Futuristic Automotive Software Architecture

Microservices, Orchestration and Networked Hardware

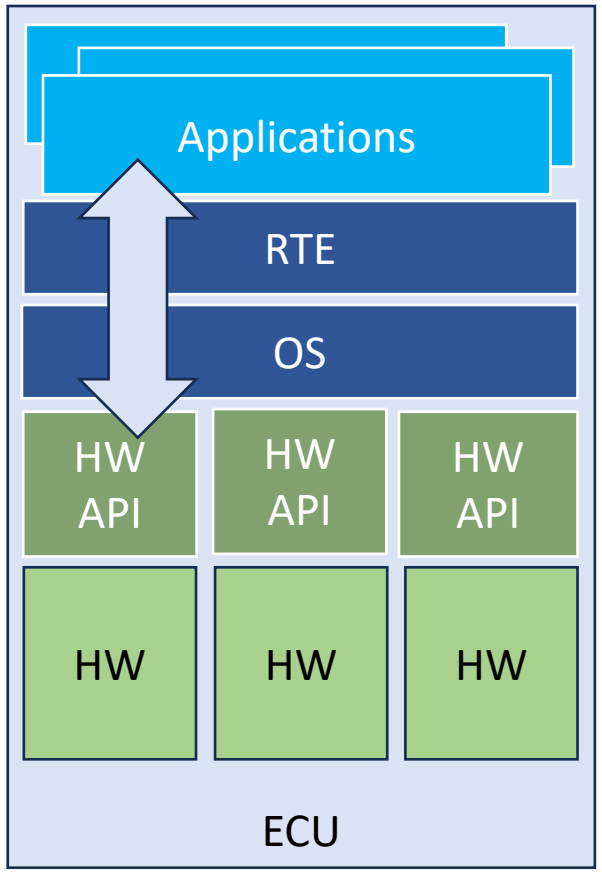
Anil Dhonde, Ethernovia Inc.

IEEE SA Ethernet & IP @ Automotive Technology Day
São Paulo, Brazil
September 2023

Agenda

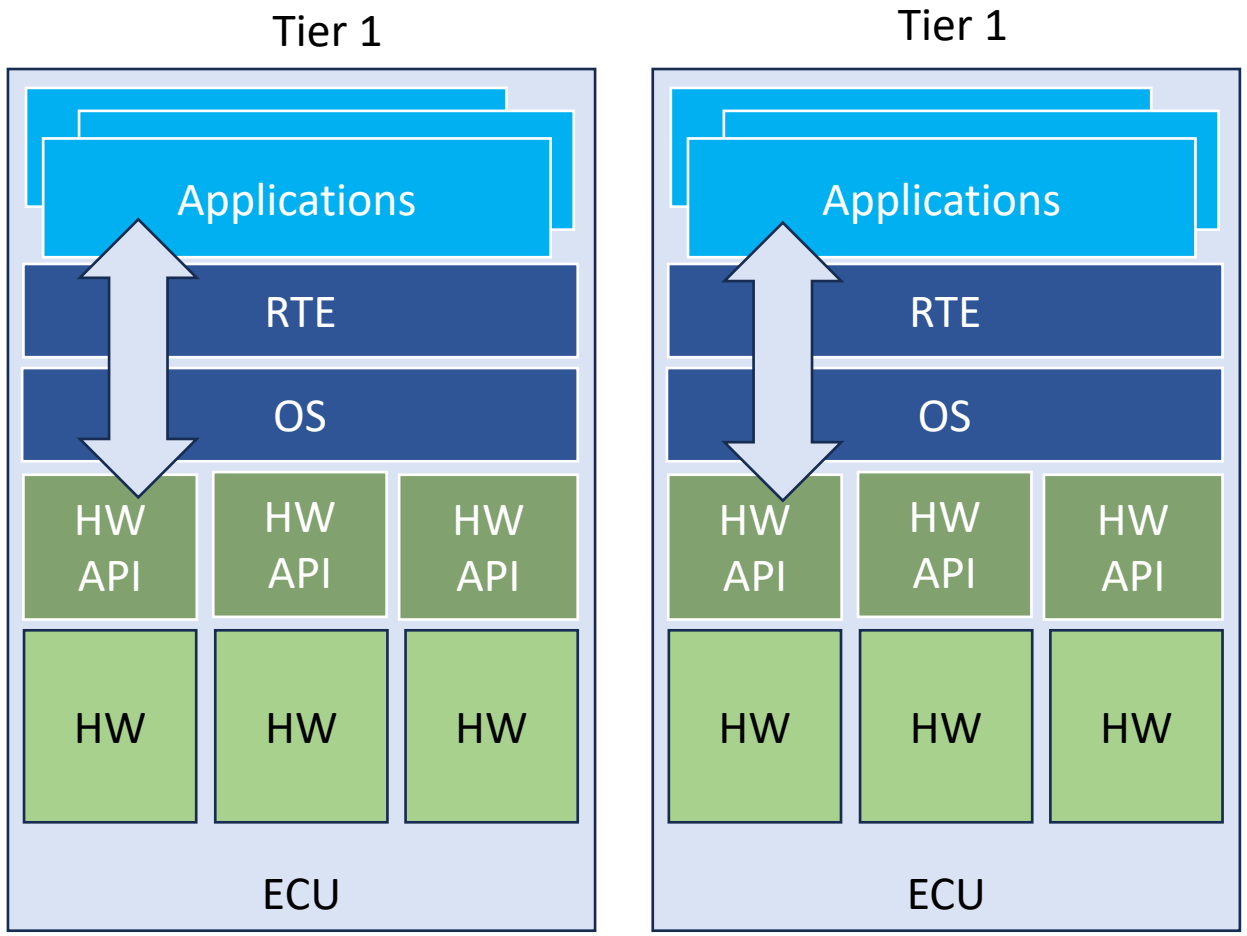
- Key Concepts
- Ethernet Today
- Futuristic Software Architecture
- DevOps
- Pros and Cons
- Future Work

