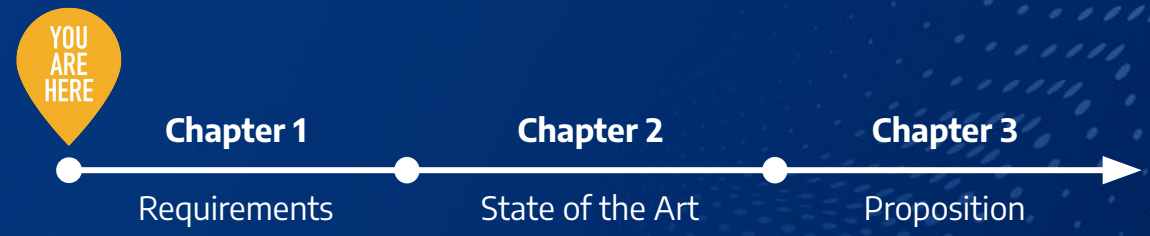
Ethernet-as-a-Service for Software Defined Vehicles:

Design objectives and orientations for an Ethernet-based network stack



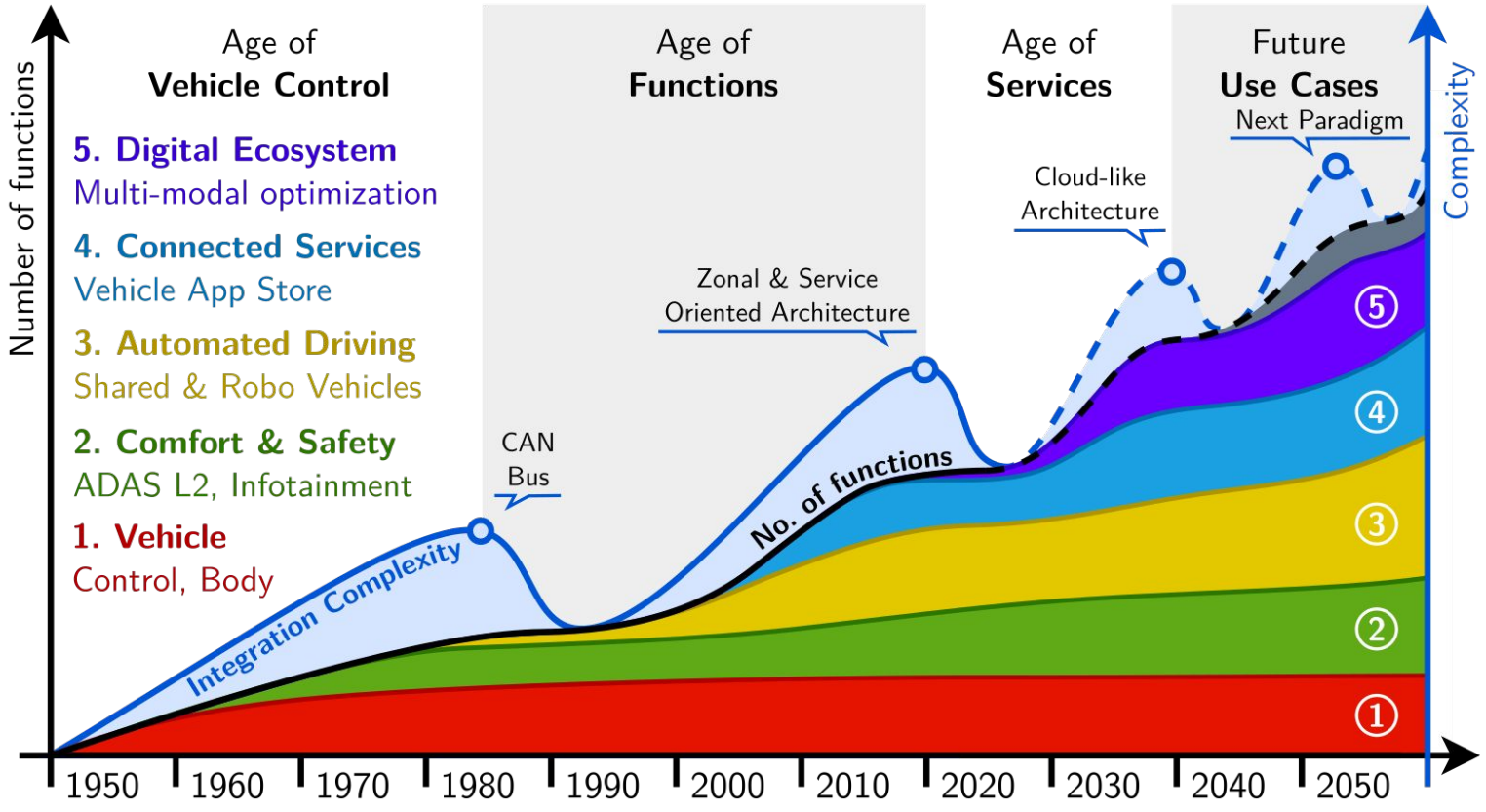
Pierre LACLAU – PhD Student
with Xiaoting LI and Trista LIN
2022 IEEE Standards Association

Agenda

STELLANTIS heudiasyc utc

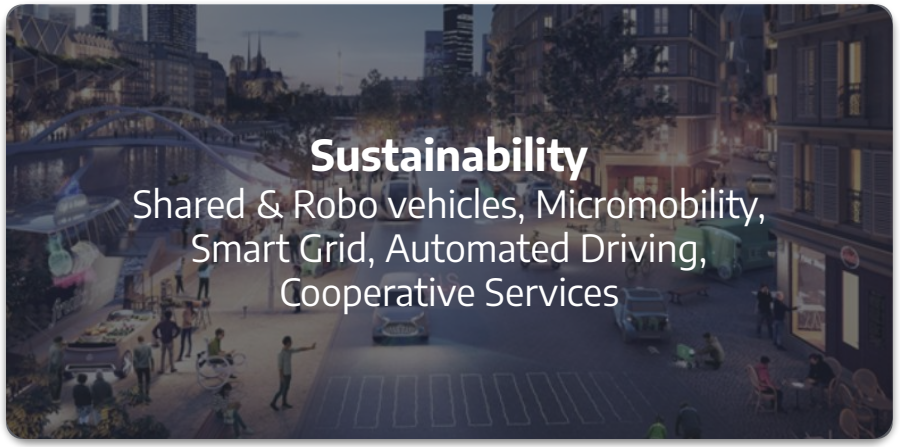
Automotive Evolution – Use Cases

A profound digital transformation



Inspired from C. Buckl et al., "The software car: Building ICT architectures for future electric vehicles" at *IEEE International Electric Vehicle Conference*, 2012

Two major digital transformations



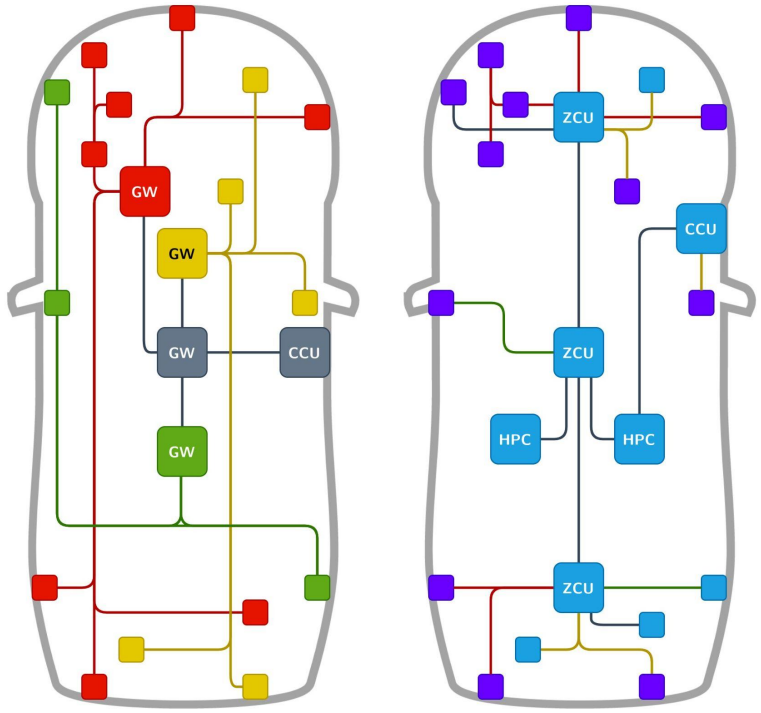
Images: Ralf Marquard from *LHP Europe*, Eliane Fiolet from *Ubergizmo*

