



**GENIVI®**

# Vehicle Domain Interaction Workshop #2

Drafting APIs

(= Define top projects and API scopes)

October 12, 2017

---

## Gunnar Andersson

*Development Lead*

*GENIVI Alliance*

This work is licensed under a Creative Commons Attribution-Share Alike 4.0 (CC BY-SA 4.0)

GENIVI is a registered trademark of the GENIVI Alliance in the USA and other countries.

Copyright © GENIVI Alliance 2017.

# Agenda and Goals

- 1. Identify and prioritize projects
- 2. Pick the top 3
- 3. Define project goals and scope
- 4. Draft list of potential already existing solutions to explore
- 5. Draft APIs  
(for supporting the understanding of goal/scope)
- (Any time) : Add new topics if something was not mentioned yesterday.

# Quick Poll: Project Priority/Interest

1. Hypervisor solutions & low-level API standardization (virtio etc.) (12 votes)
2. Generic Communication Protocol Task Force (12 Votes)
  - for cross domain GNU-Linux/AUTOSAR/Android/other
  - Safe network protocol (End-2-end, etc.) (3 Votes) (combined with above topic)
3. Graphics Transfer/Sharing (11 Votes)
  - Distributed HMI Compositing (6 Votes) (combined with above topic)
4. Input Handling coordination (6 Votes)
5. System Health/Debugging/Analysis (incl. Log & Trace)(5 Votes)
6. Distributed System Lifecycle / Node State (4 Votes)
7. Network Traffic routing & accounting (priority/bandwidth/payment...) (3 Votes)
8. (Distributed) Audio management (2 Votes)
9. Distributed User / Login / Profile management (1 Vote)
10. + **Input Handling coordination** (Was not included in Poll)

# Graphics sharing & Distributed HMI Compositing

- **Description:**

- Toplevel compositing across domains for the same physical display with or without mixed safety levels  
Graphics transfer encoding/technology

- **Goal:**

- 1 to 2 applicable, open standard/specifications, implementable on Linux *and other OS*
- Demonstration implementation, education material
- **Existing technologies to explore**
  - 1. Waltham
  - 2. Graphics framework “R”, to be open source?
- Is there another level of protocol needed, not like Wayland?

# Graphics sharing & Distributed HMI Compositing

- Main interested companies
- Renesas, Harman, BMW, ADIT, Luxoft
  
- **“Prior Art” / Existing Tech**
- VNC/RDP
- Qt / HTML5 specific solutions? Android?
-

# Graphics sharing & Distributed HMI Compositing

- Is this one project or two? – one
- **Sub-topics / investigations:**
- Waltham
- Waltham on non-Linux – especially Android (missing these features today)
- Graphics framework “R”
- Multidisplay vs multi system – diff solutions?
- Gather use cases
- Adjacent proj: app coord, input coord,

# Top choice #2 Hypervisor standardization

- Goals
  - - HV investigating available choice (maybe separate activity)
    - extending/developing HV APIs eg virtio
- Scope
  - - Multiple HV, safety-critical and more generic

## API Functions

- Cross-VM low-level communication standard
- Existing technologies to explore
  - - virtio – where is that coordinated? LF?

# Top choice #3 - Generic comm protocol

- Goals
  - - Survey choices among MQTT style and other messaging level protocols
    - Do the same for SOME/IP
  - - Reach consensus on preferred options:
  - - Among 10-15 choices, find 3-5 recommended (and for which scenarios they are recommended)
- Scope
  - Messaging and generic level protocols
  - approx. OSI level 4-5?
- API Functions

Existing technologies to explore...