Traditional Monolithic SW Arch



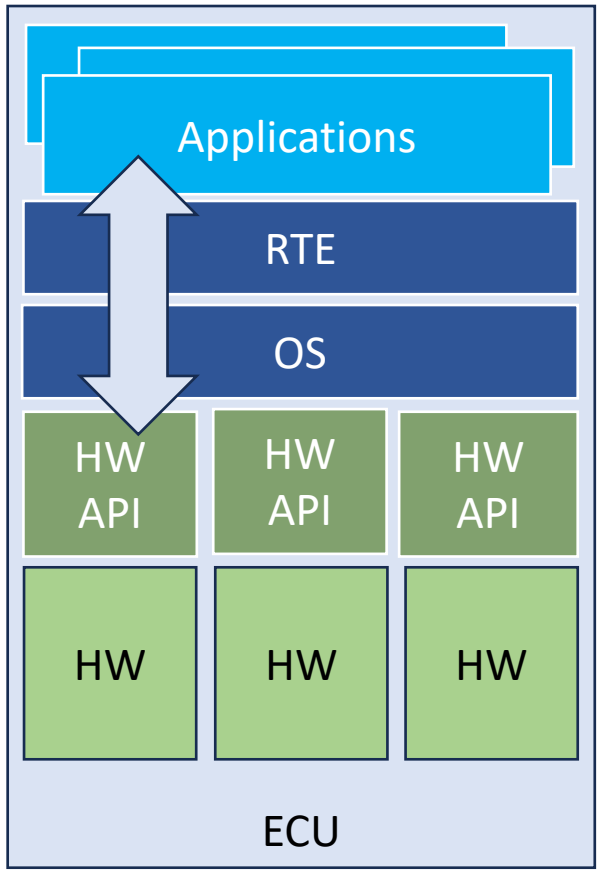
- Tightly Coupled Architecture
  - Applications chained to HW
- One SW Image
  - Single Build
  - Build/Test for every change
  - Major Release/Update

OEM Software Supply Chain

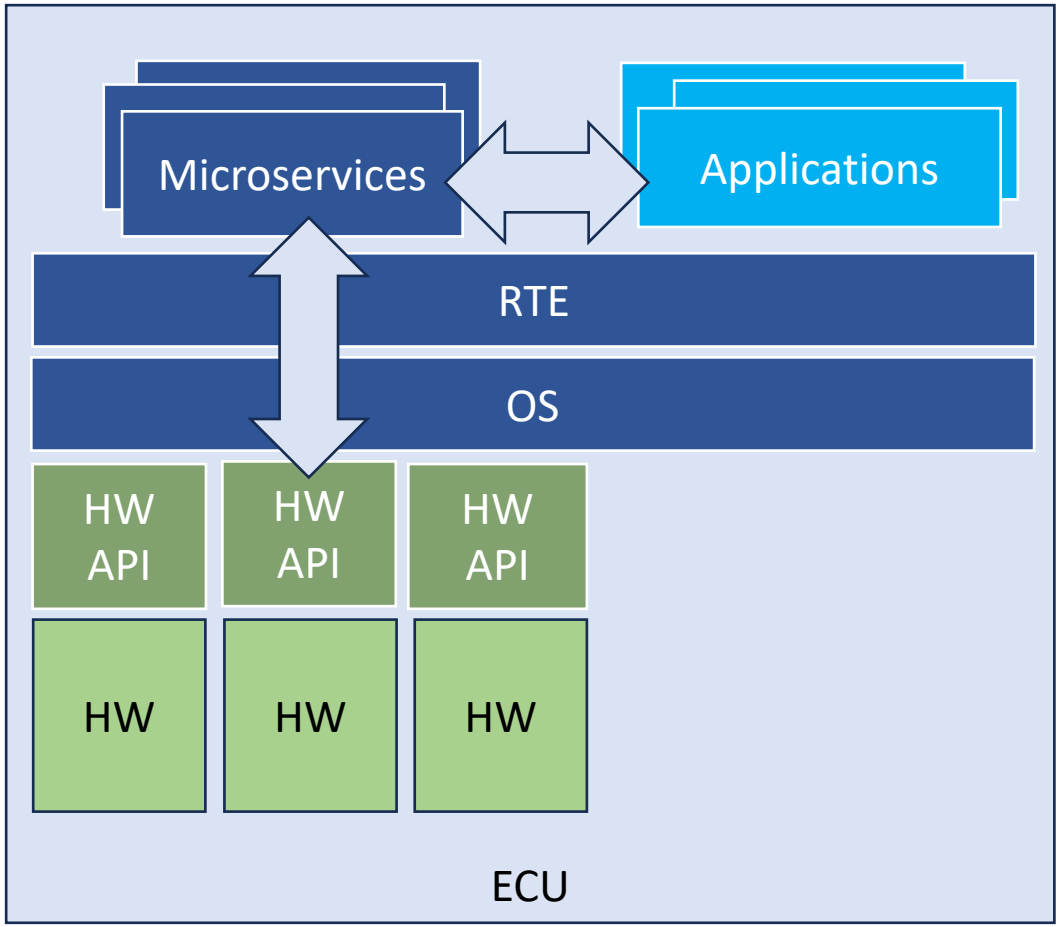


- OEMs offload SW to Tier 1's
- Multiple SW suppliers
  - Different platforms
  - Different architectures
- Painful integration
  - Test, Release, Update