Architectures Evolution

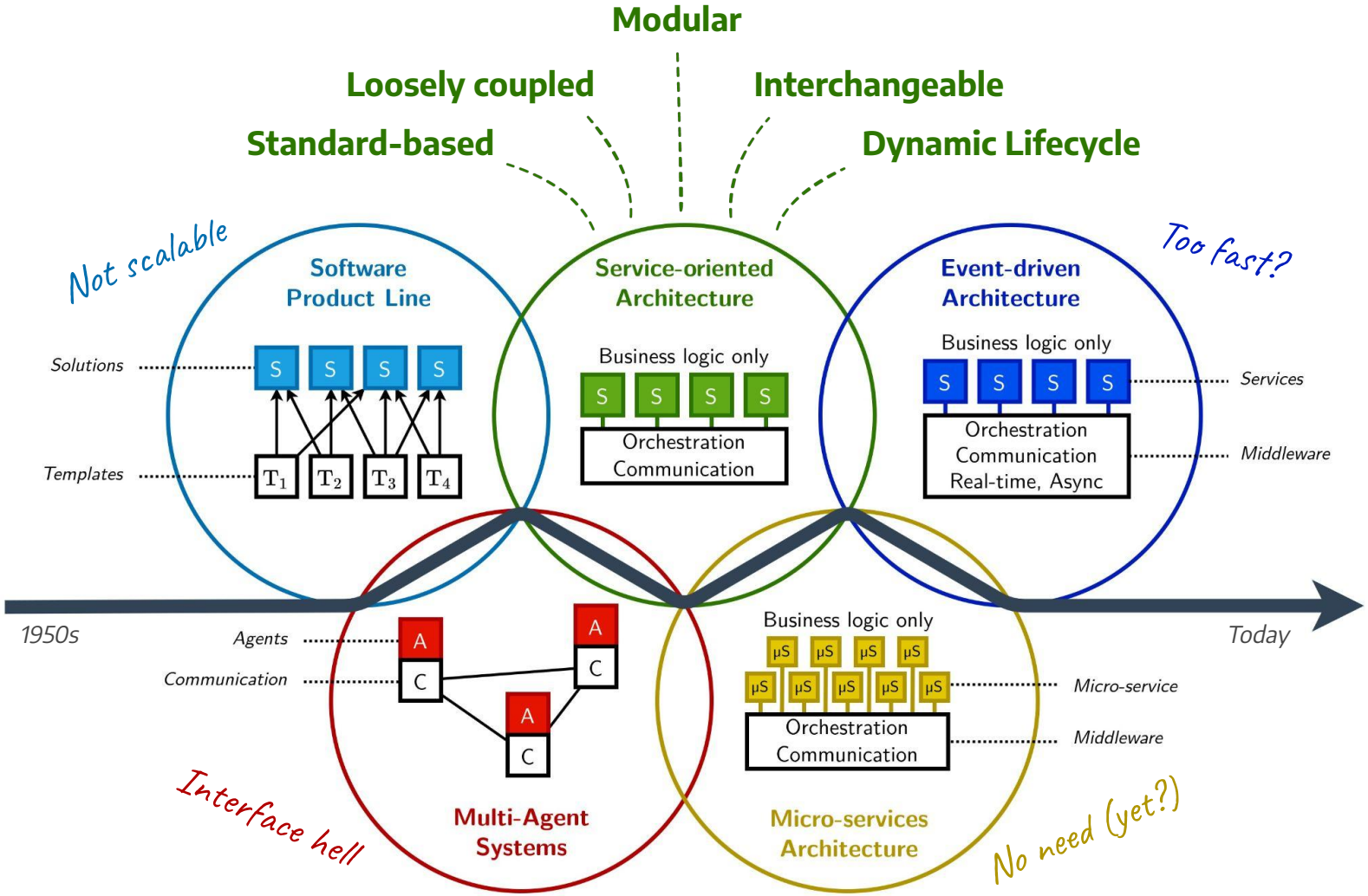
Software is eating the car

Domain-oriented
2000-2020

Zonal-oriented
2020 and beyond



GW - Gateway **ZCU** - Zonal Control Unit
HPC - High Performance Computing
CCU - Connectivity Control Unit
With or Without Virtualization
CAN Ethernet



↦ **Service-oriented:** The right compromise for automotive

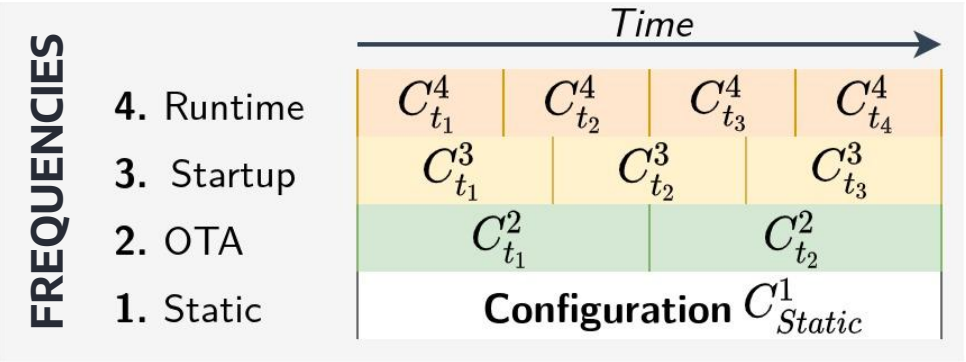
Ethernet Requirements

Dynamic future use cases

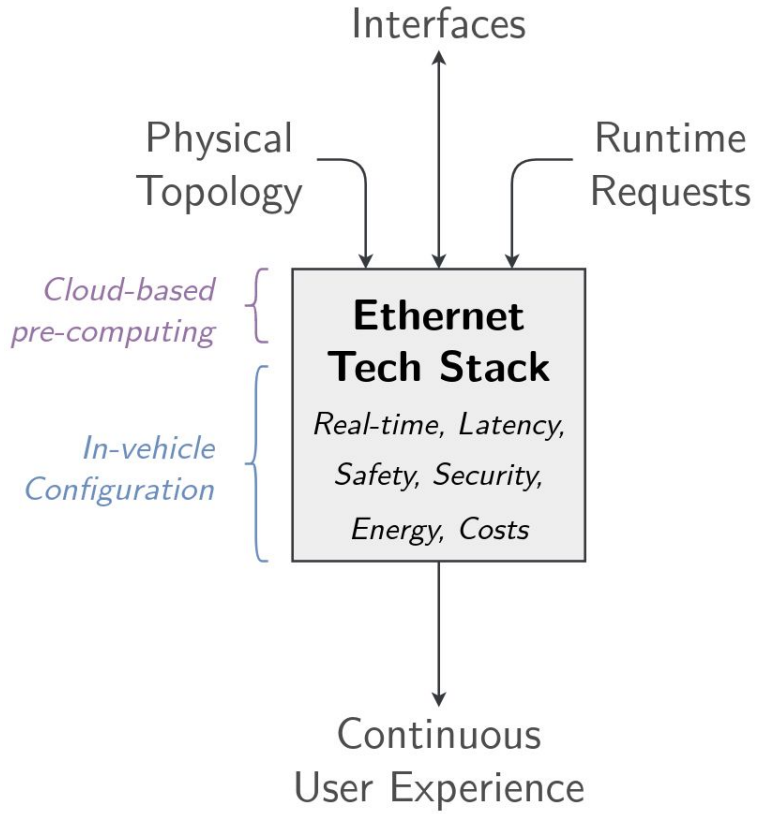
DYNAMIC

STATIC

- **Cooperative Services** *(Awareness, Parking, Intersections...)*
- **User Applications** *(Subscriptions, Infotainment, Games, Tourism, Maps...)*
- **Automotive Services** *(ADAS L1-L3, Individual RT Insurance, Voice assistant...)*
- **Platform Services** *(OTA Update Manager, Control Services...)*
- **Vehicle Features** *(Regulations, Logging, Region...)*
- **Vehicle Platform** *(E/E control, OEM Theme...)*
- **Hardware** *(E/E Phy Topology, Motor...)*



