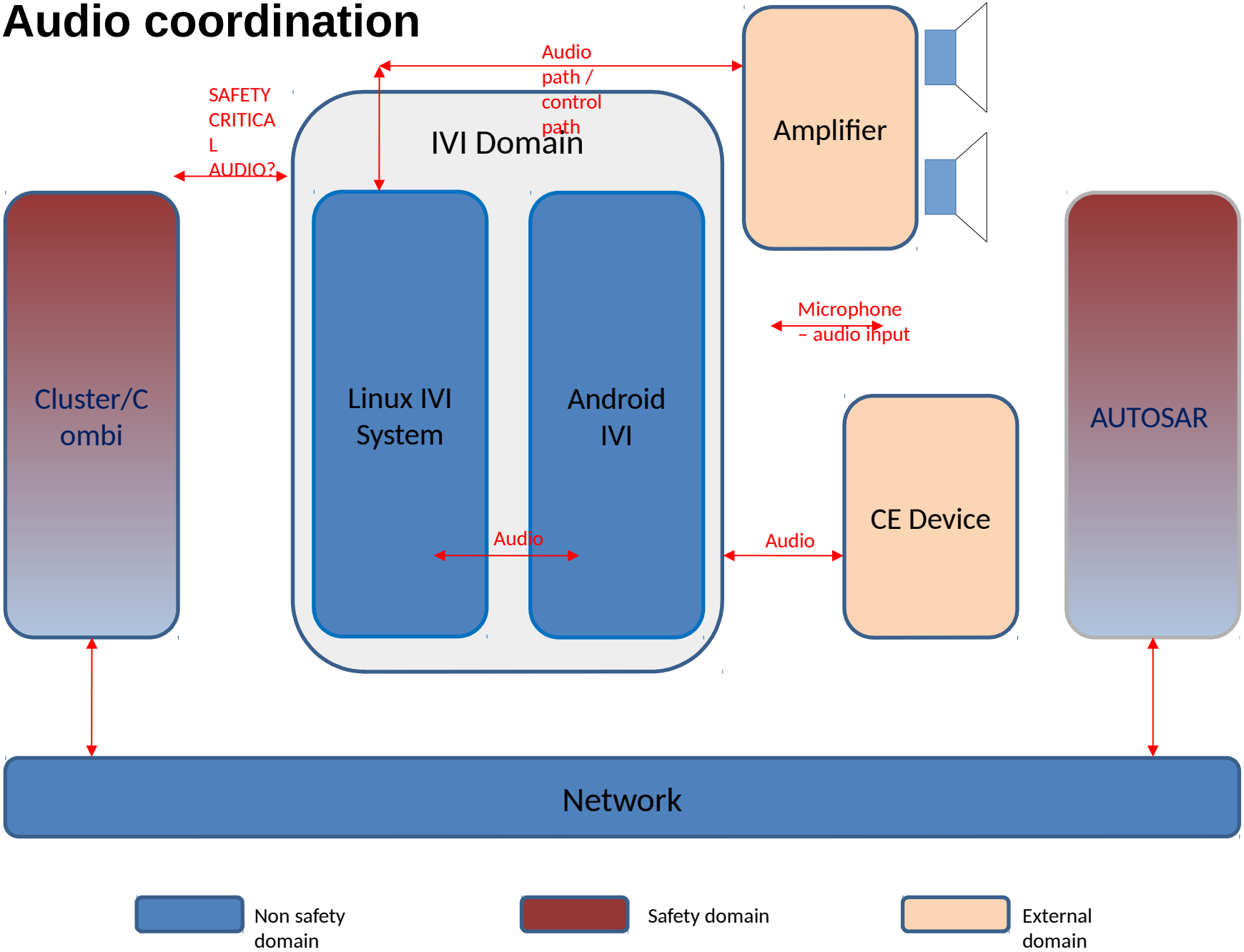
Application proxy for intents between services and apps in different domains

Audio Use cases

- Audio streaming to any non safety domain like Spotify, Deezer, Internet radio
- Audio playback via device mediaplayer (e.g. mp3)
- Handsfree speeking to/from CE device
- Text to Speech
- Navigation speach outpout from CE Device navigation (e.g. Google Maps)
- CE Device voice assistent (Siri, Bixbi, Google, Alexa, ...)
- Audio Notifications
- Audio mixing (standard audio management)

- Audio Safety Domain relevant
- - warning sound (chimes)

Audio coordination



Discussion points – Safety #1 (Hypervisor focus)

Examples of typical Safety requirements / use cases with attached Safety Requirements

Safe hardware sharing with Hypervisors
- What's the problem?
- What's the solution?