Monolithic -> Microservices

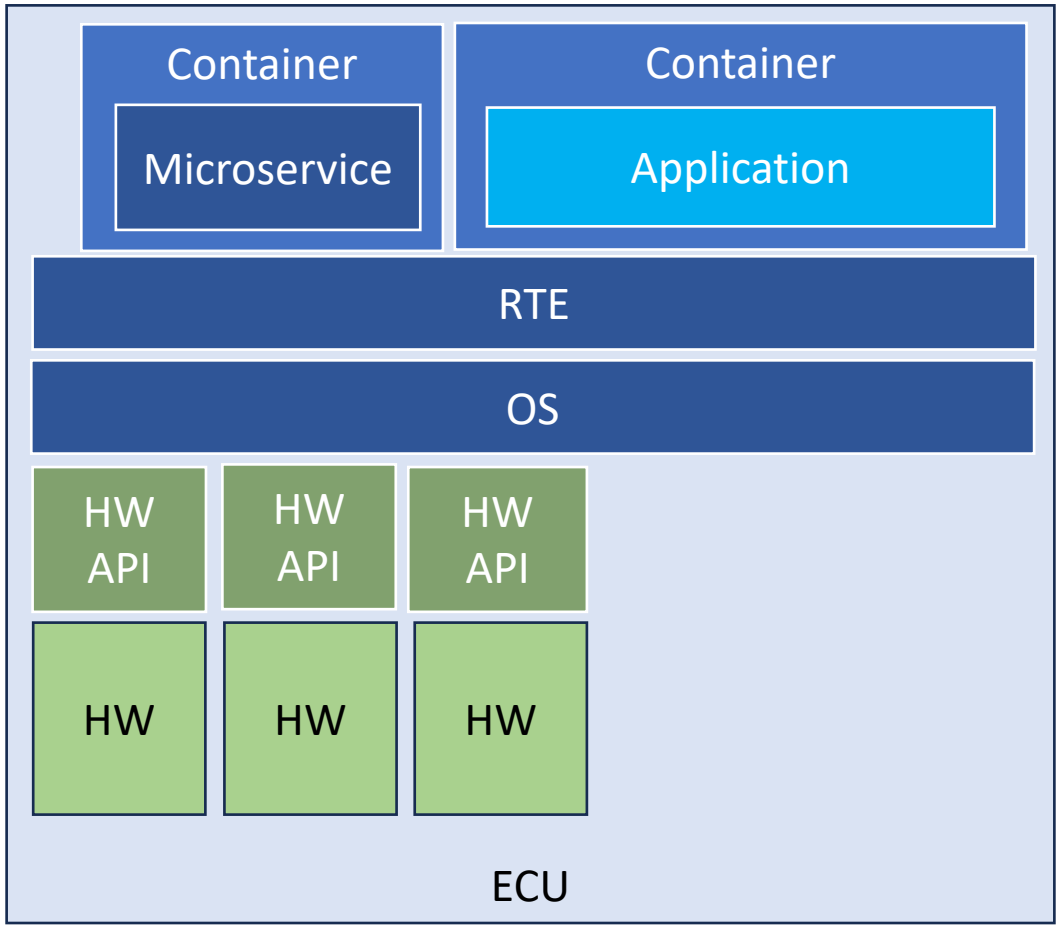


- Loosely coupled
- Network comm
  - Microservices
  - Applications
- Location Freedom
- Design breakdown
- Independent Lifecycle
  - Build, Release, Update



Containers

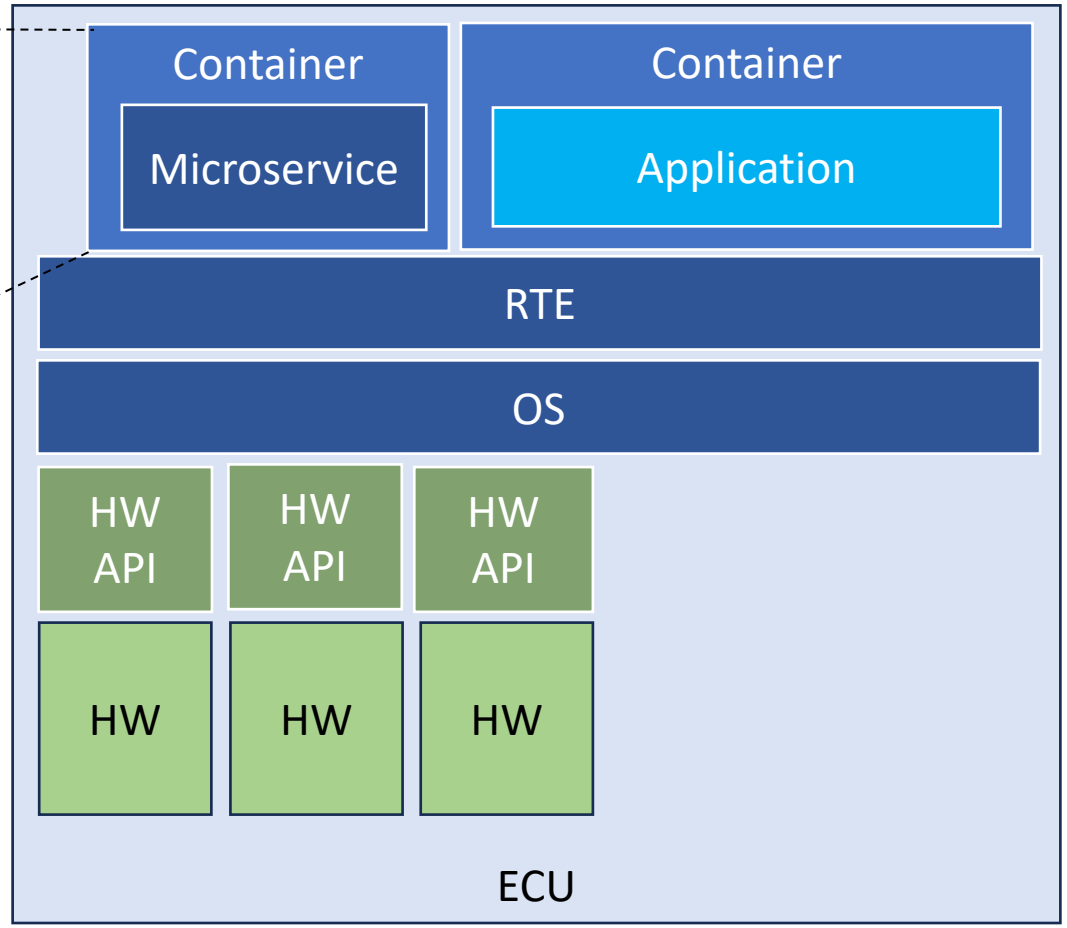
- Containers Manage
  - Microservices
  - Applications
- Platform independence
- Libraries and services, ex: network
- Build Once, Run Anywhere



Orchestrator

Orchestrator

- Manage Containers @ Run Time
- Configuration, Lifecycle
- Load Balancing
- Redundancy, Fault recovery
- Routing services



OEM's Software Destiny

- Software is the Differentiator of the future
- OEMs want to build Software applications in-house
  - Some OEMs have already setup up software entities
- OEMs will need full control over the Software destiny