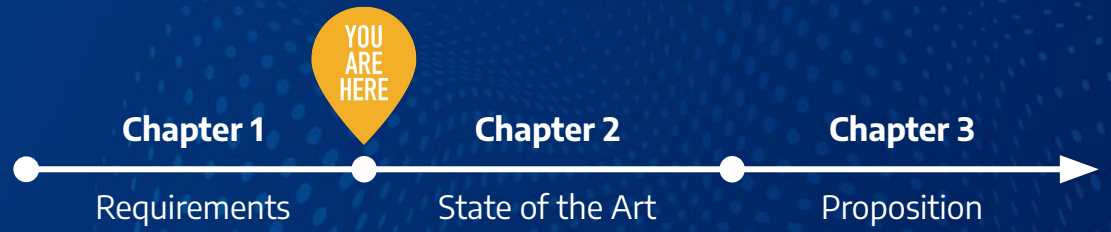
Need for dynamic reconfigurations
+
Stack of configurations
for a semi-dynamic and safe network



↦ **Question:** What could a complete Ethernet software stack look like?

Chapter 2: State of the Art

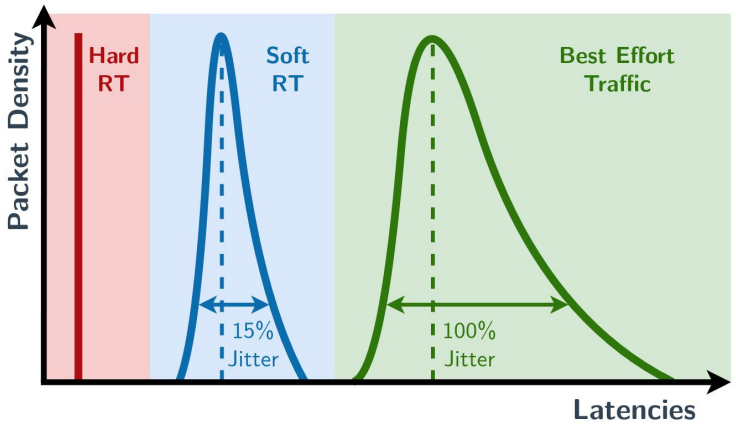
Next up: Overview of the current technologies (existing or under development, research, standards)



Ethernet Timing

One cable, Mixed-criticality QoS Transport

Mixed-QoS Networking



High diversity of available technologies

Choice of solutions	Interchangeability
Adaptability	Interoperability
Flexibility	Configuration
✓	✗

Hard Real Time

TSN - Time Sensitive Networking

1. Synchronization
- **802.1AS** Time Synchronization

2. Latency
- **802.1Qav** Credit Based Shaping
- **802.1Qbv** Time Aware Shaper
- **802.1Qcr** Asynchronous Traffic Shaping
- **802.1Qbr/bu** Frame preemption

3. Reliability & Safety
- **802.1Qci** Per-Stream Filtering, Policing
- **802.1CB** Frame replication, elimination

4. Management & API
- **YANG** Configuration via NETCONF
- **Dedicated APIs** for Qbv, Qav, ...

Soft Real Time

Strict priorities, higher bandwidth

5. Strict priority
- **Statistical analysis** Like in CAN buses
- **Traditional** priority-based shaping

6. Rate Limiting
- **802.1Qav** Credit Based Shaping

Best Effort

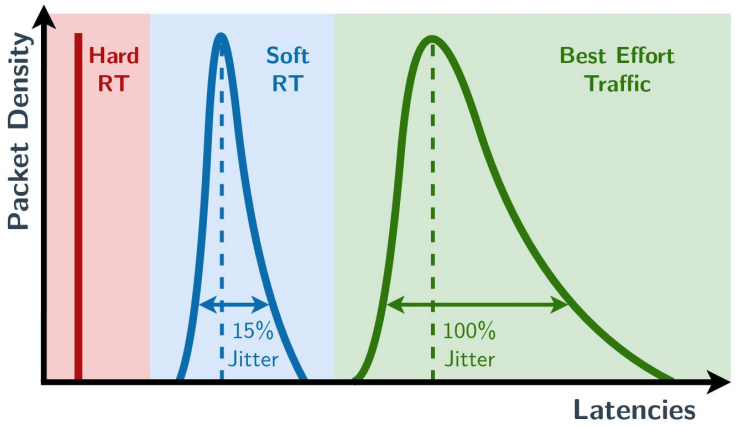
Stream reservation, higher bandwidth

7. Reservation
- **Bandwidth reservation** at design stage
- **Higher bandwidth** despite costs
- **Scheduling** best-effort aware

↦ **Question:** How can we make modules interact?

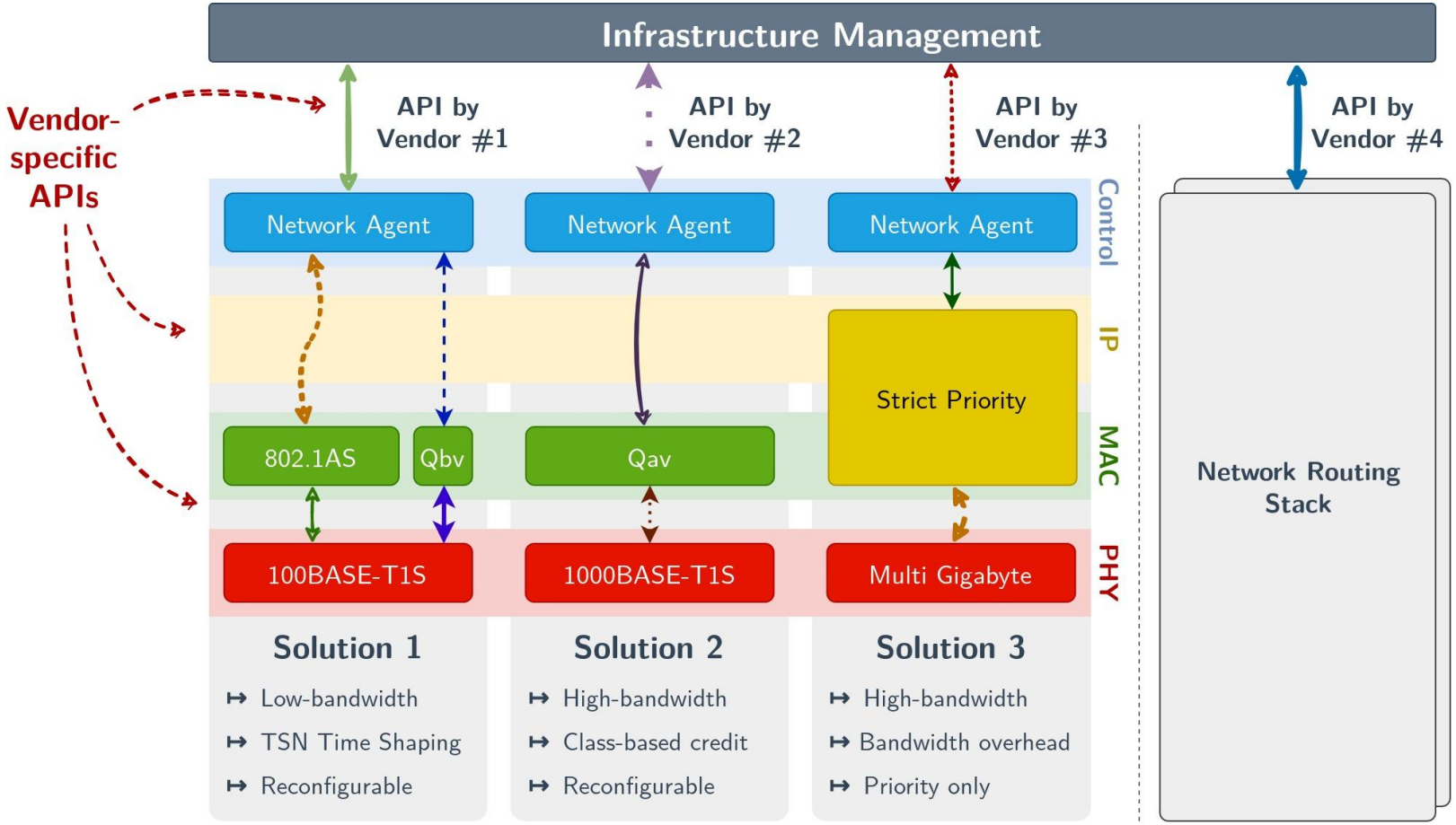
Ethernet Timing
Multiple possible technology combinations

Mixed-QoS Networking



High diversity of available technologies

Choice of solutions	Interchangeability
Adaptability	Interoperability
Flexibility	Configuration
✓	✗



Ethernet Timing
Mixed-criticality QoS requirements

1.

