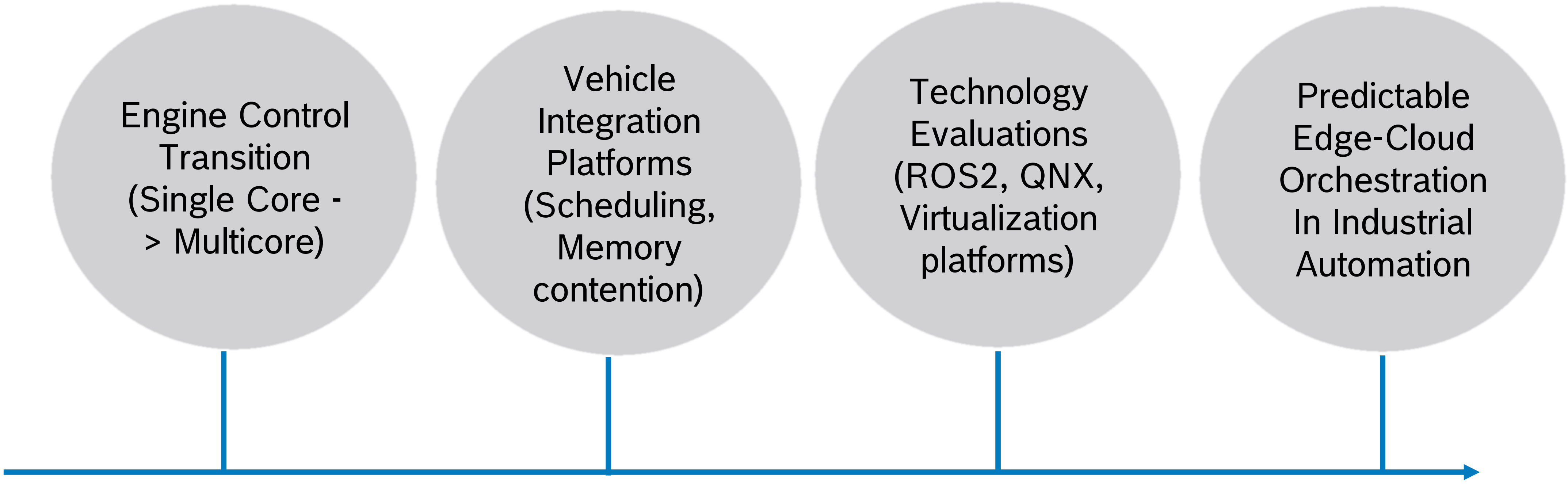
Temporal Symphony: Meeting the Timing Challenge in Edge Computing

Dakshina Dasari

BOSCH

Research Work at Bosch



BOSCH

Cloud Native meets Embedded Real-time Computing

cloud native computing

safety-critical realtime computing

*Breaking Down the Edge Continuum - LF Edge

Convergence of Technologies in the IT/OT world

BOSCH

Meeting the Timing Challenge in Edge Computing
The Industrial Metaverse

Source: ABB

The Industrial Metaverse is a world ...

- where **the physical and the digital worlds co-exist, collaborate**
- Playground for Immersive training and AI enabled data-driven decision making
- where **problems** can be **found**, or **discovered before** they arise

BOSCH

The Industrial Metaverse: Virtual Commissioning and Digital Twins

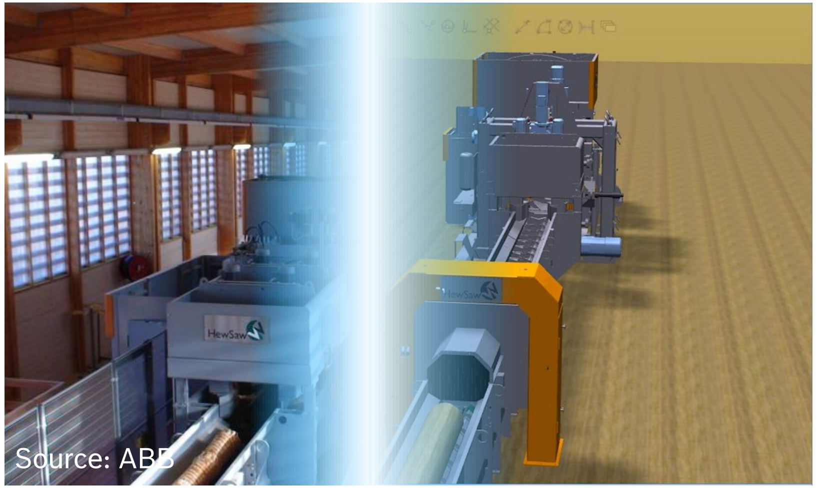
Virtual model of production system: Staging area

Simulate the behaviour of the real-system

- Prototype and Validate new concepts
- Seamless transition from design to production

Digital replicas of physical assets/plants

- Real-time data acquisition, monitoring, simulation
- Live Feedback, Predictive Maintenance