# Input Coordination

- Problem statement
  - - Complexity in encoding rules for input to effect
  - - Bounded latency guarantees
  - - Modes/modality, dynamic rules
- Goals
  - - Distributing the info using a shared common protocol
  - - Design guideline + method for rule/logic definition
- Scope
  - - All input devices:
    - Buttons, touch, gesture, speech control, etc.
    - Multi-modal scenarios
- API Functions
  - Existing technologies to explore
  - VSM

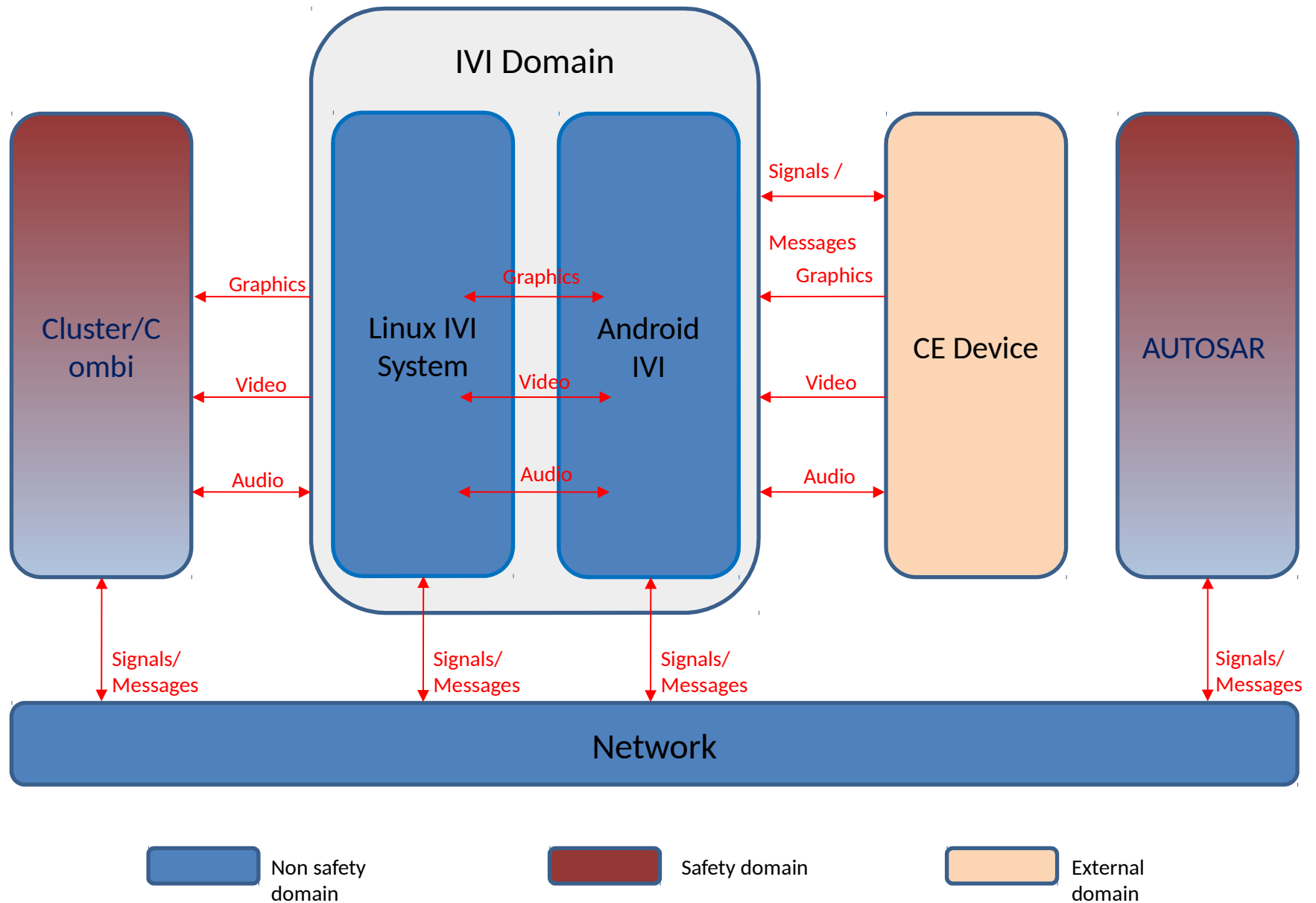
# Distributed Audio coordination

- Question: How does the Audio subsystem deal with multi-user scenario particularly in ride-sharing setting.
- We concluded this was a question from the day before – need to understand which aspect of “dealing with” the person was asking about.