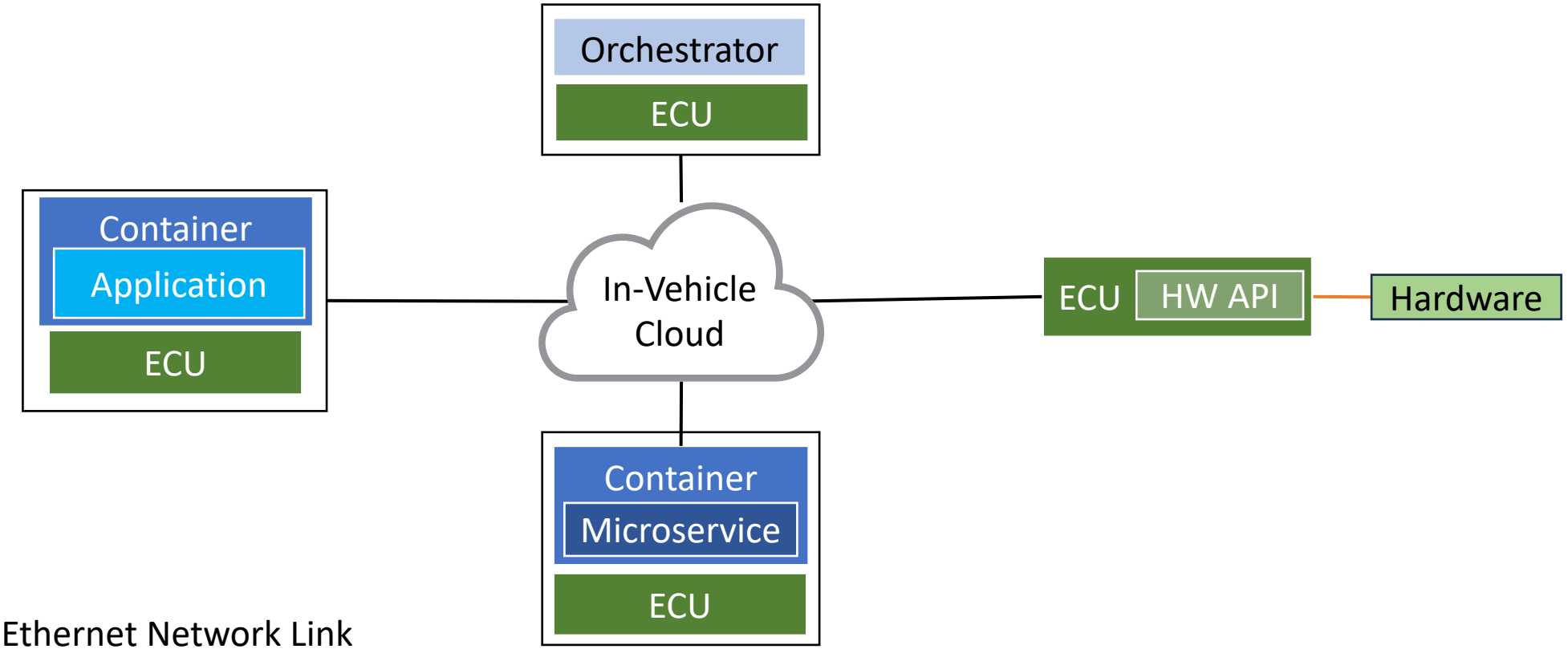
Ethernet Today

- Due to the ongoing work in IEEE802.1 TSN group, Automotive Ethernet is now:
  - Time Bound
  - Deterministic
  - Reliable

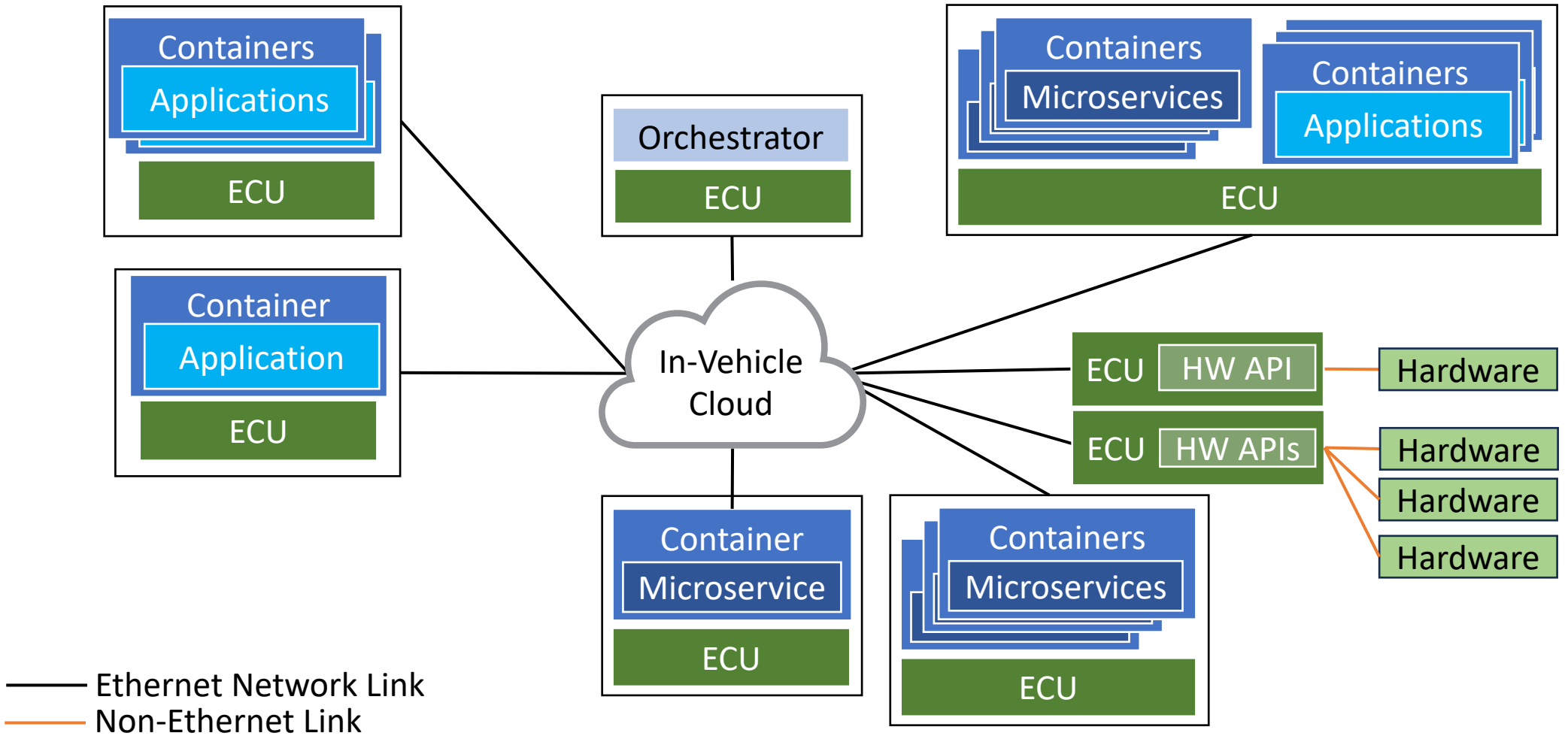
… hence, we can deploy a Software Architecture with Ethernet as the key enabler!