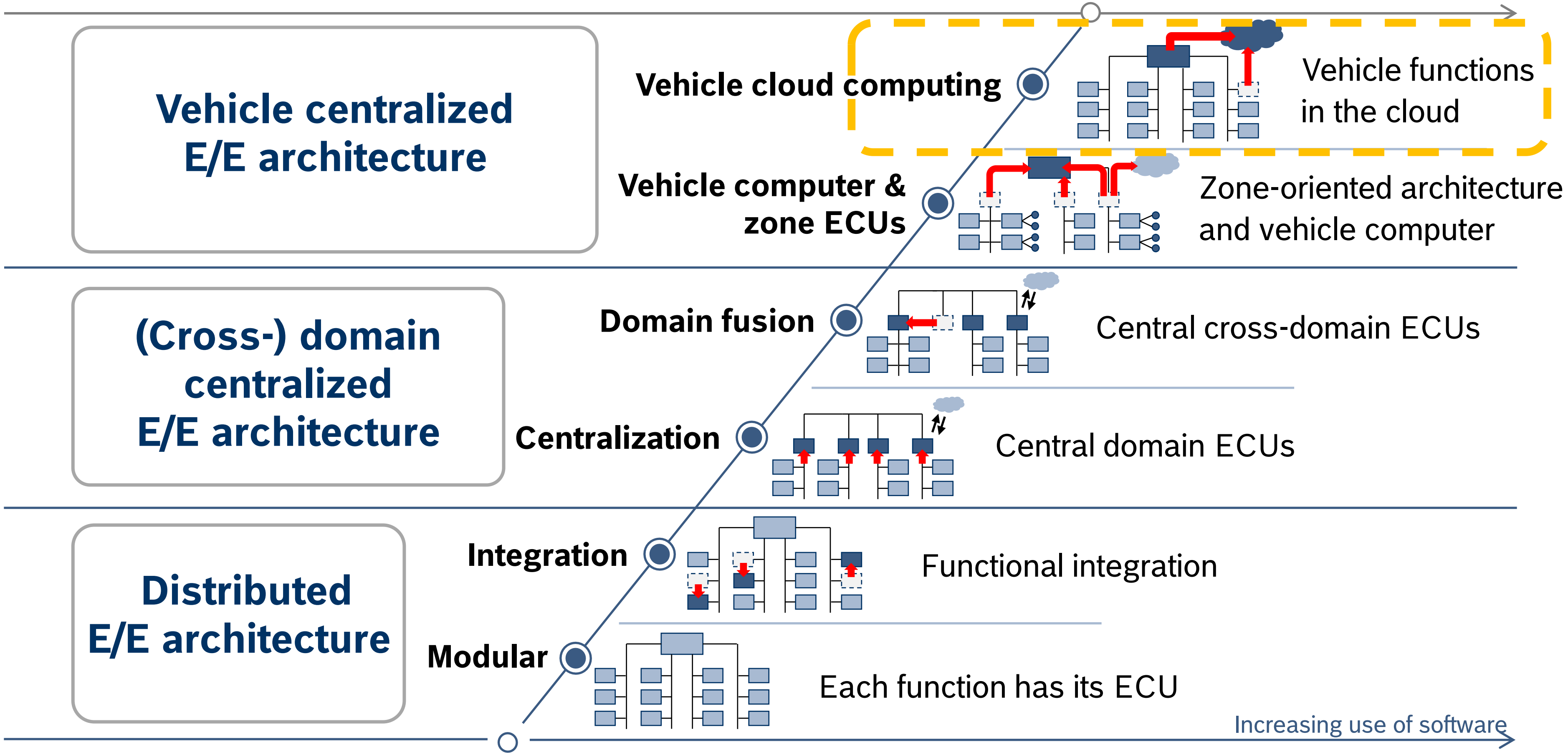
Virtual Commissioning

Digital Twin

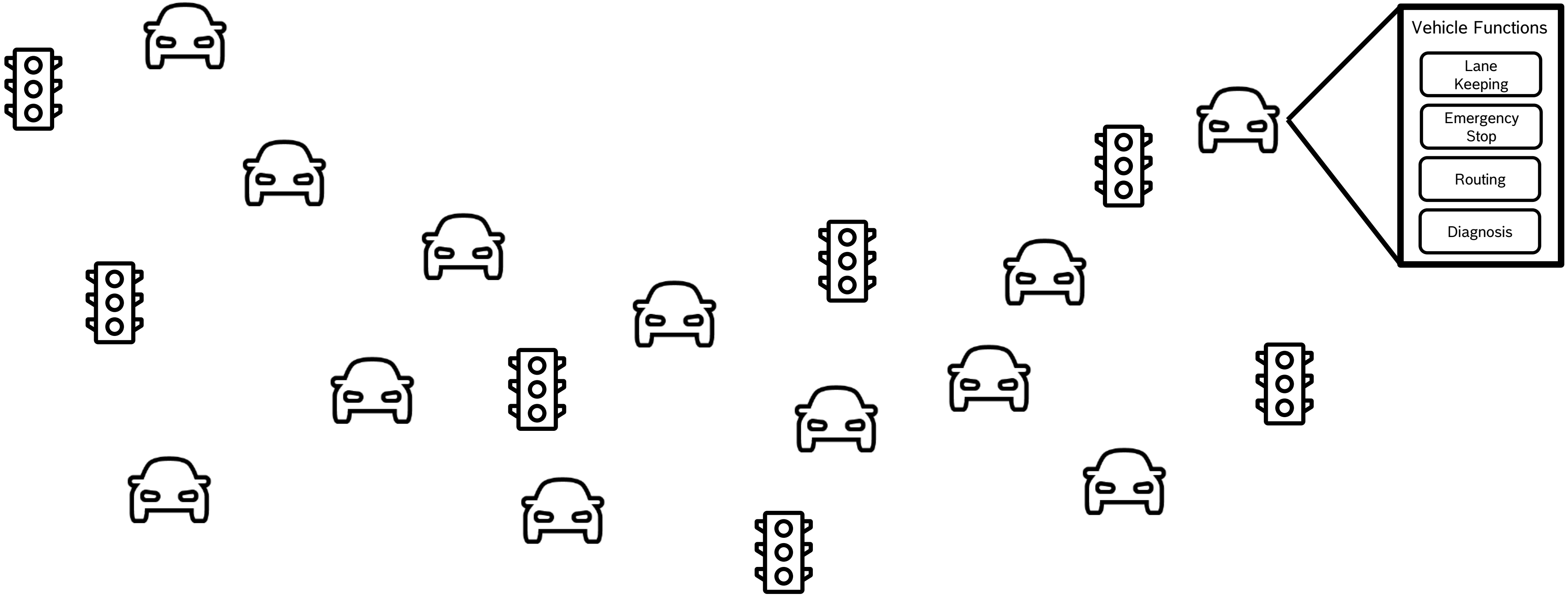
Common Denominator: Compute Intensive Simulations + Real-time Capabilities

Meeting the Timing Challenge in Edge Computing

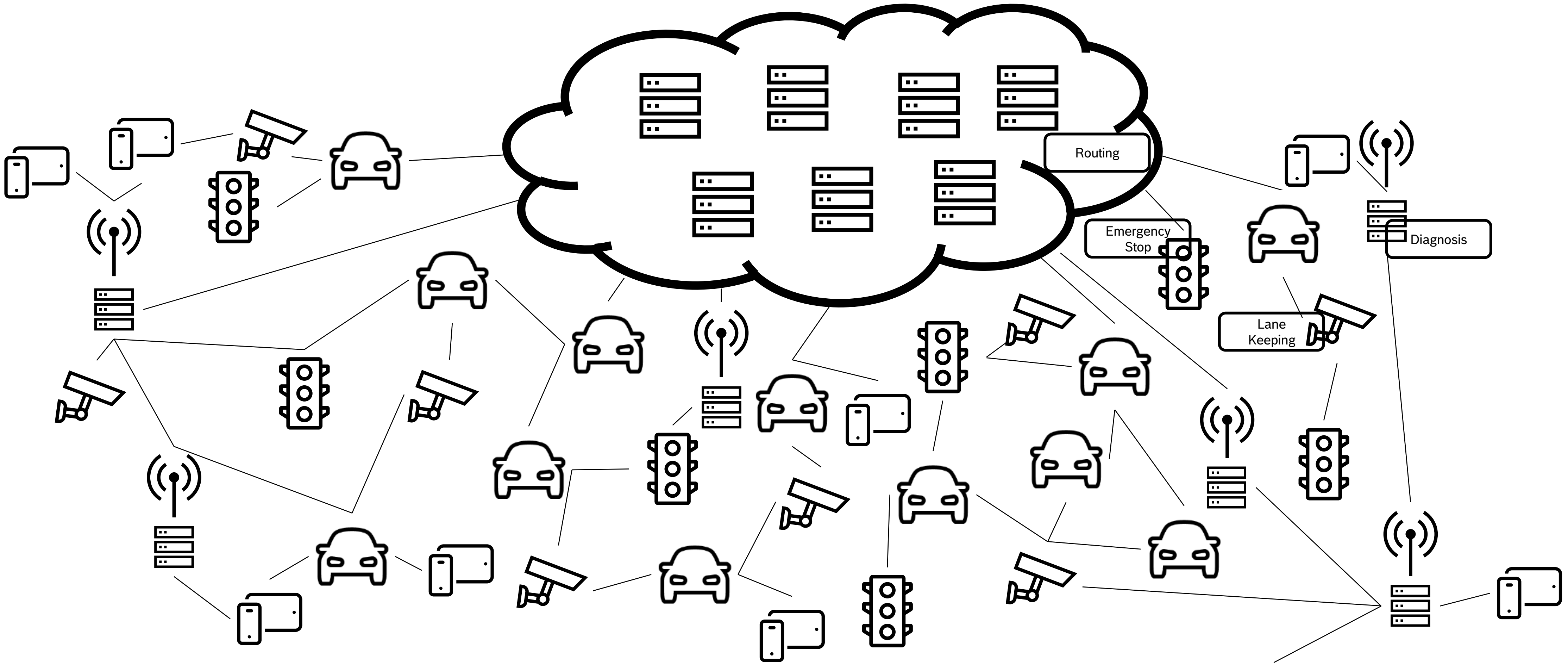
The Trend of Offloading to the Edge: E/E Architectures



Meeting the Timing Challenge in Edge Computing
Automotive Domain: In-vehicle Functionality

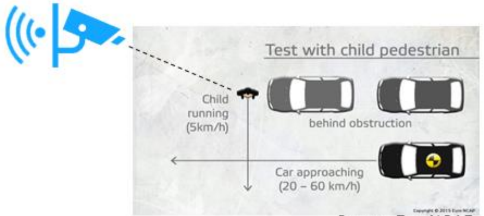


Meeting the Timing Challenge in Edge Computing
Automotive Domain: Function offloading to the Cloud/Edge