# **Backup/Support slides**

(From introduction workshop #1)

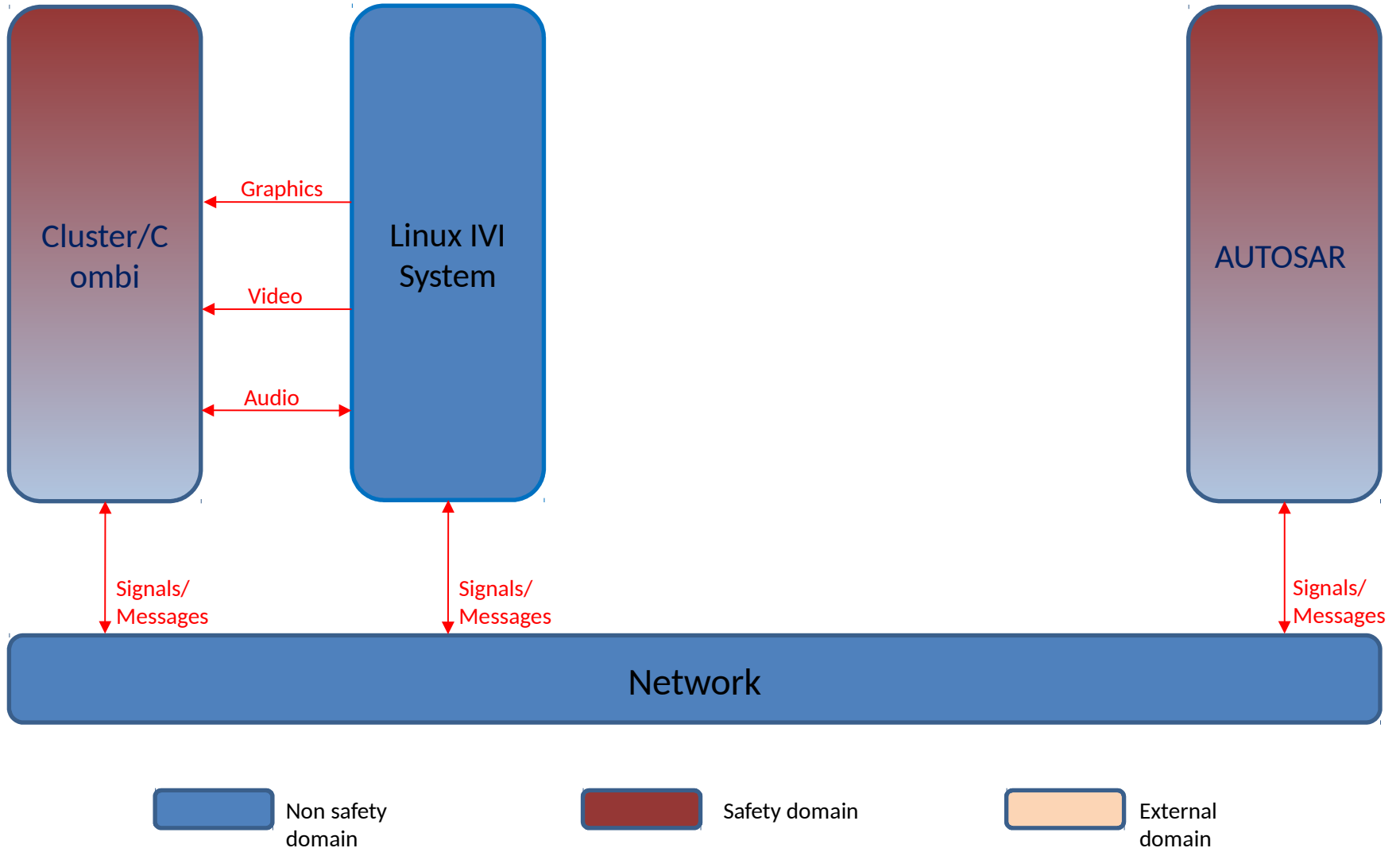
# Example Architecture – #1 Combined IVI system