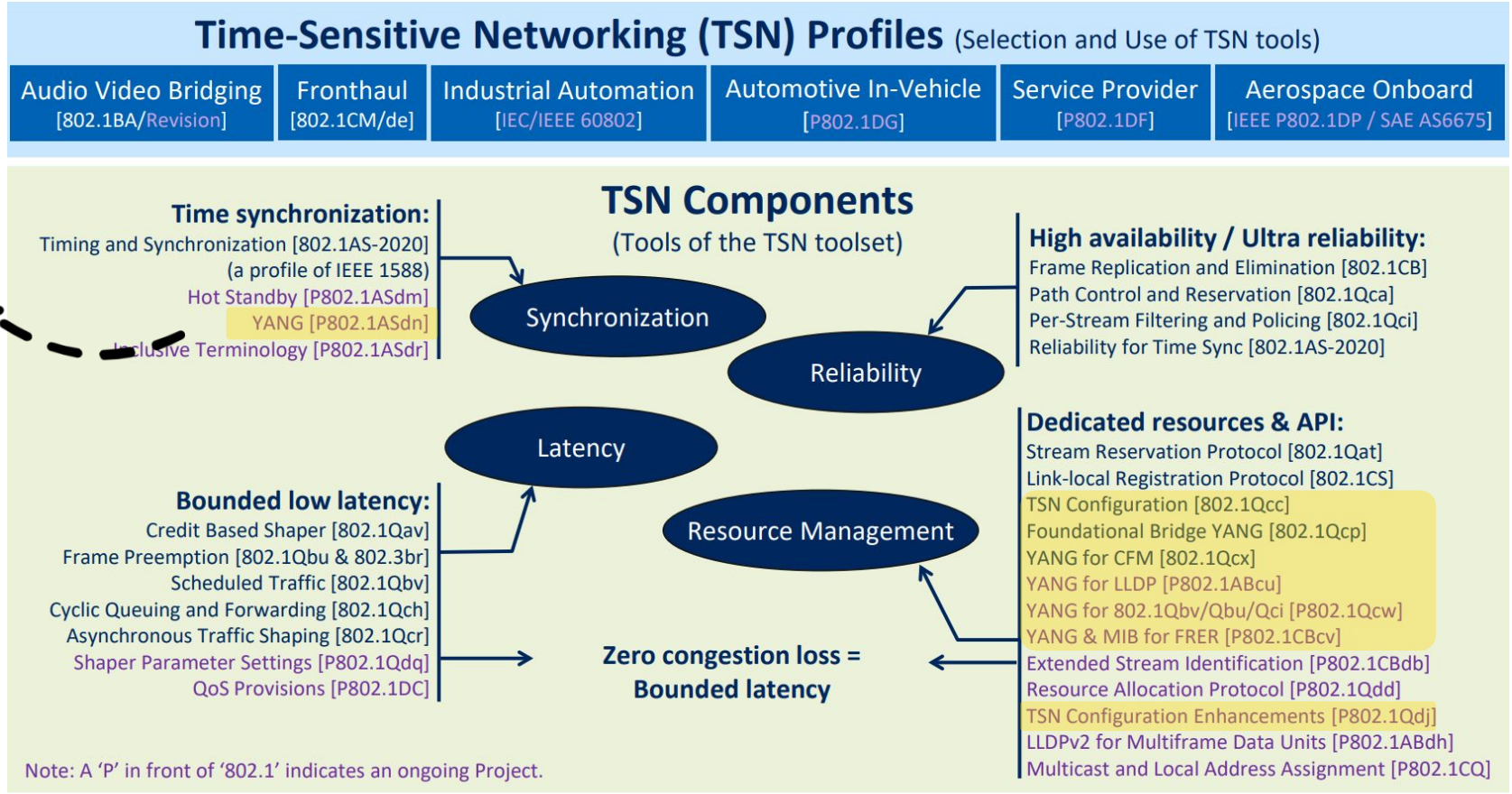
Ongoing Interface Projects

Great start !

2.

What about other modules?

How to build a coherent global configuration?



More on TSN standards and ongoing projects at: https://www.ieee802.org/1/tsn

11/16/2021

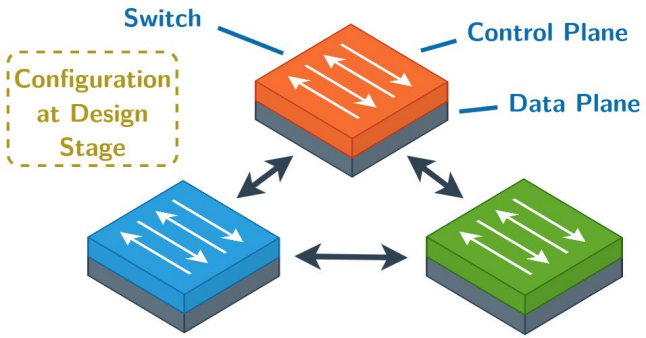
3.

Keyword: Interoperability

Necessary for collaboration on the next SDV platforms

Ethernet Routing
Dynamically reconfigurable networking

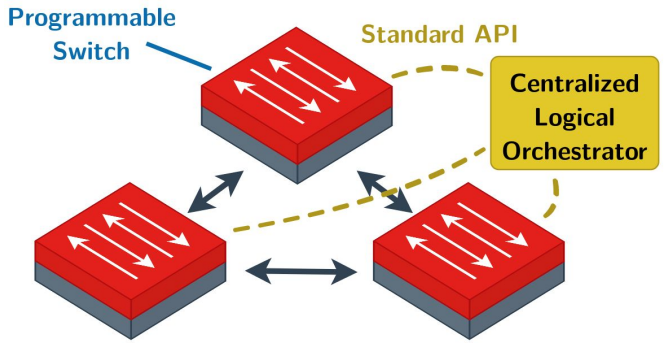
Traditional Networking
Local Switch Management



1 Switch, 1 interface
Local management
Custom reconfiguration solutions

Works, but hard to reconfigure

Software Defined Networking
Centralized Orchestration



Common interfaces
Centralized monitoring
Global & Dynamic (re)configuration

(Still under research)

Cold Topics

VLANs & VxLANs
Cybersecurity
Tunneling
CAN/LIN Gateway
Cloud Bridge
External Gateway
Firewall
Energy management

↦ **Problem:** TSN must be configured based on external parameters...

↦ **Question:** How to create coherent configurations through the stack?