Meeting the Timing Challenge in Edge Computing
Improved Functionality with Function Offloading

Bird’s eye view



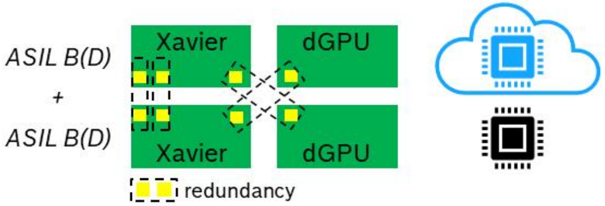
Source. EuroNCAP

vs. limited in-vehicle visibility or overly conservative behavior

Safe traffic light information

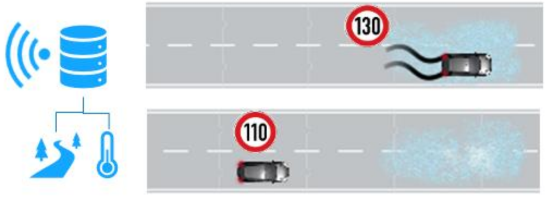
vs. less-dependable & costly (data+compute) determination via AI detectors

Redundancy in the cloud



vs. complex & costly (compute+peripherals) redundant HPC architecture

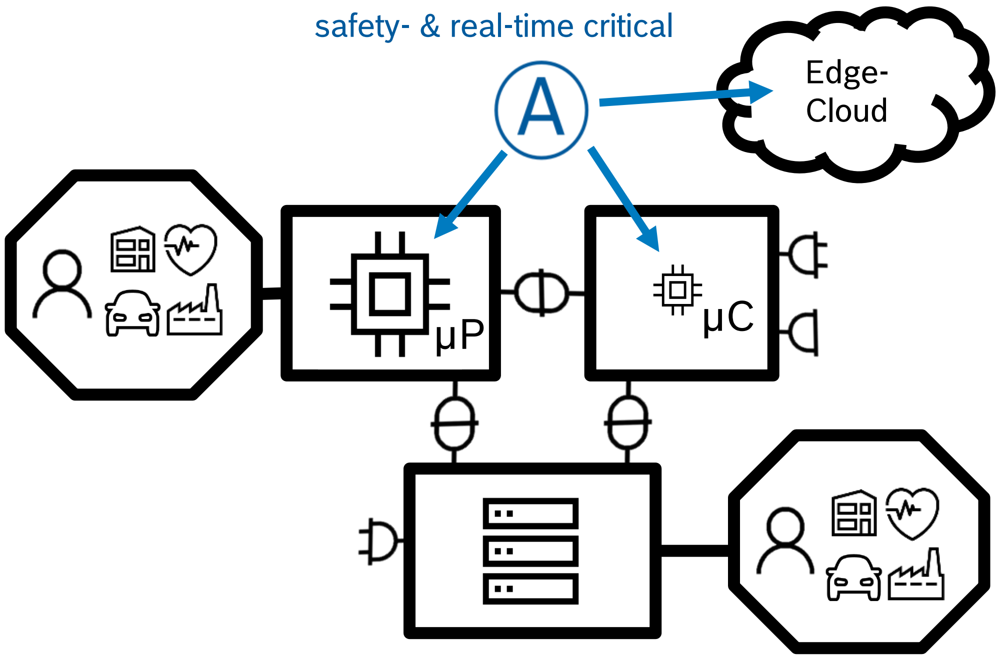
Safe road condition information



vs. less-dependable direct measurement or conservative fixed parameterization

Improved Safety Functions by leveraging the power of the Cloud

Meeting the Timing Challenge in Edge Computing
Merging both the Worlds



- Embedded software architectures for safety & real-time critical applications are **rigid and lack flexibility**
  - Applications are hard-baked onto embedded platforms
  - Vendor lock-ins, difficult to migrate, update
- IT software architectures are flexible but **do not scale to smaller devices** and do **lack capabilities to control QoS in a fine-grained manner**
  - Containers require hundreds of MB of RAM, suffer from high spin-up times, struggle to multiplex low-level devices (e.g., sensors)

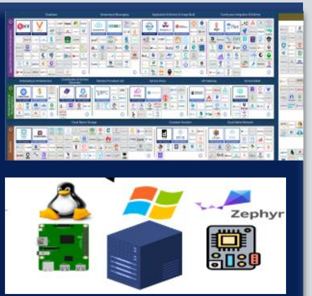
How can we leverage the strengths of both the worlds to create a framework for Reliable Distributed Systems?

Meeting the Timing Challenge in Edge Computing

What does not translate well to Edge systems

Stateful applications

Diverse Tech. Stacks



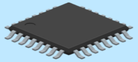
Heterogeneous Hardware

High power, homogeneous hardware

Data Center

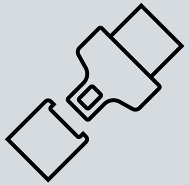
Personal Computer

Microprocessor



Microcontroller

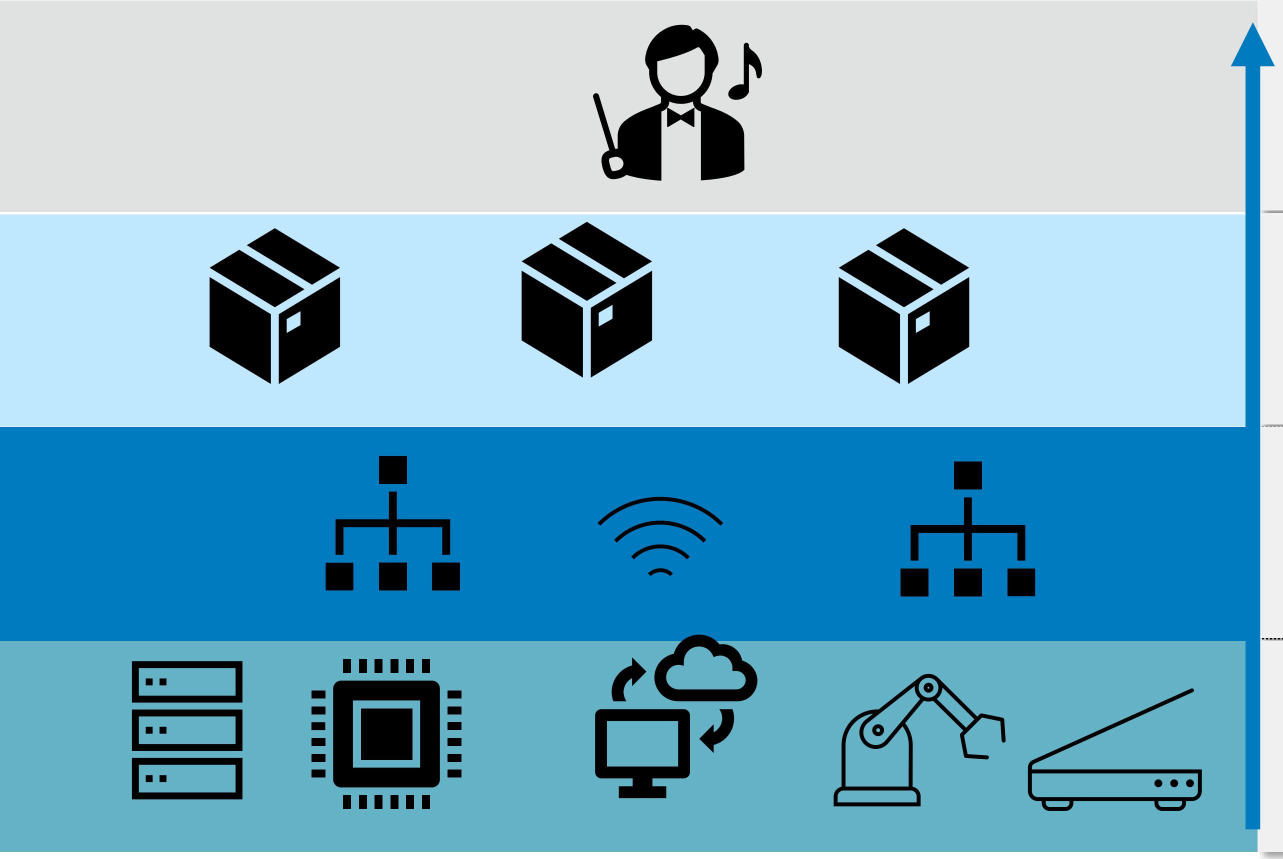
Low power, embedded hardware



Edge Dynamics: QoS, Safety, Security, Reliability, (With unreliable networks: Edge Autonomy)