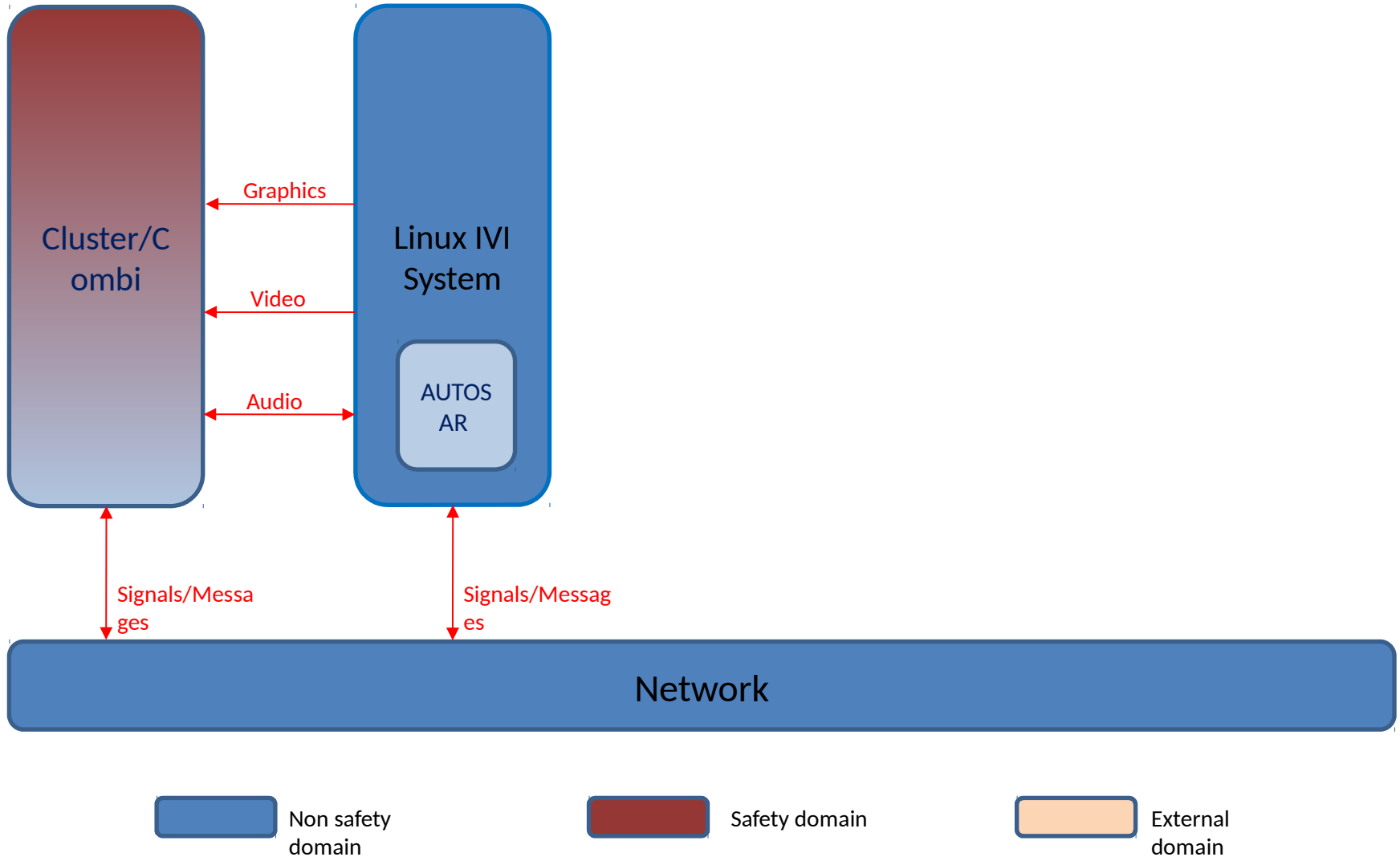
# Definitions

- Graphics:
  - A) Pictures, transfer already rendered bitmaps
  - B) Drawing commands, OpenGL, Rendering needed receiving side
- 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