Ethernet Control Plane
Dynamic Control Services

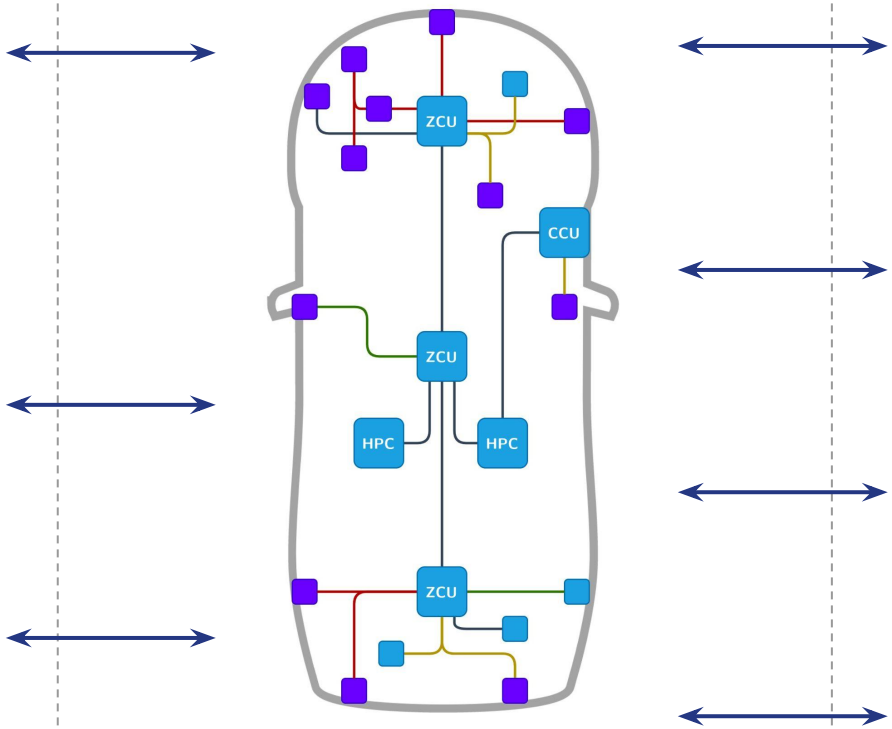
Ethernet Control Services

1. Network Scheduling
- Global Configurator
- Time sync Manager
- Firewall Scheduler
- Offline Scheduler
- SDN Scheduler
- TSN Scheduler

2. Network Monitoring
- Intrusion Detection
- Self-healing
- Diagnostics

3. Gateways
- External Configurator
- Hardware Discovery
- Legacy Configurator
- Cloud Bridge

Interaction Map



↦ **Problem:** How to make all services interact together?

SOA Control Services

4. Vehicle Management
- Vehicle State Manager
- Telematics & Logging
- Global Configuration

5. Service Management
- SOTA/FOTA Update Manager
- Lifecycle Manager
- Service Monitor

6. Communication Management
- Dynamic Flow Mapping
- Service Discovery
- Load balancing

7. Platform Management
- Time Synchronization Manager
- Global Diagnostics
- Data Collector

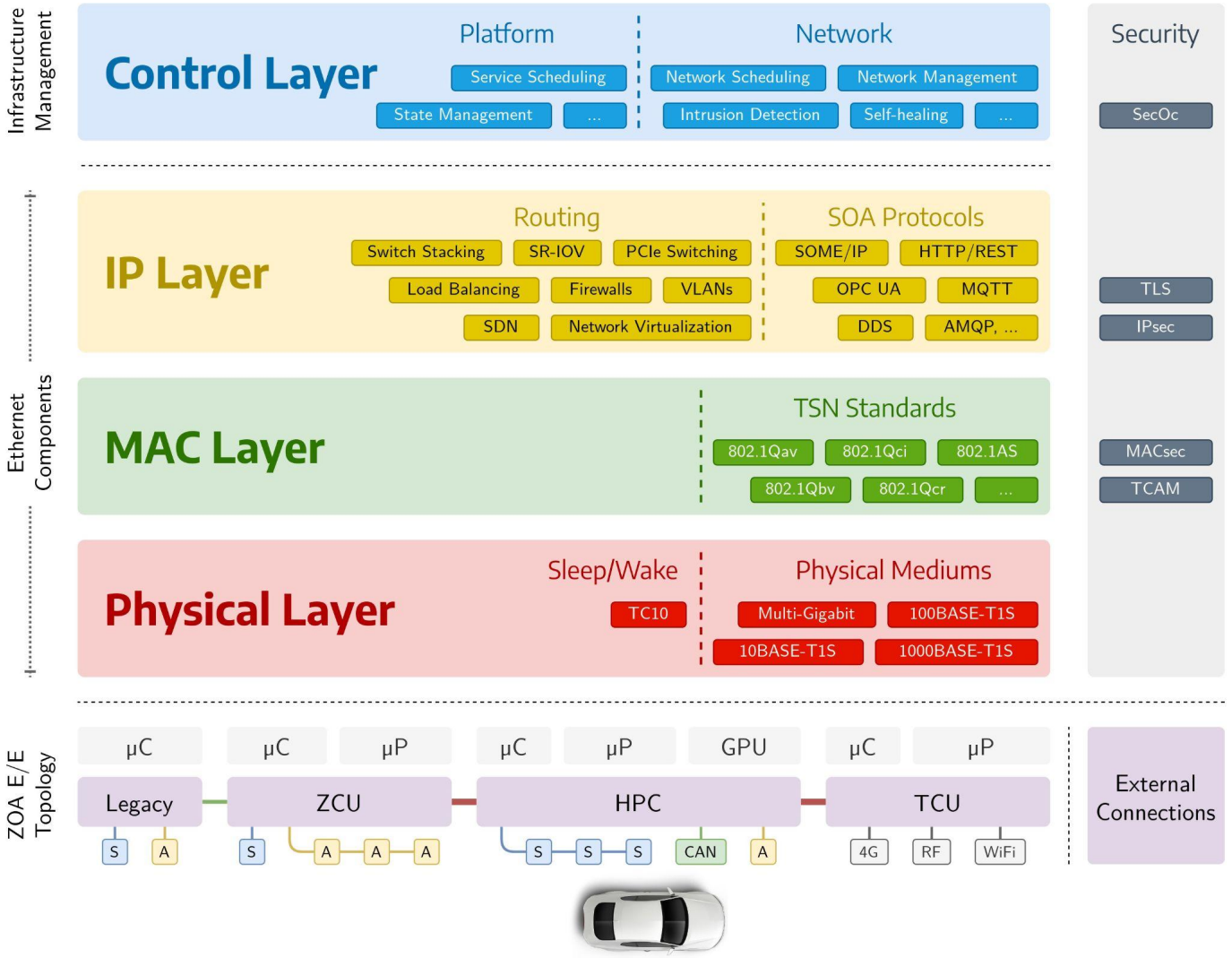
Ethernet Technology Map
High diversity, high complexity

Conclusion so far...

1. High diversity is good for choice
2. Different combinations will exist
3. 1 module, many implementations

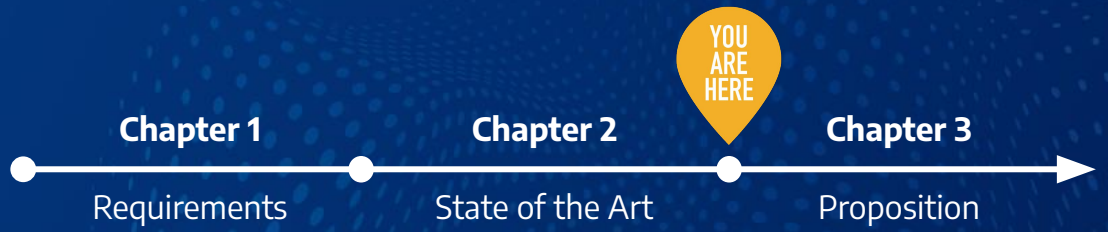
New objective

1. **Objective:** Interchangeable solutions
2. **Method:** Interfaces + States standardization
3. **Solution:** Take inspiration from Cloud Computing