Meeting the Timing Challenge in Edge Computing

A wish-list across the Edge-Cloud Continuum



Resource Aware Orchestration
(Failover, Scaling, Deployment, Resource Management)

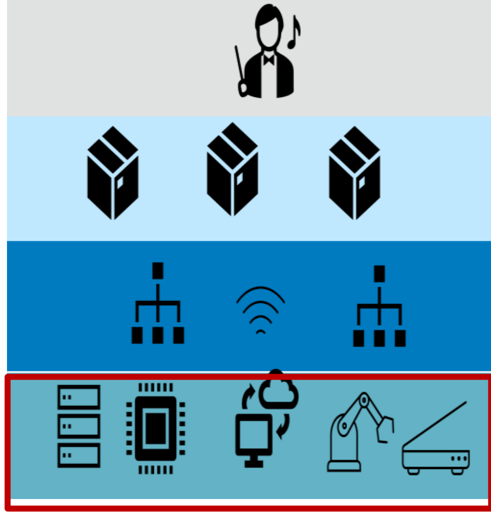
Lightweight Containerization
(Temporal Isolation, Low footprint, Arch-agnostic Runtime Engines)

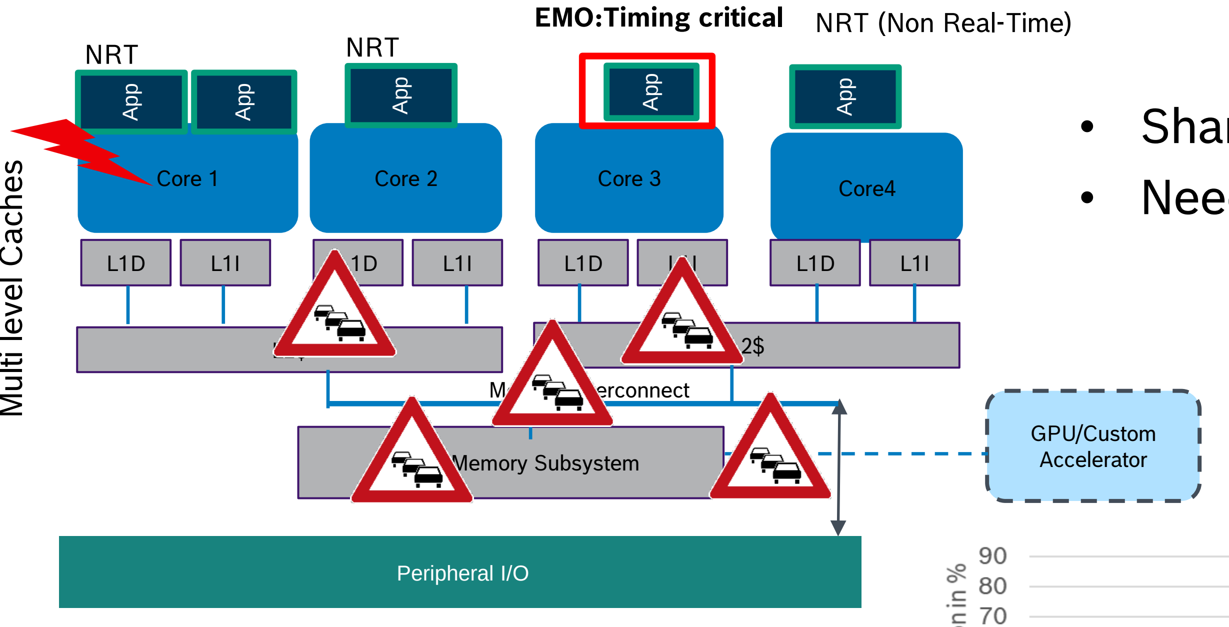
Mechanisms to guarantee Reliable Differentiated QoS on the Network

Resource Reservations, Safety Fallback Mechanisms, Resilient App. Design, Timing abstractions for distributed applications