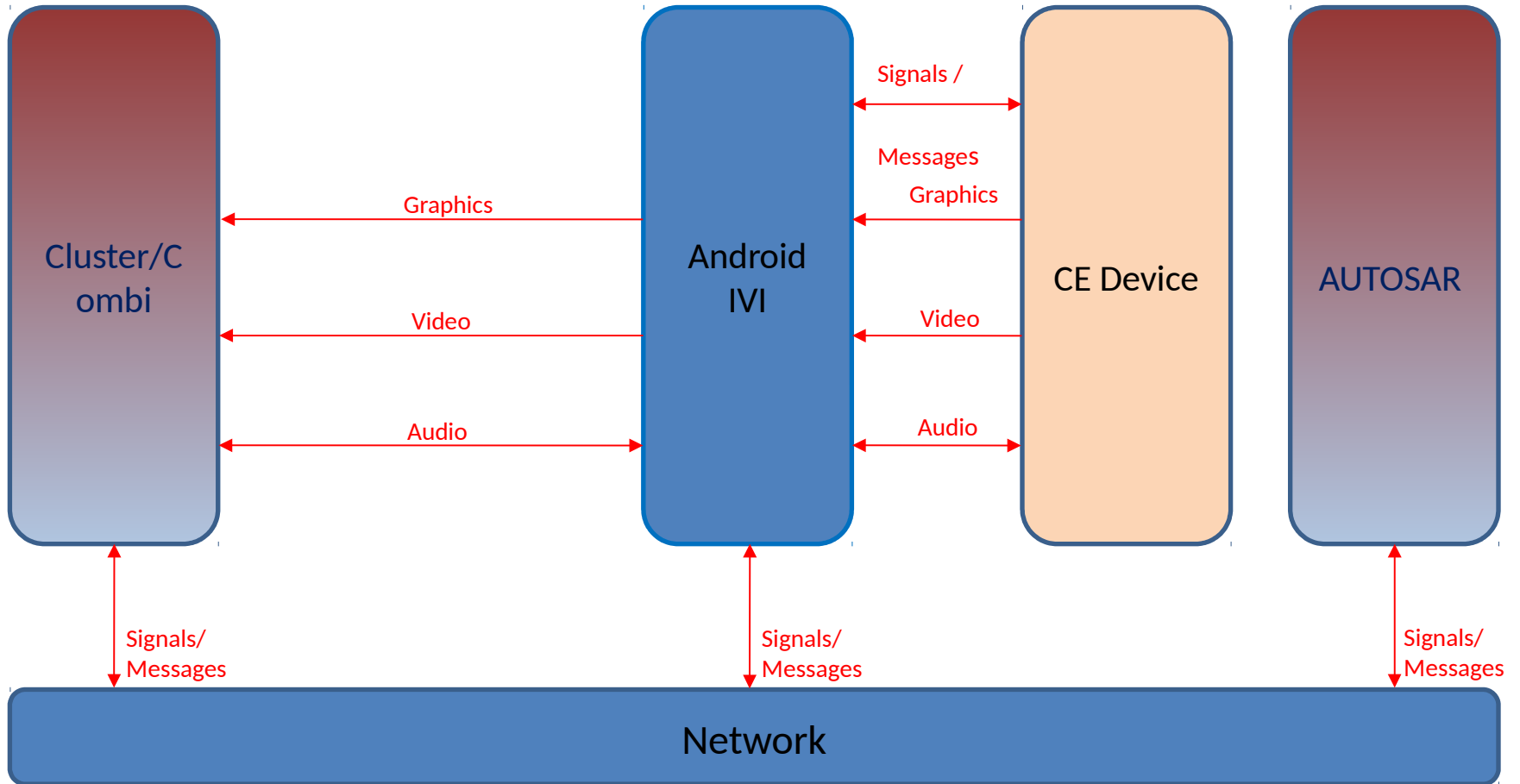
# Example Architecture – #2 Linux-only IVI system