Chapter 3: Integration

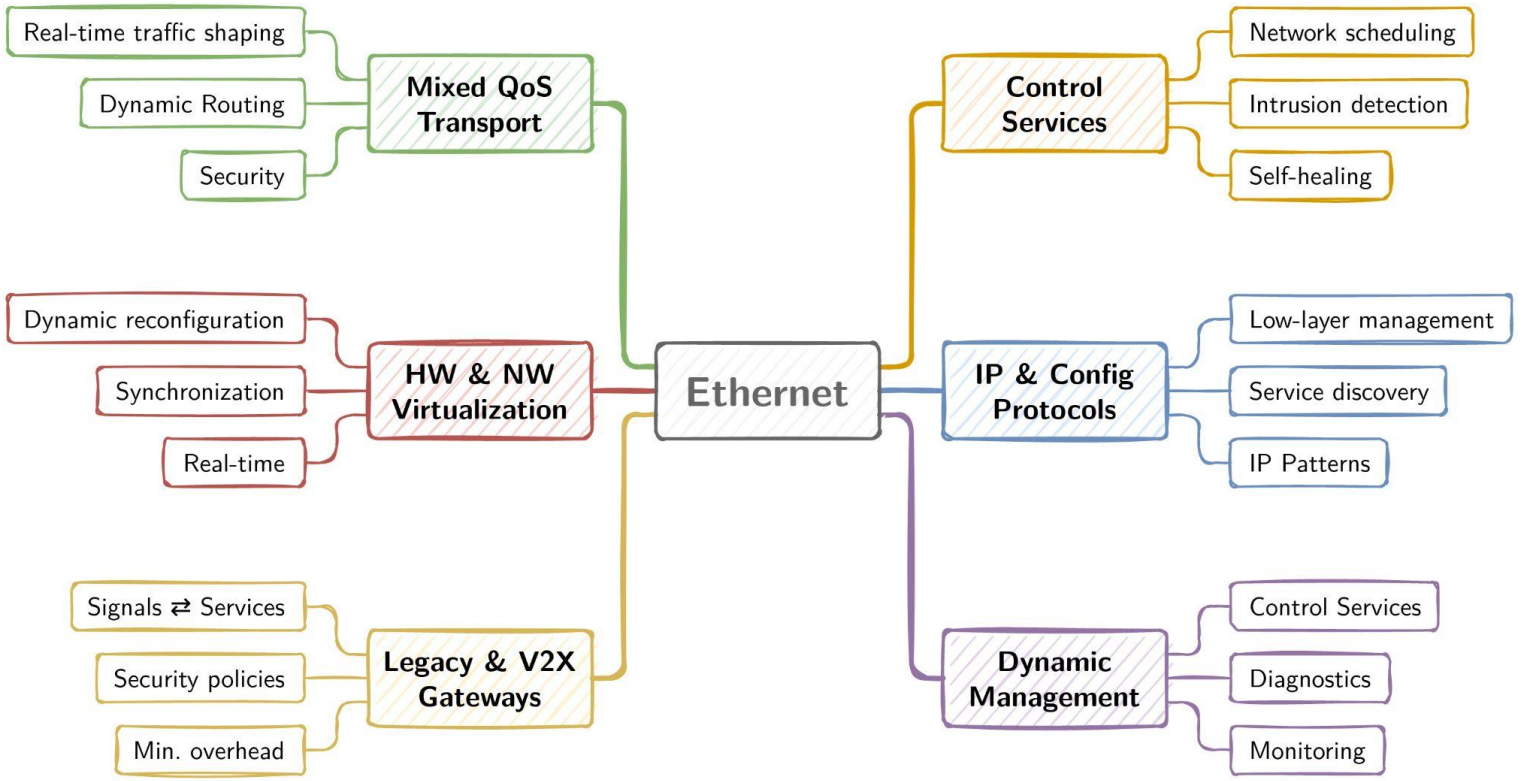
Next up: Inspirations from other industries that could help us define a fully integrated Ethernet solution



Ethernet Requirements

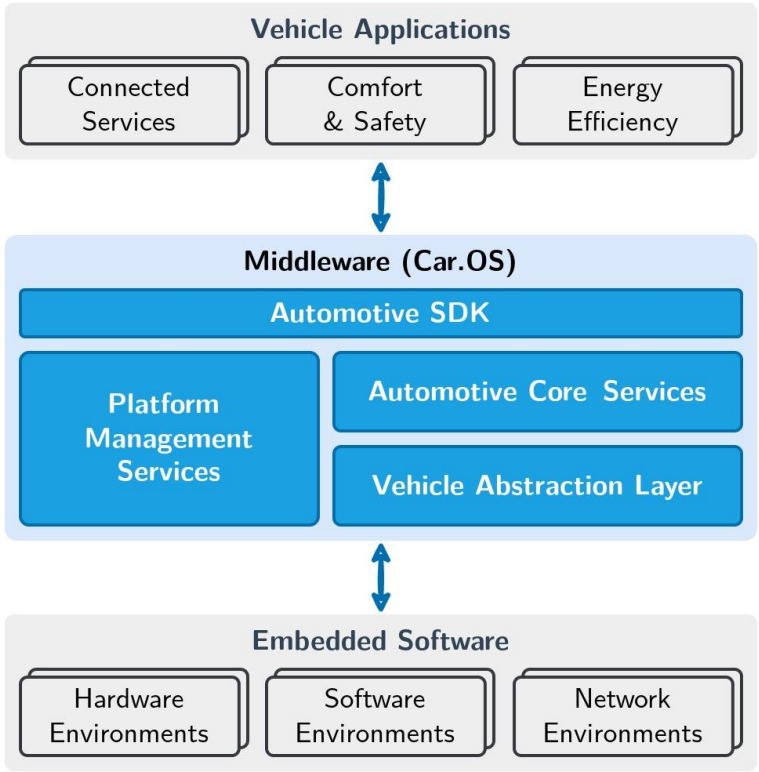
For a continuous user experience

How to integrate Ethernet?



SDV Ecosystem

Hardware is abstracted from Services



Architecture Design Requirements

Time-to-market *Easy to develop* *Maintainability* *Security* *Standardized* *Legacy compatible*

Lightweight *Interface to ZOA* *Interface to SOA* *Dynamic reconfigurations* *Traceability*

Cloud Computing Architectures
Everything-as-a-Service (XaaS)

IT only:
- ❌ Storage Orchestration
- ❌ Backups & Rollbacks
- ❌ Batch Execution

Automotive only:
- 🚗 Heterogeneous environments
- 🚗 Low & embedded resources
- 🚗 Real-time functions
- 🚗 Safety

Common features:
- ✅ Service Scheduling
- ✅ Service Discovery
- ✅ Network config.
- ✅ Self-Healing
- ✅ Redundancy

Software Defined Vehicle
"Data Center on Wheels"

1. Flexible & Instantaneous Updates
2. Easy monitoring & diagnostics
3. Virtualized networking
4. Plug-and-play solutions
5. Dynamic service scheduling
6. **Standardized Interfaces**

kubernetes openstack.

Image from Linode

SDVs are (kind of) like the Cloud

Similar problem, different constraints

Why not do the same?

Anything-as-a-Service

pizza!

On Premise	Infrastructure as a Service	Platform as a Service	Software as a Service
			Restaurant La Pizza
Made at Home	*Take and Bake*	*Delivery*	*Restaurant*
Dining Table	Dining Table	Dining Table	Dining Table
Soda	Soda	Soda	Soda
Oven	Oven	Oven	Oven
Electricity	Electricity	Electricity	Electricity
Ingredients	Ingredients	Ingredients	Ingredients