Setting the stage for the higher-level protocols

- Interfaces between VM / Hypervisor
  - Virtual networking "sockets" – is that all there is to it? (Is a hypervisor architecture *identical* to a networked distributed system?)
  - Hardware sharing interfaces?
  - What else?
- Building the design on top of Hypervisor APIs

Discussion points – Safety #2 (Networked systems)

- Designing safe network protocols
  - What exists, and what (maybe) needs to be created?
  - Robustness guarantees
  - Delivery guarantees?
  - Diagnosing of errors
  - Gateway nodes – are they safety critical?
  - Can “safe” communication go through a non-certified network / ECU (e.g. challenge/response design)

Video Use cases

- User function:
  - Video and graphics streaming to any non safety domain like Youtube
  - Video and graphics distribution to cluster domain like (navigation map, turn by turn navigation, Album art, ...)
  - Toplevel compositing across domains for the same physical display with or without mixed safety critically

- **Video coordination**
- Only a matter of resource management (# codecs, bandwidth)
- Or is there more...

HMI Compositing

- Toplevel compositing across domains for the same physical display with or without mixed safety levels

- Linux: Distributed Wayland setup
  (e.g. Waltham protocol)

- What can we do across multiple systems?
- e.g. Linux to QNX?
- Is there another level of protocol needed, not like Wayland?

Distributed shared states

- Distributed Lifecycle state
  (Wakeup, Boot state, Diagnostic & Software update states, Power management, Temperature Management, …)

- Reuse GENIVI NodeStateManager

  - Today: Single node (D-Bus)
  - Next: Multiple nodes (networked)
  - Same protocol is likely applicable!

- Related: Boot sequence control
- Especially: On Virtualized system, use resources for prioritized node.

Distributed shared states (2)

- Supervisor processes & watchdogs

(Distributed) Resource Management

- Sharing resources, e.g. Bandwidth, Interfaces,

- Multiple domain using single HW (USB, Bluetooth)

- Prioritization (e.g. Real-time requirements)

•

• **Messages**

  - encryption of messages?
  - integrity, authenticity
  - Automotive log and trace
  - ADAS information distribution

(Distributed) Log & Trace

Messaging, other triggers

- ADAS information distribution

- encryption of messages

- integrity, authenticity

(Distributed) User / Login management

GENIVI Profile Manager

A protocol for synchronizing information about “who is logged in” (at every seat).

Used for “Native Applications” concept in GENIVI architecture.

Since it’s a protocol – it’s applicable for extending to the network.

Networking challenges

Trend: Networking bandwidth is used for different purposes

- Car-provided Internet Connection
- User-provided Internet Connection

Both are used for:

- Services the OEM should pay for
- Services the User should pay for
- Critical functions, e.g. autonomous driving-related map data?
- Convenience functions
- Non time-critical functions
- Streaming media (not critical, but annoying if there are drops)

Networking challenges - needs

Routing of data to the right network

Prioritization of data for functions

Accounting of data usage (payment)

Time-sensitive Network data (AVB/TSN)

And of course: Network Security

Networking and Cloud/Internet challenges

“IoT style” protocols like MQTT are used to some effect in industry, but would be better if supported all the way.

- OEM IT departments are often in charge, and are conservative to “modern” protocols

Networking and Cloud/Internet challenges

Franca to Networking:

Common IDL (Franca) code generation exists for

- Common API C++, SOME/IP (and D-Bus)
- Good start but…
  - Need Franca version upgrade
  - Need more extension for outside car networking

…for cloud connectivity other standards are more popular. Bindings are coming, but slowly

- Franca to Web connection and WAMP → with REST bridge → started. Automatic? WAMP to MQTT bridge. Other options should be similar.

Not enough shared open-source development

Security

Trusted Execution Environment

– a separate domain?

Thoughts & Ideas?

→ Will be more discussed in GENIVI Security Team

Other topics

Input Coordination