# Example Architecture – #3 Embedded AUTOSAR



# Example Architecture – #4 Android only IVI system



 Non safety domain

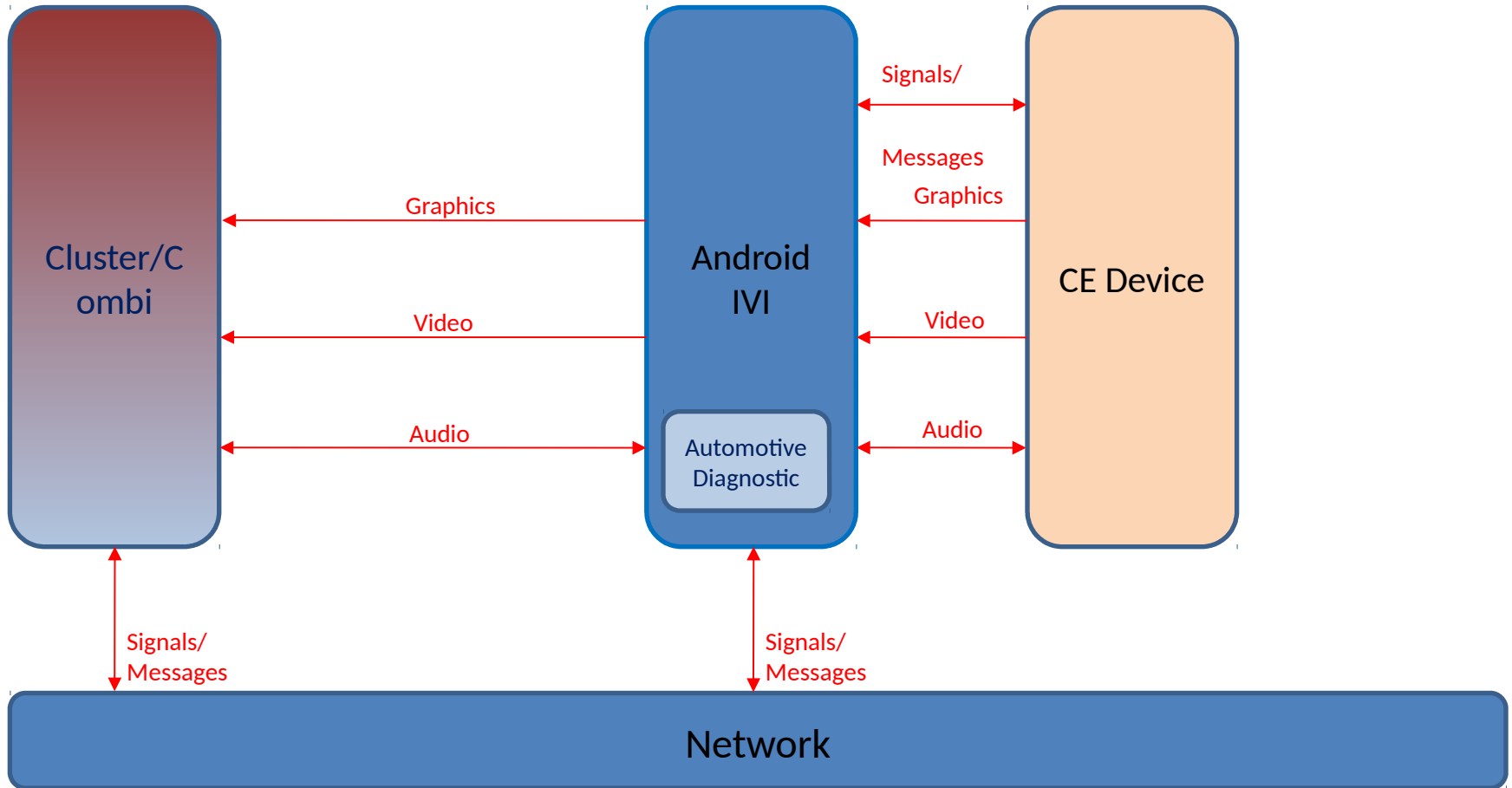
 Safety domain


 External domain



# Example Architecture – #5

## Android AutomotiveFeatures



 Non safety domain

 Safety domain

 External domain

## **Audio Management:**

- Separate Audio Management needed in Safe domain?  
vs. common 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