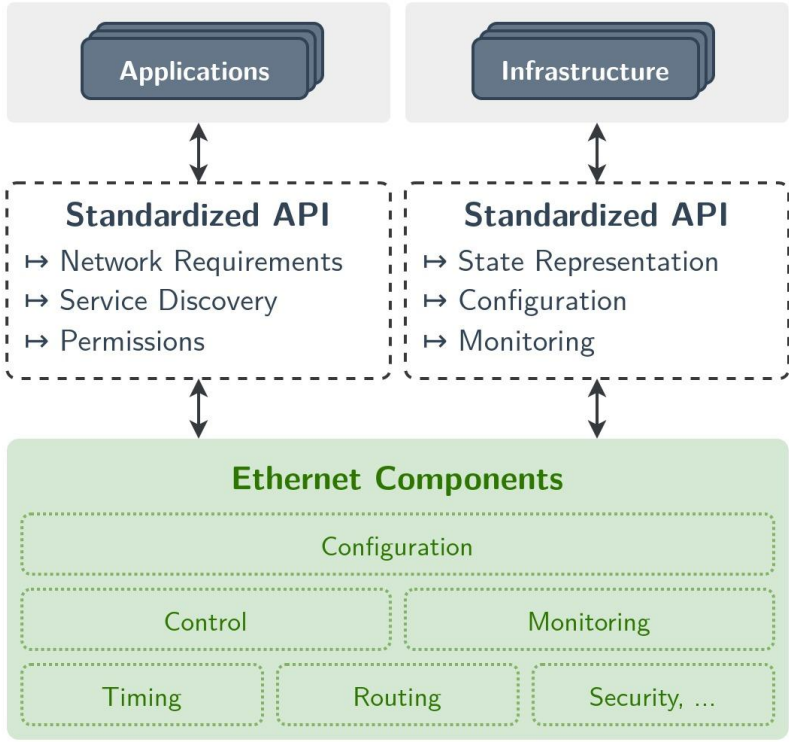
You Manage | They Manage

Ethernet-as-a-Service

Ethernet as a Service

Managed Ethernet

- SOA
- Infrastructure
- Network
- Ethernet
- E/E Topology



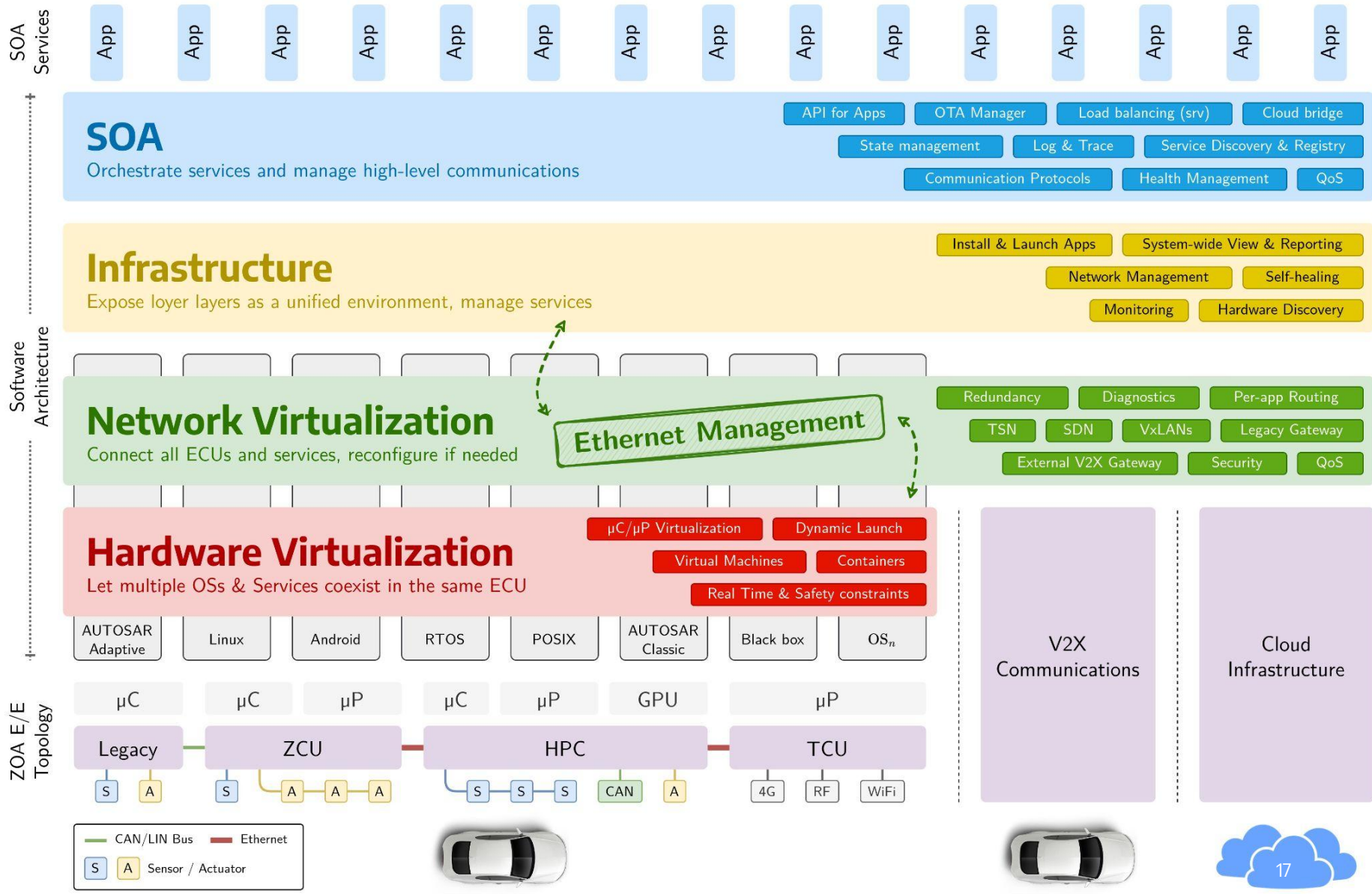
Software Defined Vehicle Stack

Dynamic service management

Current challenges:

1. Virtual environments
2. QoS Management
3. Dynamic mechanisms

↦ **Question:** How can we design and implement a self-contained Ethernet?

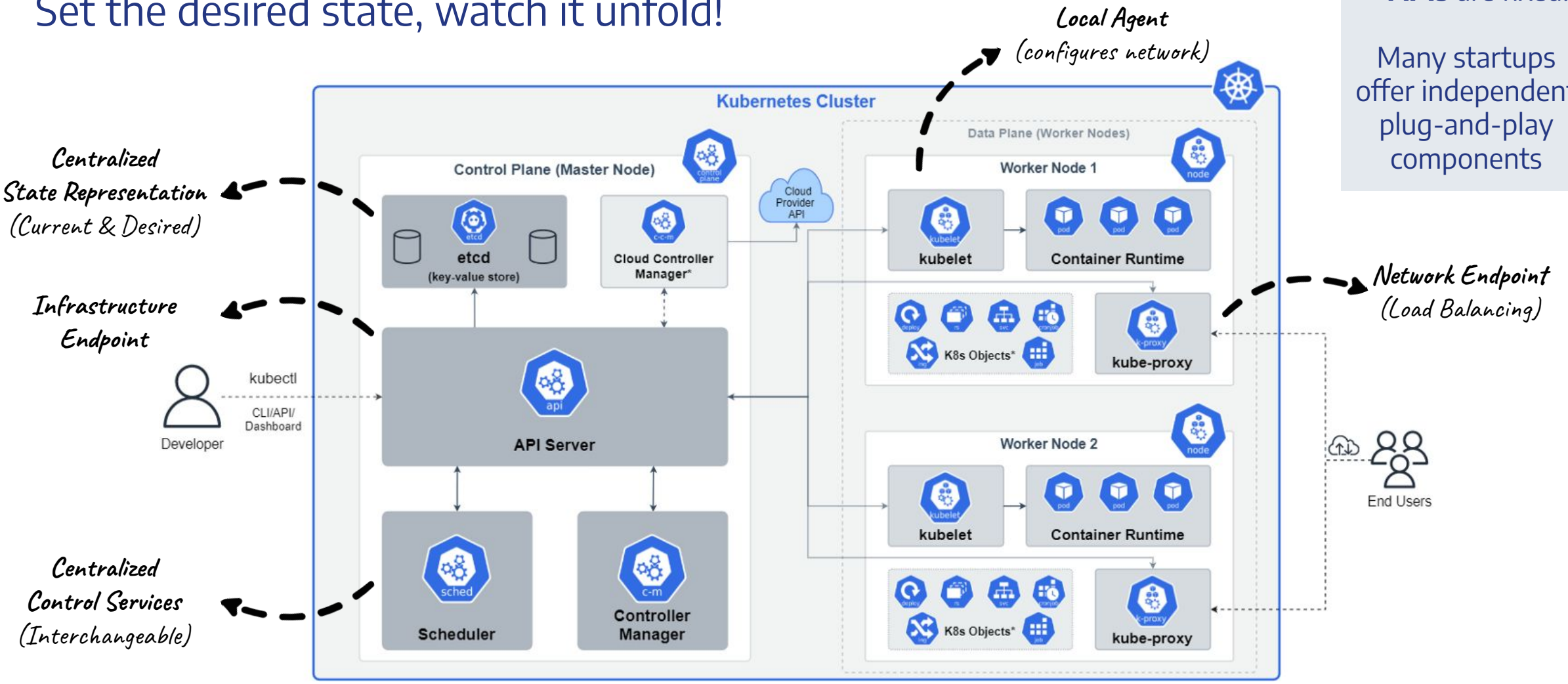


Cloud Orchestration Architecture

Set the desired state, watch it unfold!