Futuristic Architecture

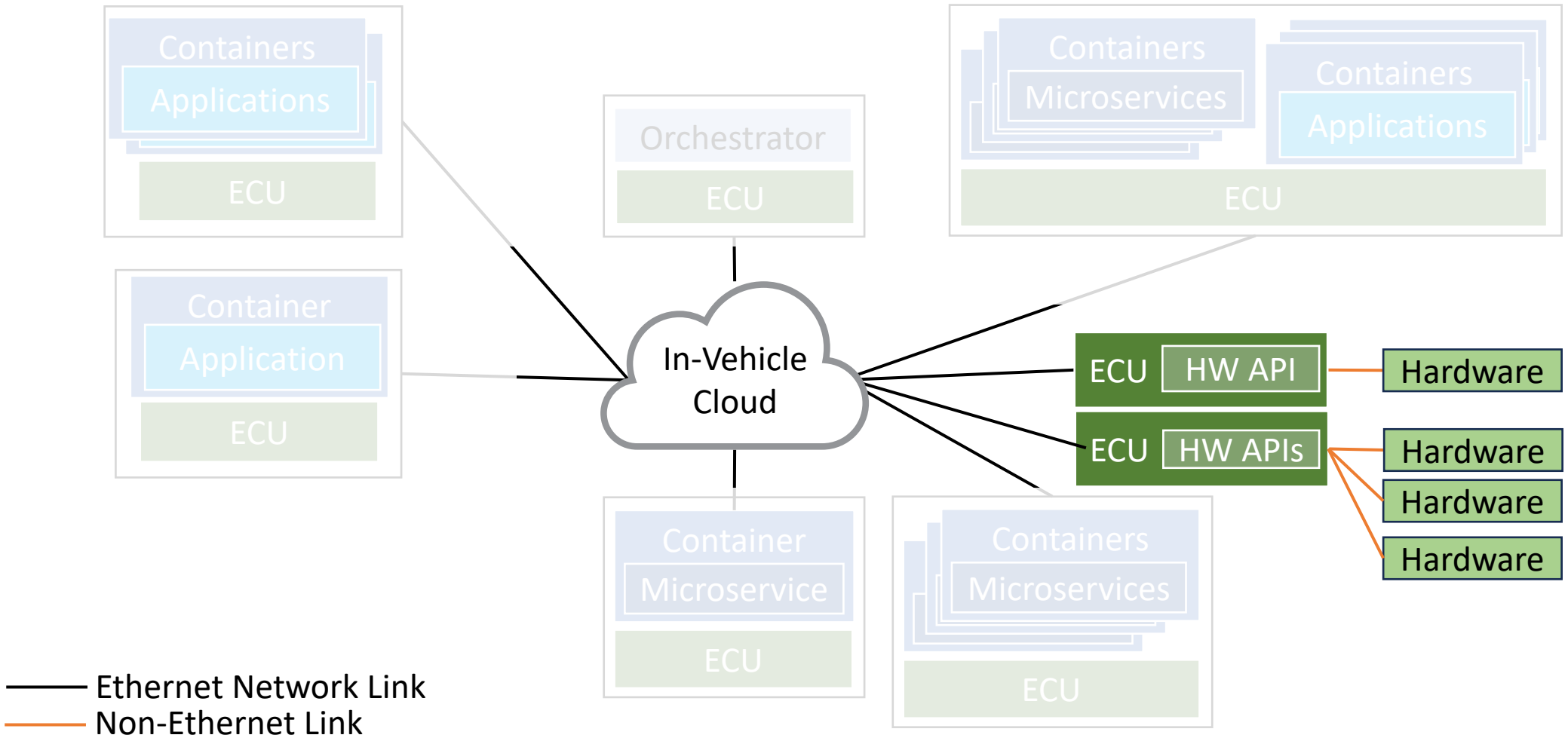
Architecture: In-Vehicle Cloud



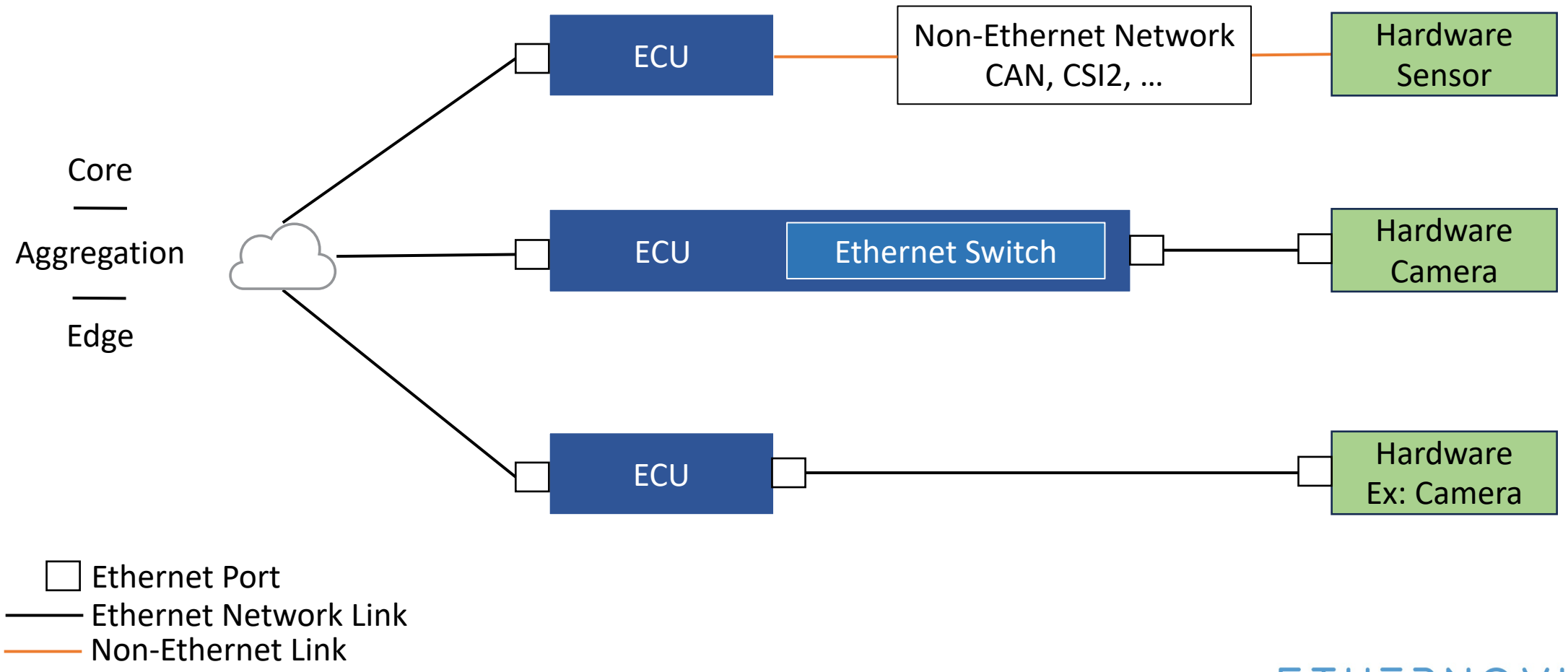
Almost Endless Possibilities



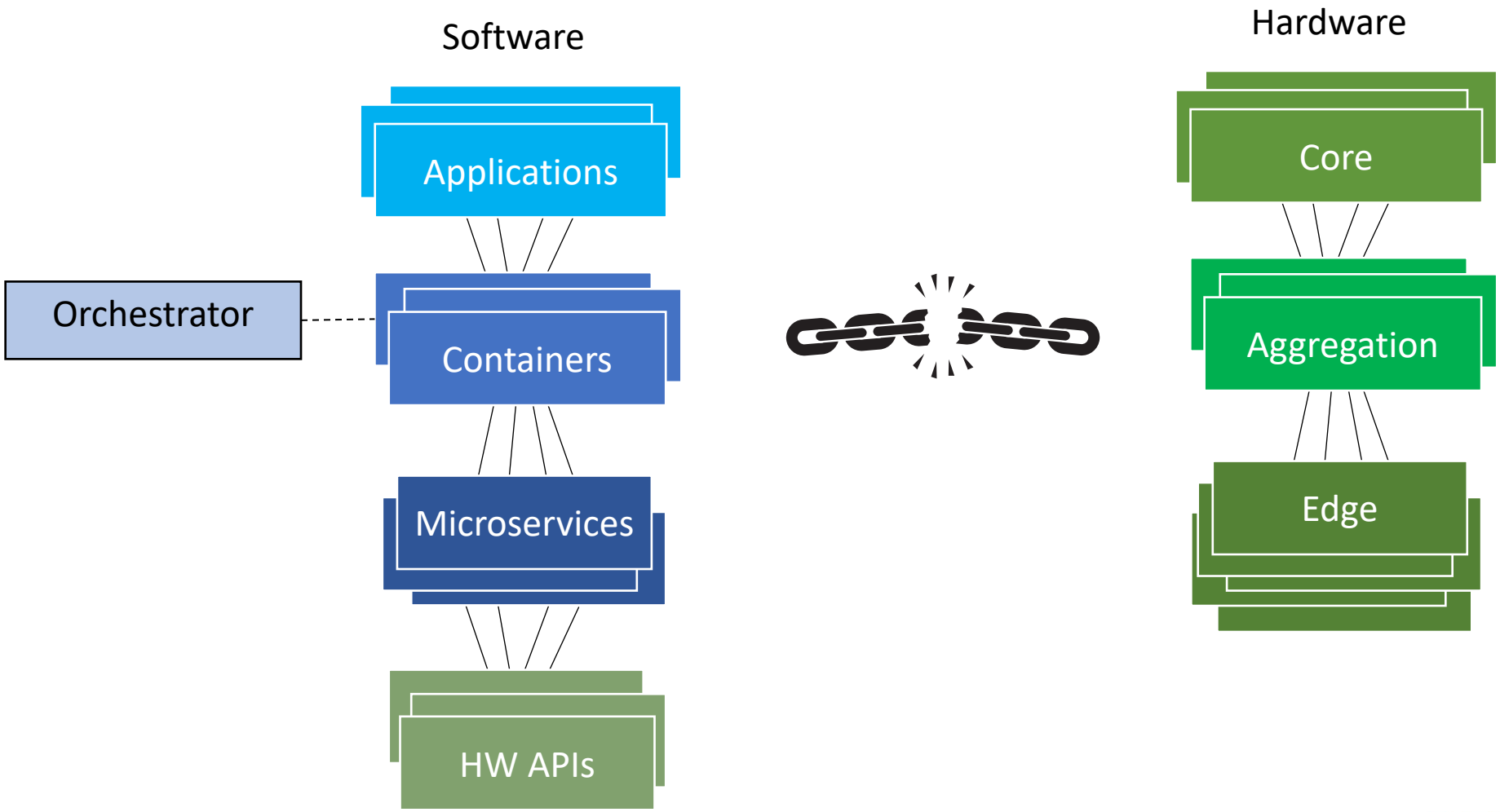
Networked Hardware



Ethernet Ubiquity



Independent Stacks

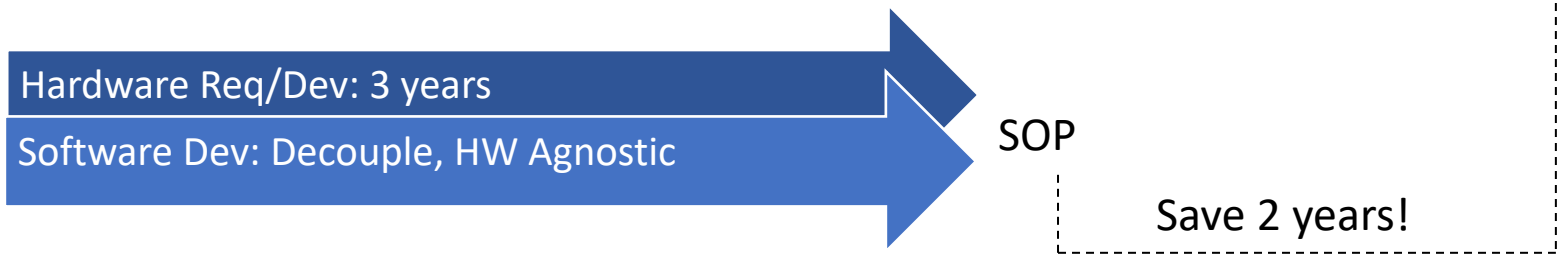


DevOps

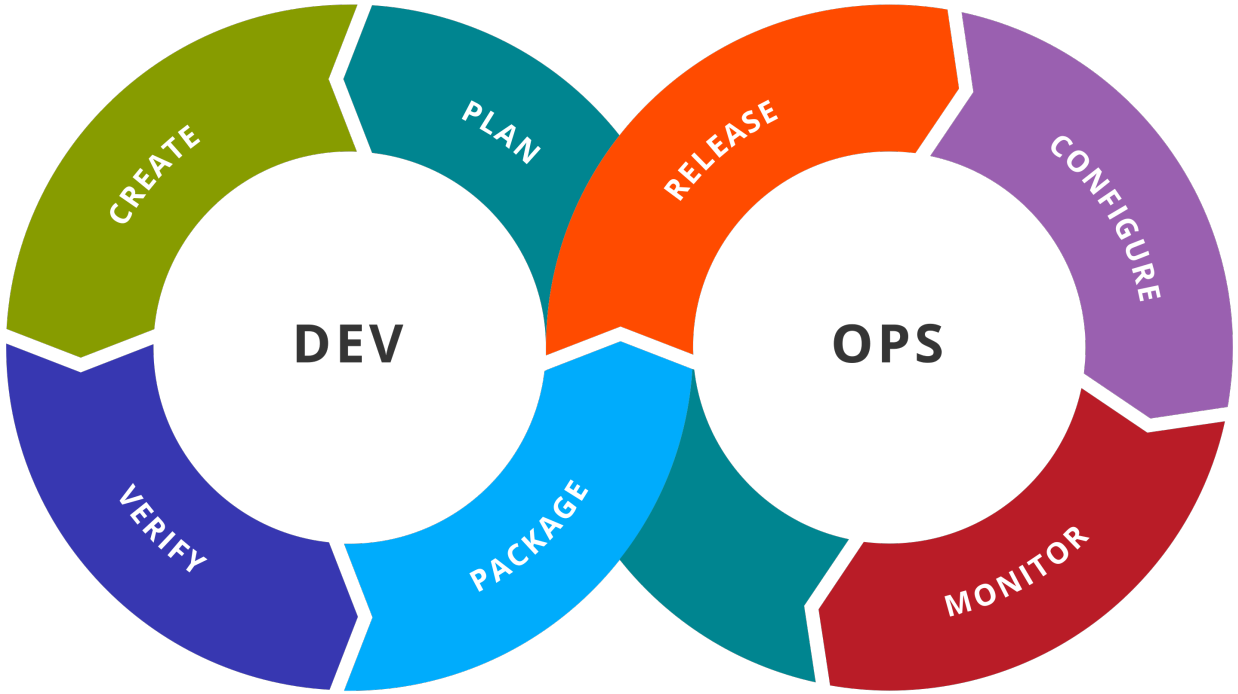
Development & Operations

Traditional

Future

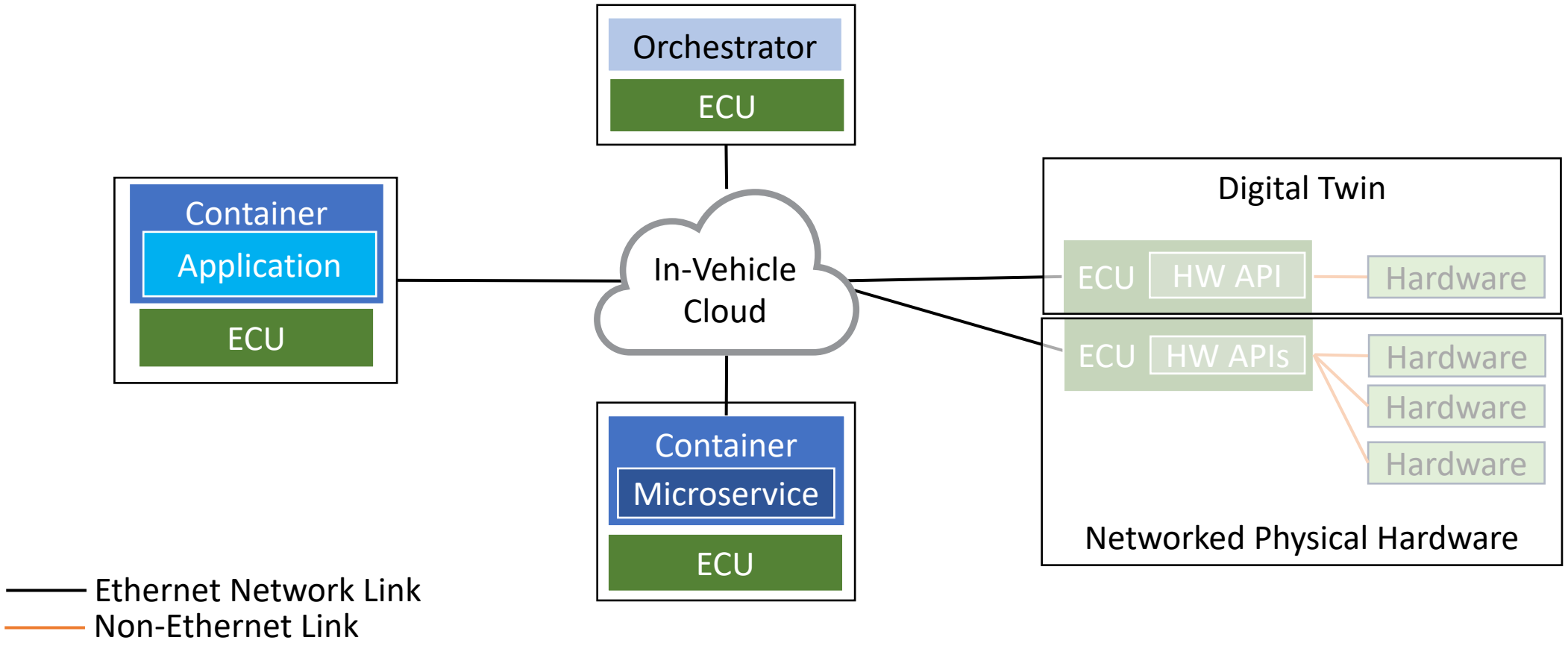


Dev and Ops