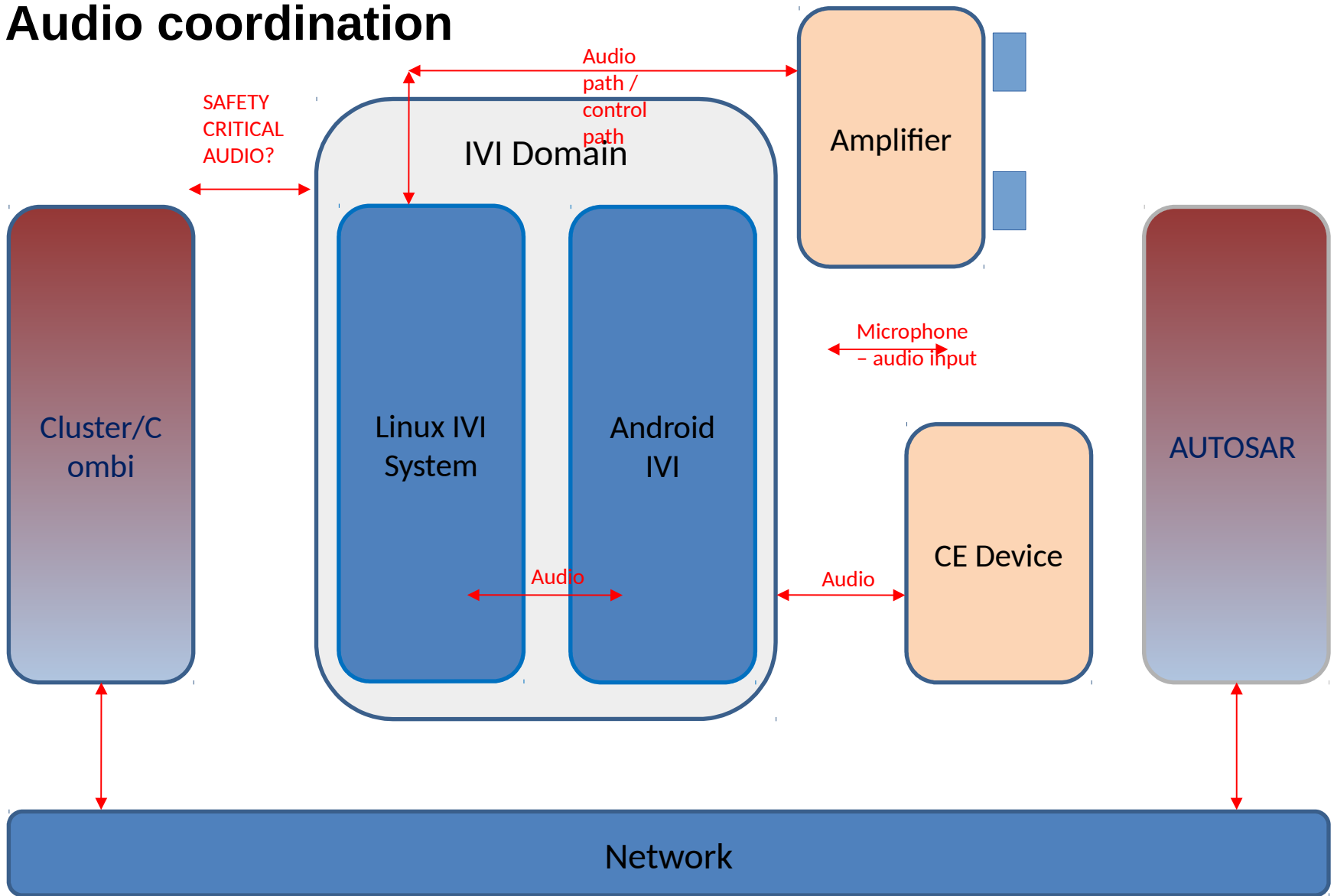
## **Cross domain data exchange**


**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 media player (e.g. mp3)
- Handsfree speaking to/from CE device
- Text to Speech
- Navigation speech output from CE Device navigation (e.g. Google Maps)
- CE Device voice assistant (Siri, Bixbi, Google, Alexa, ...)
- Audio Notifications
- Audio mixing (standard audio management)
  
- Audio Safety Domain relevant
- - warning sound (chimes)

# Audio coordination



 Non safety domain

 Safety domain

 External domain

# 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.

**Remember:** to carry **IDENTITY** when doing connectivity protocols, e.g. to/from cloud etc.

# Analysis

- \* Understanding hypervisor behavior / total behavior with multiple VMs running.
- \* Tooling to get the full system view
- \* TCF – needs dev effort to bring up to quality

Runs on autosar and others

# 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