Meeting the Timing Challenge in Edge Computing

Reconciling Performance and Predictability



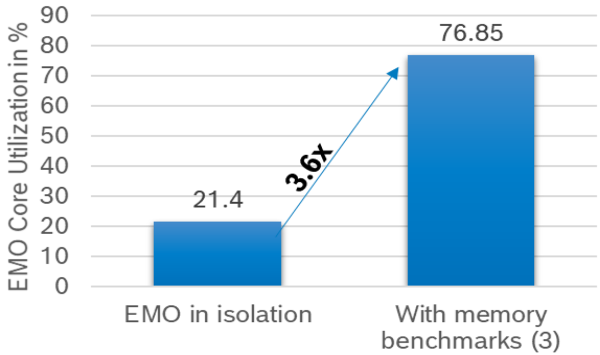


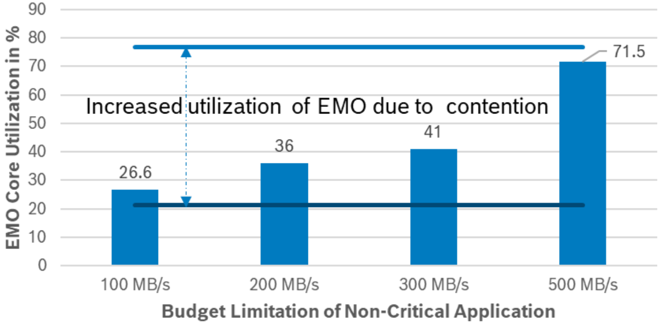
µProcessor

Simplified view Infineon NXP S32V

- Shared Resource Contention
- Need for resource reservations

- Software Approaches: Monitor and Regulate Approaches:
- MemGuard, MemPol

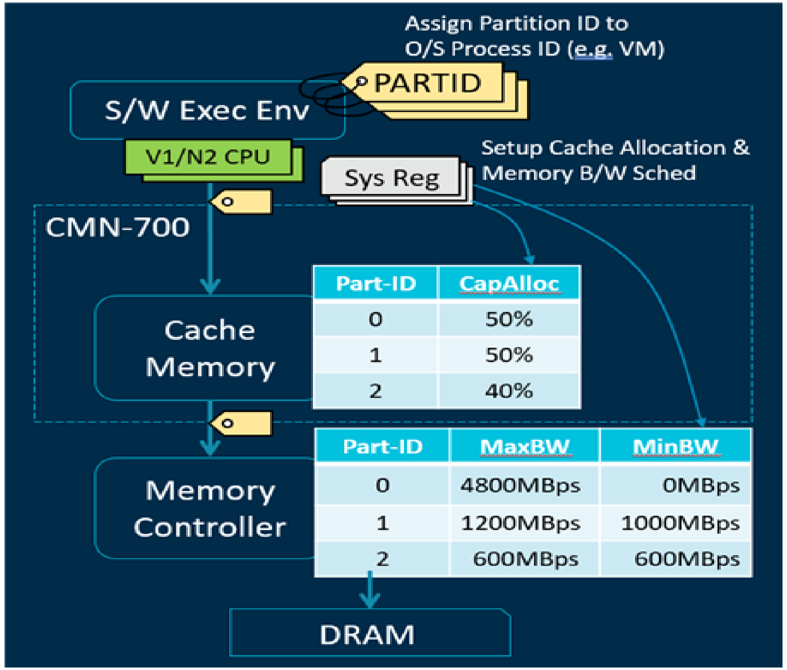




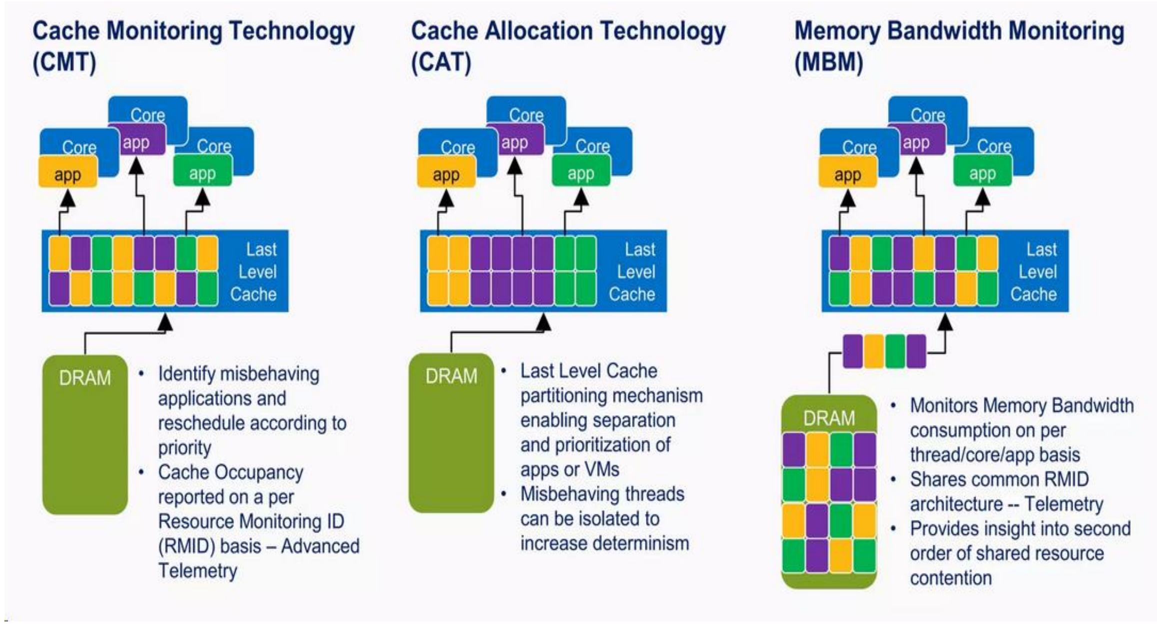
H. Yun, G. Yao, R. Pellizzoni, M. Caccamo and L. Sha, "MemGuard: Memory bandwidth reservation system for efficient performance isolation in multi-core platforms," *2013 IEEE 19th RTAS*

BOSCH

Meeting the Timing Challenge in Edge Computing
Hardware Mechanisms for Resource Regulation



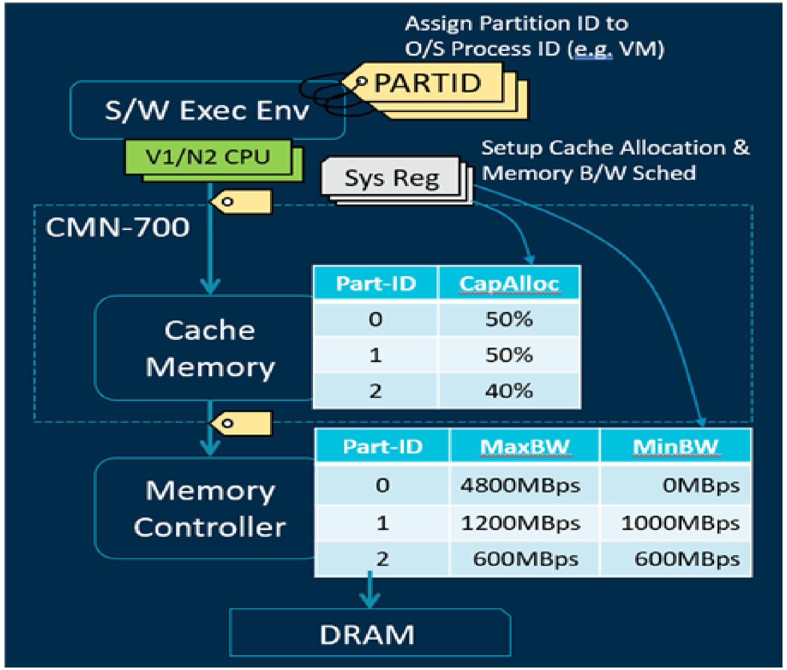
ARM Memory Partitioning And Management



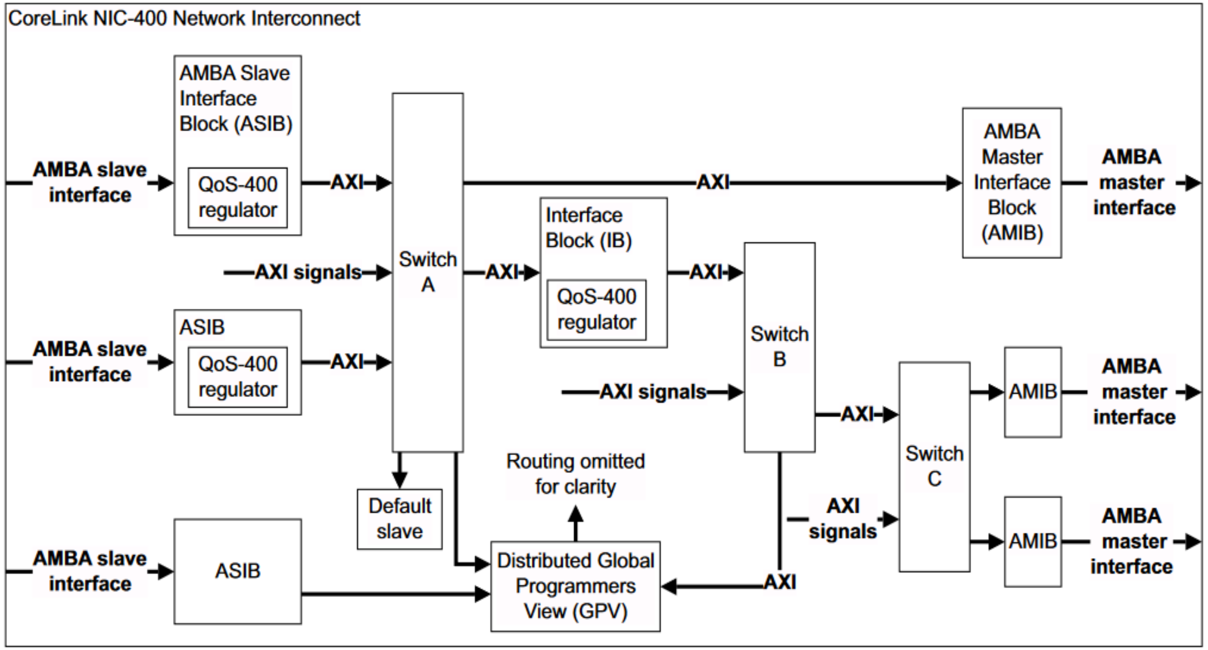
Intel Resource Director Technology

https://community.arm.com/arm-community-blogs/b/architectures-and-processors-blog/posts/arm-neoverse-n2-industry-leading-performance-efficiency

Meeting the Timing Challenge in Edge Computing
Hardware Mechanisms for Resource Regulation