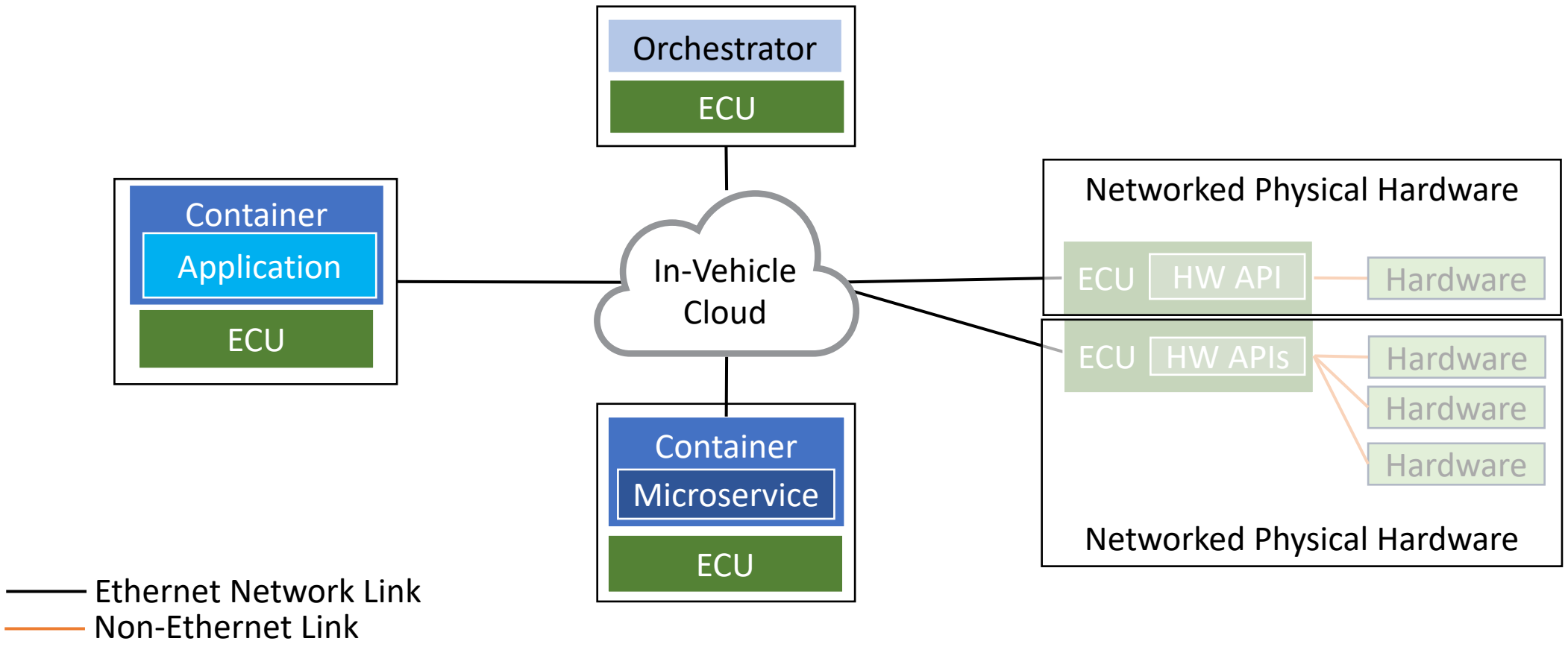


Source: https://commons.wikimedia.org/w/index.php?curid=51215412

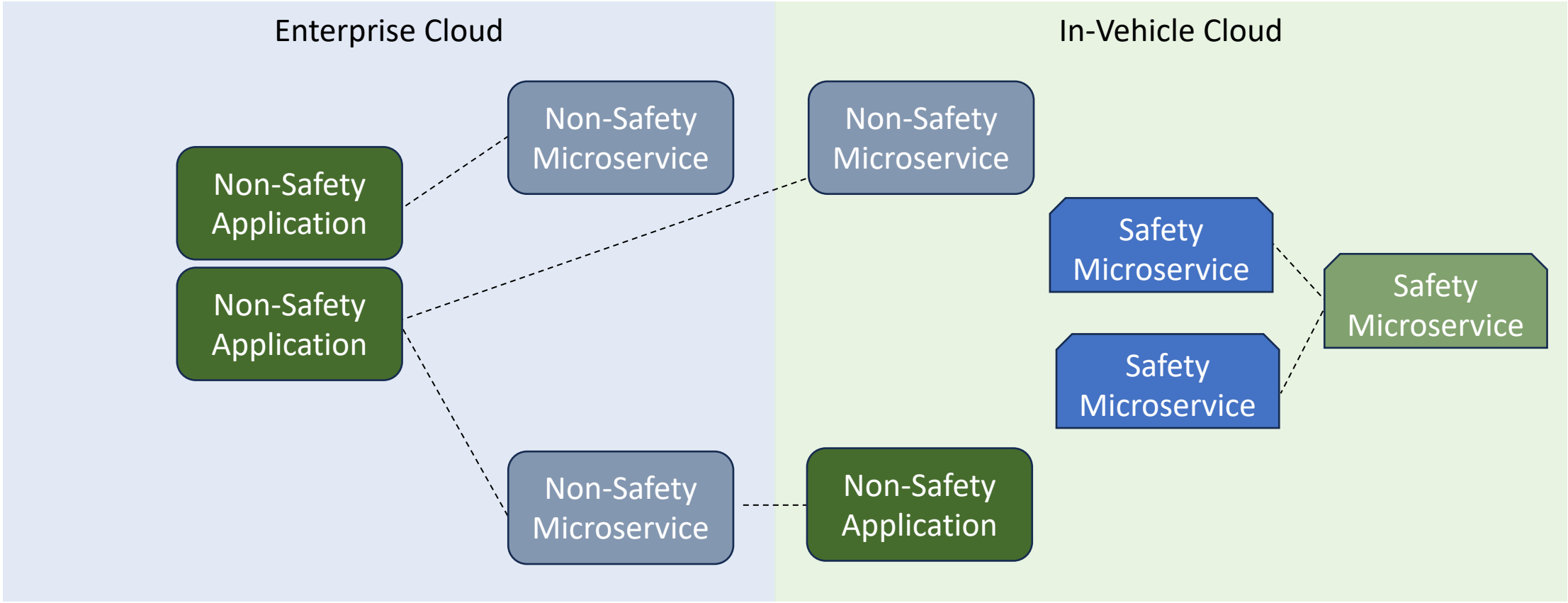
Dev: Virtual or Networked HW



Ops: Networked Physical HW



Mix Of Enterprise and In-Vehicle Cloud



Pros and Cons

Pros: Dev

- Complexity breakdown
- Scalability
- Independence: Processing platform, Hardware and Topology
- Agile: Move away from Waterfall, Shift Left (Early Validation)
- Supplier friendly development
- Enhancement and Future revenue streams
- Hybrid possibilities: “Move” ECUs to the Cloud

Pros: Ops

- Time to market
- Load Balancing
- Failure Detection, Isolation and Recovery
- Lifecycle/Updates: CI/CD
- Canary deployment
- Policing

Cons

- Cultural shift, steep learning curve
- Validation: Limit endless possibilities
- Overhead: Processing, Memory
- Legacy hardware integration
- Compatibility

Summary

- Monolithic SW is difficult to Develop and Deploy
- Ethernet allows Breaking Down the Monolith
- Futuristic Software Architecture
  - Endless Possibilities
  - Ethernet Uniquity
  - Networked Hardware
- Benefits for DevOps
- Mix of Enterprise and In-Vehicle Cloud

Future Work

- Lightweight Containers and Orchestrators
- Networked Hardware and Standardization of APIs
- Safety, Reliability and Security needs a modern approach

Concluding Thoughts

Software: Think Cloud First.

Hardware: Ethernet Ubiquity.


ETHERNOVIA
Thank You