↦ **APIs** are fixed!

Many startups offer independent plug-and-play components



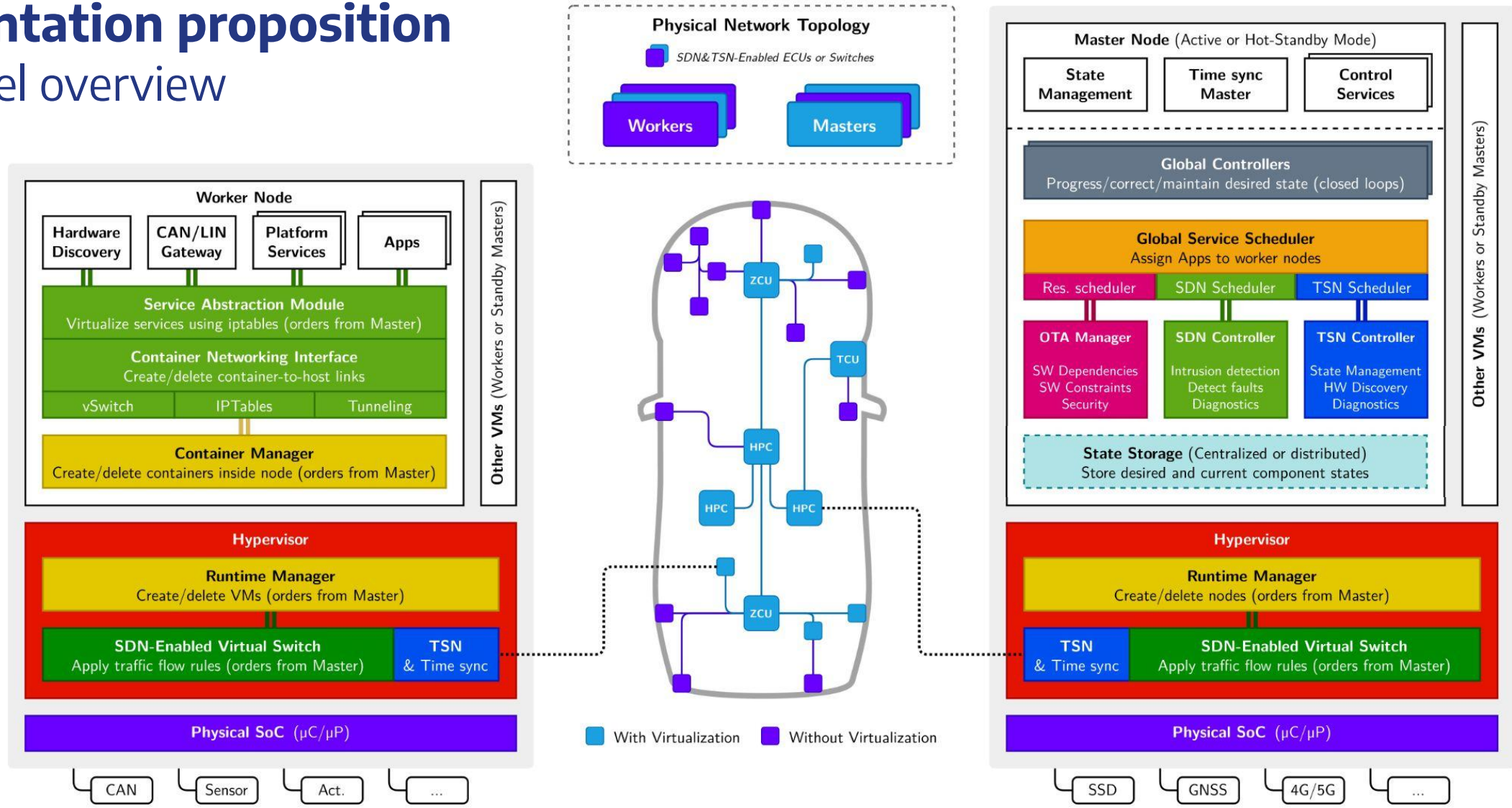
Ashish Patel, "Kubernetes - Architecture Overview", *Medium*, 2021

Implementation proposition

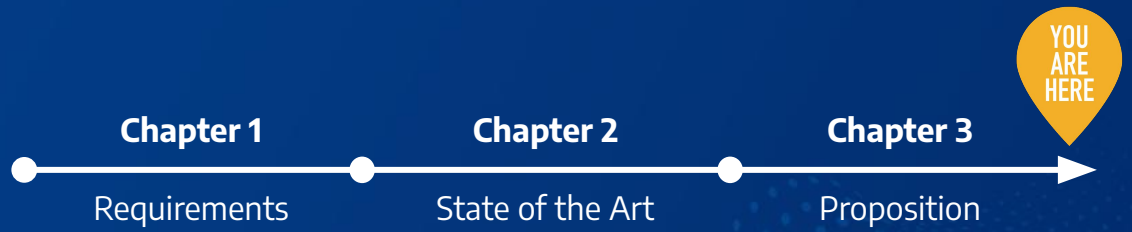
A high-level overview

Full Network Stack
Virtual + Ethernet

↦ **Standards** will be needed across suppliers!



Discussion


YOU ARE HERE
Chapter 1
Requirements
Chapter 2
State of the Art
Chapter 3
Proposition

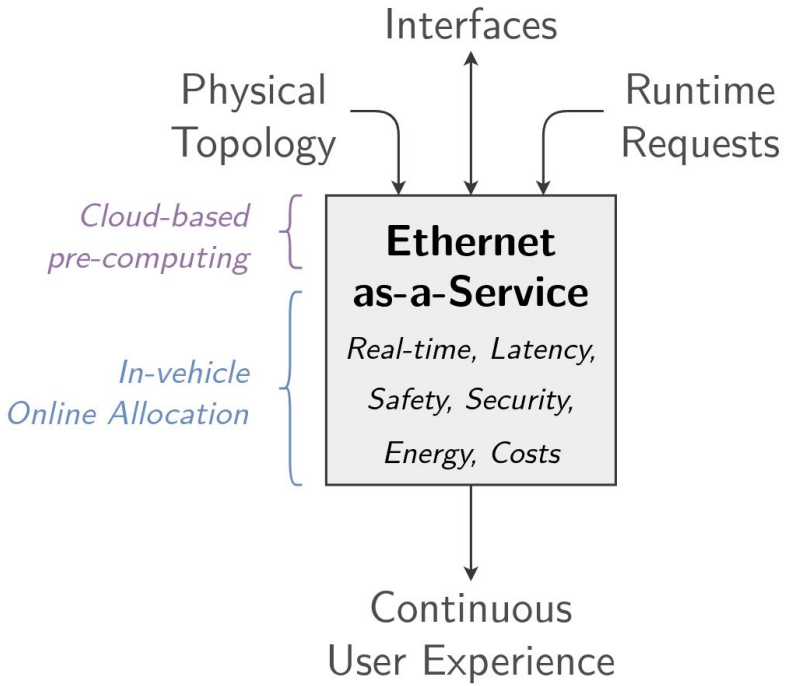
Discussion
Our take on the next steps

Summary

1. **Interchangeability** is what the industry needs
2. **A common language** can be made from standardized APIs
3. **Ethernet-as-a-Service** is a promising way to organize our standards

Propositions

1. **Discuss** how to define an application's requirements
2. **Standardize** the common vehicle state representation first
3. **Adopt XaaS** from Cloud Computing for a loosely coupled architecture



Takeaway

SDV development will need attention on
"Common Data Representation for Ethernet Requirements"

STELLANTIS

Designing a safe, real-time, secure, embedded, and cost-effective Data Center that can be used like a Smartphone *(oh, and it can drive)*
aka. Software Defined Vehicles.

More questions or discussions?
Let's chat about details offstage or by email!

Thank you for your attention!

Pierre LACLAU, Xiaoting LI and Trista LIN
IEEE SA 2022 – Automotive Technology Day
pierre.laclau@stellantis.com

Watch the talk replay on IEEE SA's YouTube channel

ALLIANCE SORBONNE UNIVERSITÉ | utc

heudiasyc