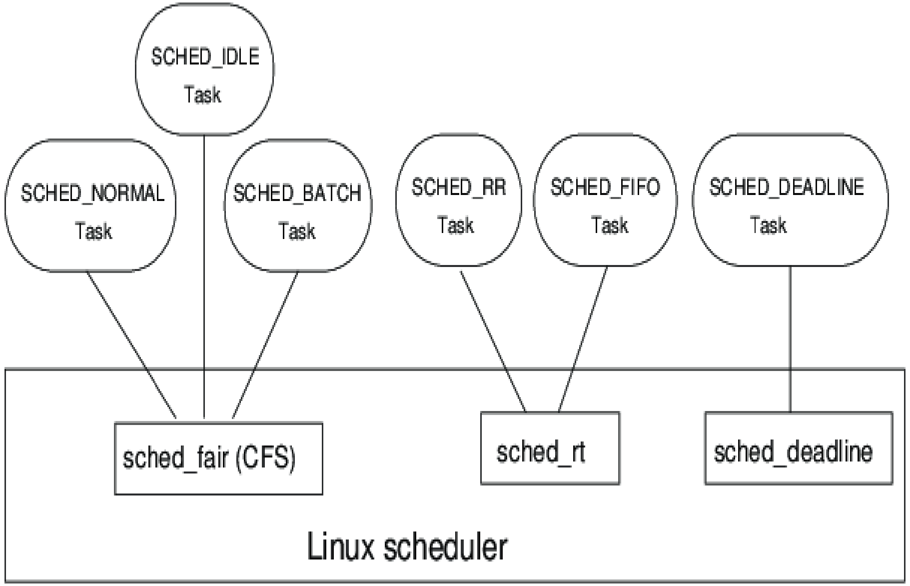
ARM Memory Partitioning And Management

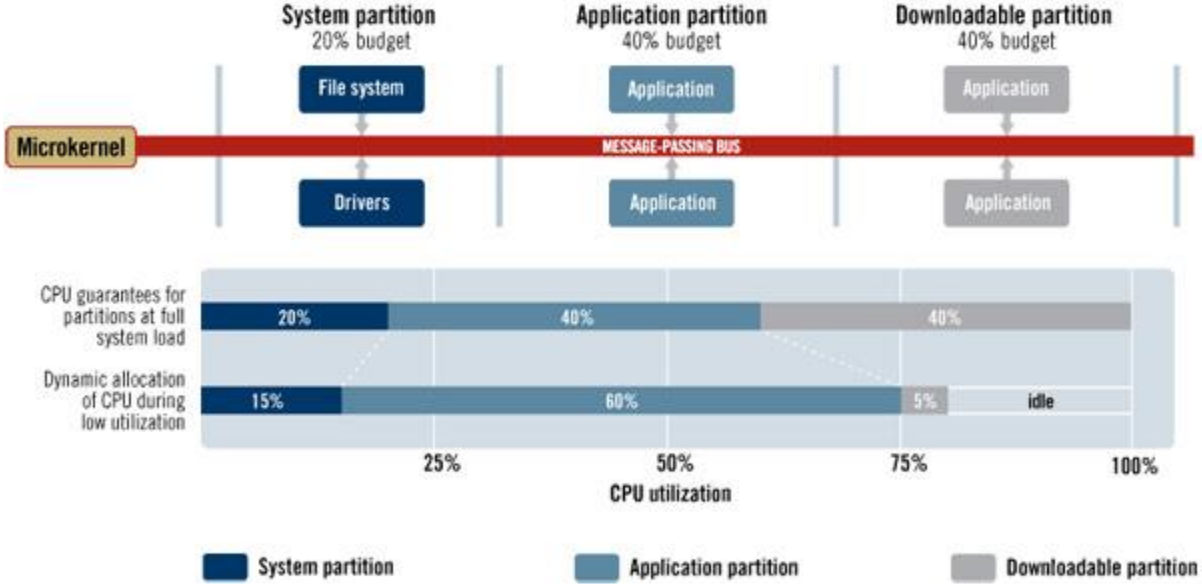


ARM Corelink NIC-400 Network Interconnect

https://community.arm.com/arm-community-blogs/b/architectures-and-processors-blog/posts/arm-neoverse-n2-industry-leading-performance-efficiency

Meeting the Timing Challenge in Edge Computing
Contention on the CPU ?



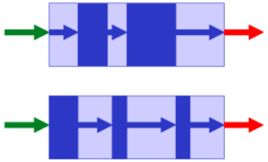


QNX Adaptive Partitioning Scheduler

Meeting the Timing Challenge in Edge Computing
Need for better execution control mechanisms

- **Accommodate Heterogeneous Applications**
  - Hard real-time applications (e.g. control)
  - Sporadic applications (e.g. predictive powertrain functionalities)
  - Applications with dynamic resource requirements (e.g. perception, MPC)

- **Efficient Temporal Isolation**
  - Temporal properties of an application shall be independent of other co-executed applications
  - Capability to use "all" system resources meaningfully
  - No hard and inflexible assignments

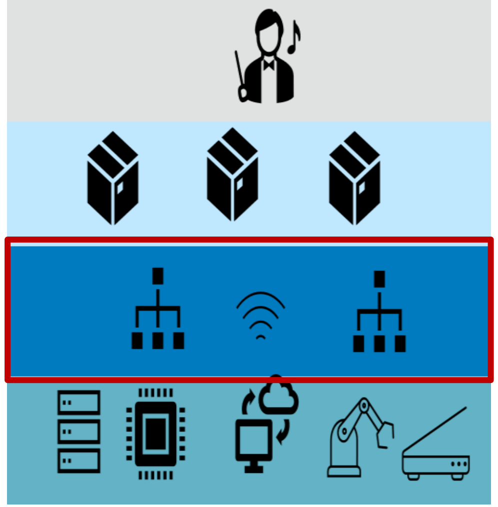


- **Controlled QoS**
  - Comprehensible abstraction for (computational) resources
  - Ensure application progress, prevent starvation
  - Ability to compute upper bounds on the response times tasks-chains ("analytically sound")

BOSCH

Meeting the Timing Challenge in Edge Computing

Exploring the Network Layer



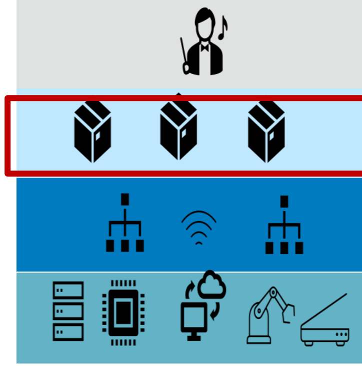
Increased Performance & Isolation

Network APIs to control Quality of Demand

TSN/DETNET: Need for easier configuration tools!

- Multivendor support

Need for contract based APIs



Meeting the Timing Challenge in Edge Computing
Exploring the Virtualization Layer

Hardware Level

Operating System Level

Application Level

Virtual Machines (QEMU, VMware, VirtualBox)

Lightweight VMs (KVM, firecracker)

Containers (LXC, Docker, Podman)

ByteCode Sandboxes (WASM / eBPF / Java)

General Apps
Lower Performance
(Larger, Slower, Higher Memory Ovehead)

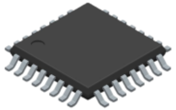
Customized Apps
Higher Performance
(Smaller, Faster, Lower Memory Overhead)

High power, homogeneous hardware

Data Center

Personal Computer

Microprocessor

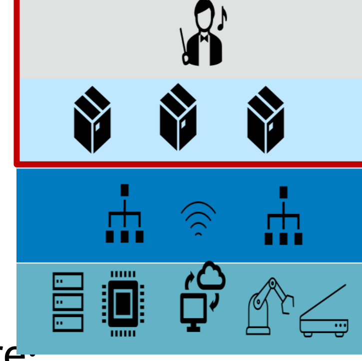


Microcontroller

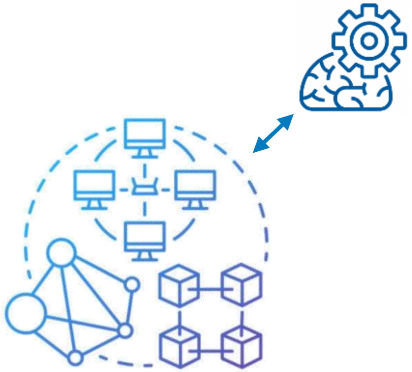
Low power, embedded hardware

BOSCH

Meeting the Timing Challenge in Edge Computing
Silverline: An Edge Orchestration Framework



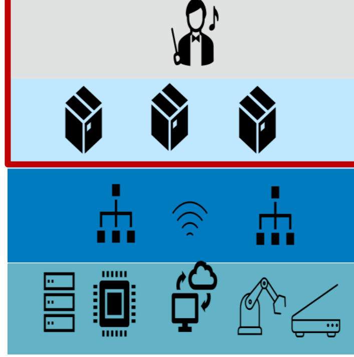
- **Lightweight virtualization with WebAssembly**
  - Enable deployment of applications on arbitrary modern hardware: programming paradigm spanning cloud, edge, and device
  - Fast, safe and portable execution semantics
  - Hardware & language independent

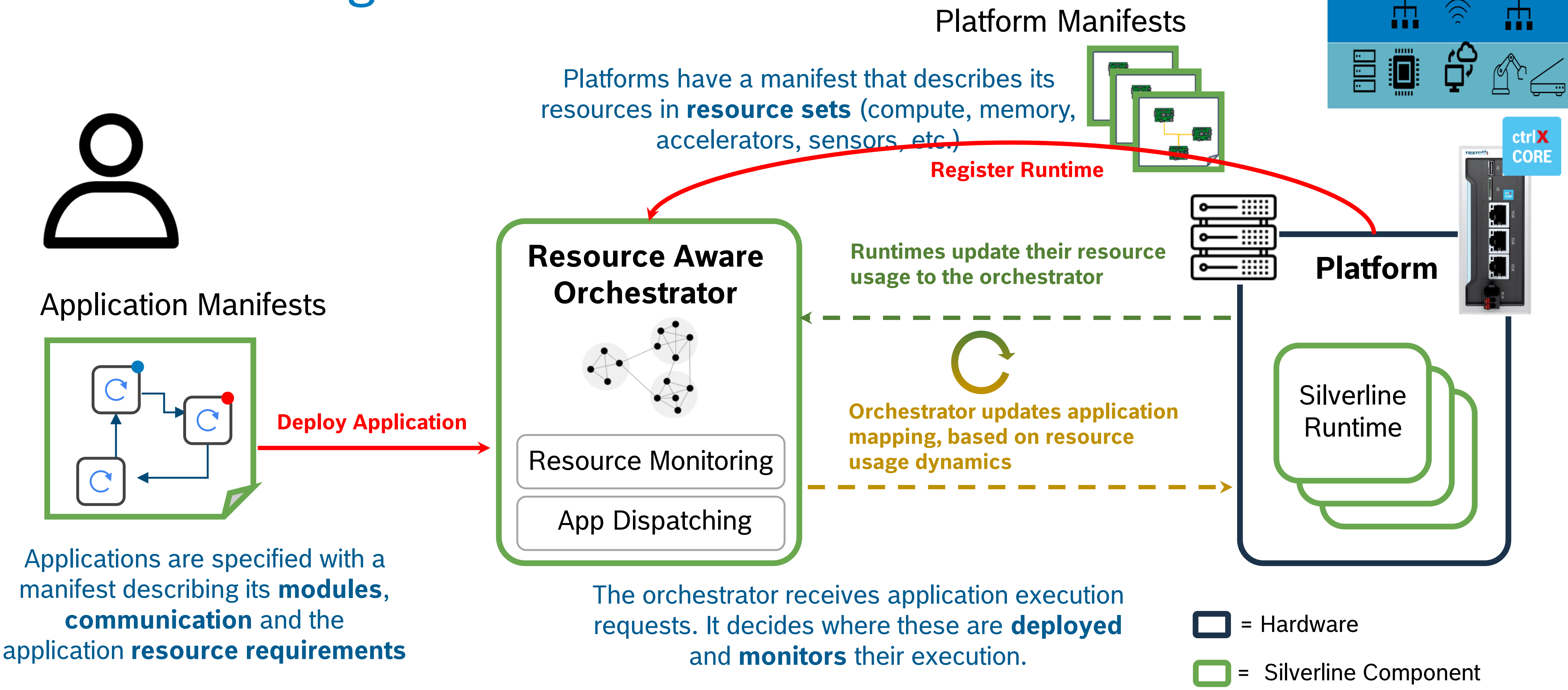


- **Resource aware orchestration**
  - Complexity & dynamic changes during operation require automated management of deployment and resource assignments
  - Deploy – Monitor – Adapt
  - Enable transparent failover, zero-downtime updates, etc.

Silverline offers IT-like flexibility with embedded qualities

Meeting the Timing Challenge in Edge Computing
Silverline: An Edge Orchestration Framework



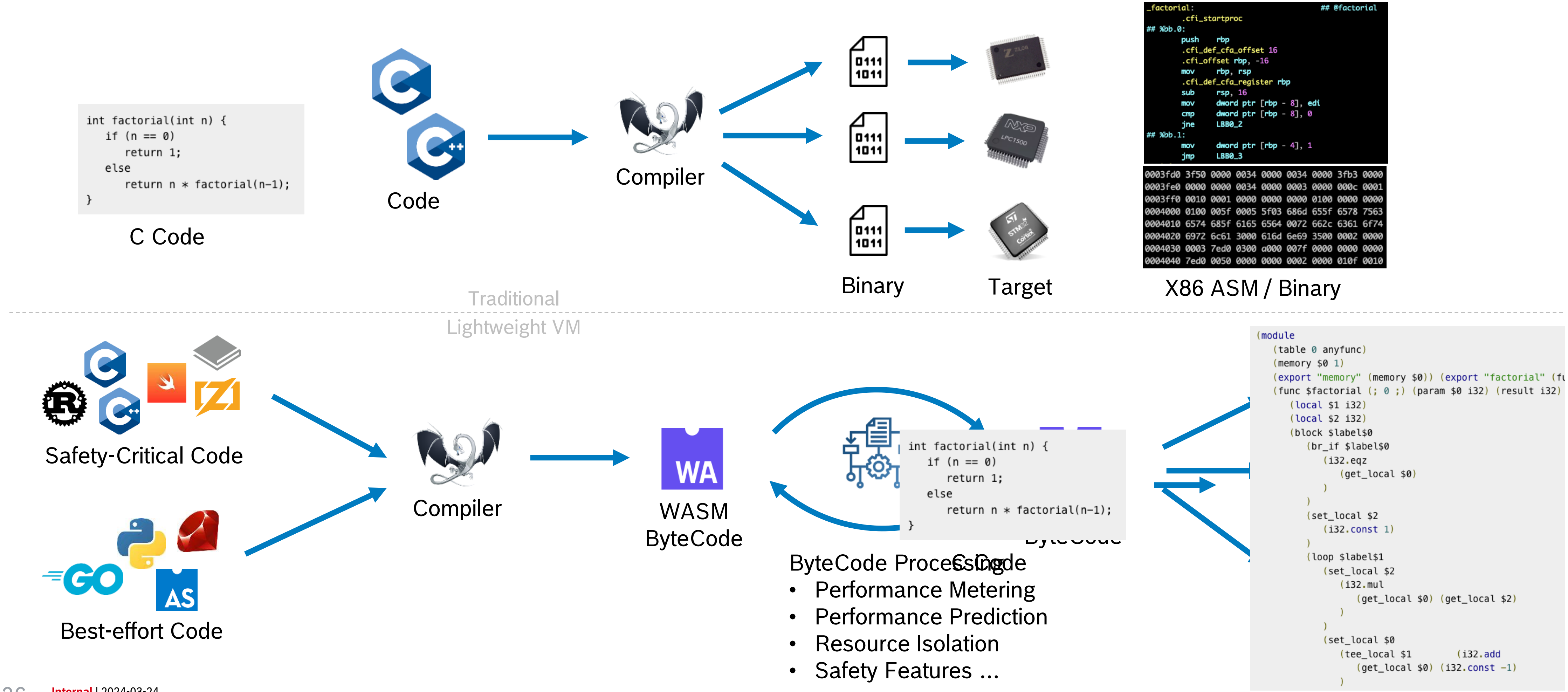


Meeting the Timing Challenge in Edge Computing
Why is Wasm a promising candidate:

- Wasm is consistently fast (no re-optimization or garbage collection)
- Wasm is safe (sandboxed, memory safe, control flow integrity, fault isolation, no access to code addresses or the call stack, capabilities-based import of external functions)
- Wasm is well-defined and deterministic (No undefined behaviors, no implementation-defined behavior, no machine-dependent behaviors, well-defined traps, no invalid calls, no illegal access to data)
- Wasm is a polyglot (compile safety-critical, real-time applications from low-level languages or even run prototype languages)
- Wasm is an Open Standard
- Wasm is formally defined and provably correct (opens door for certified compilers)

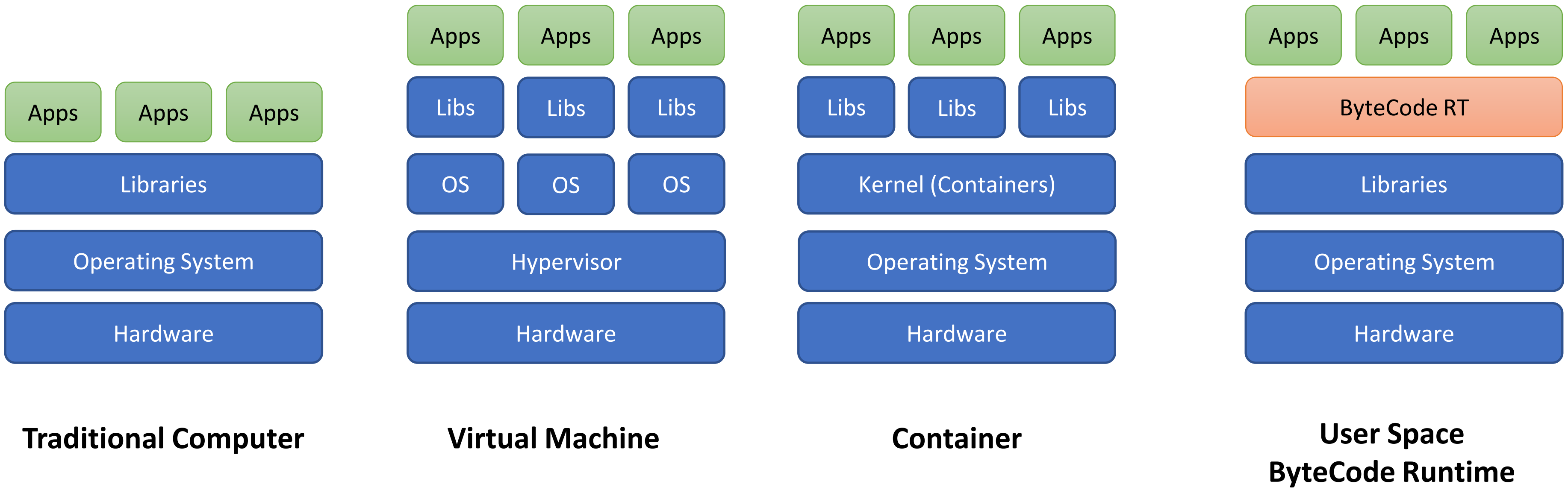
BOSCH

Meeting the Timing Challenge in Edge Computing
WebAssembly Workflow

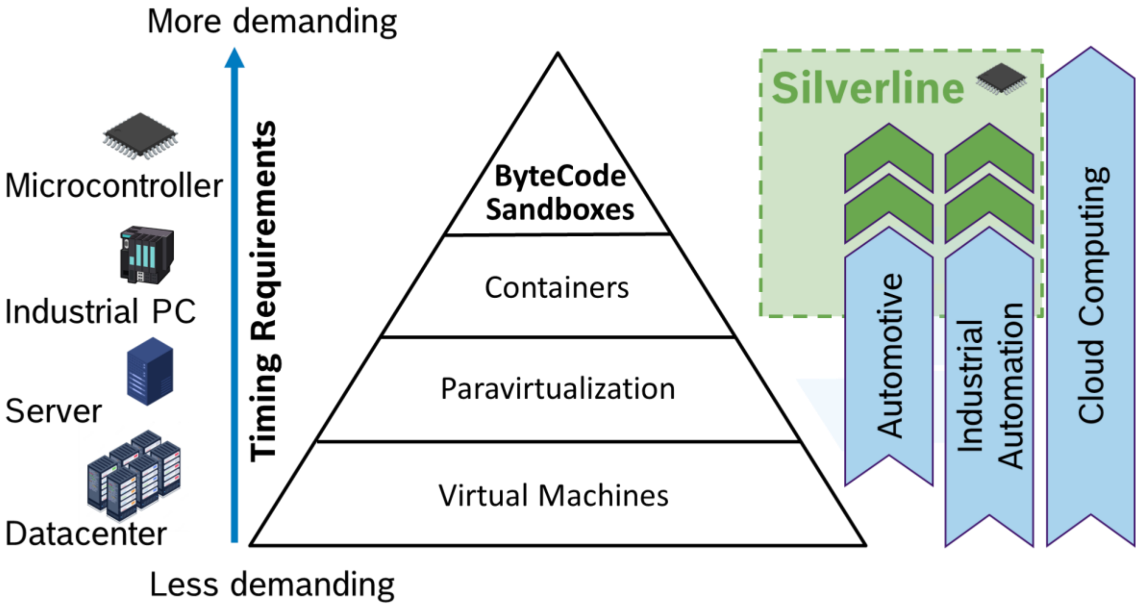


Silverline: Framework For Reliable Distributed Systems

Virtualization Trends – Architecture View

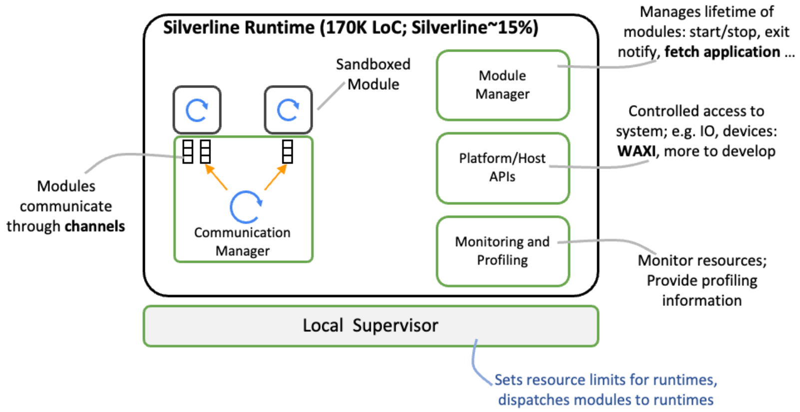


Reliable Distributed Systems Challenges
Application Portability



Virtualization is key to portability

Lots of traction for WASM



Bosch's Silverline WASM Framework
Lightweight Virtualization for CPS

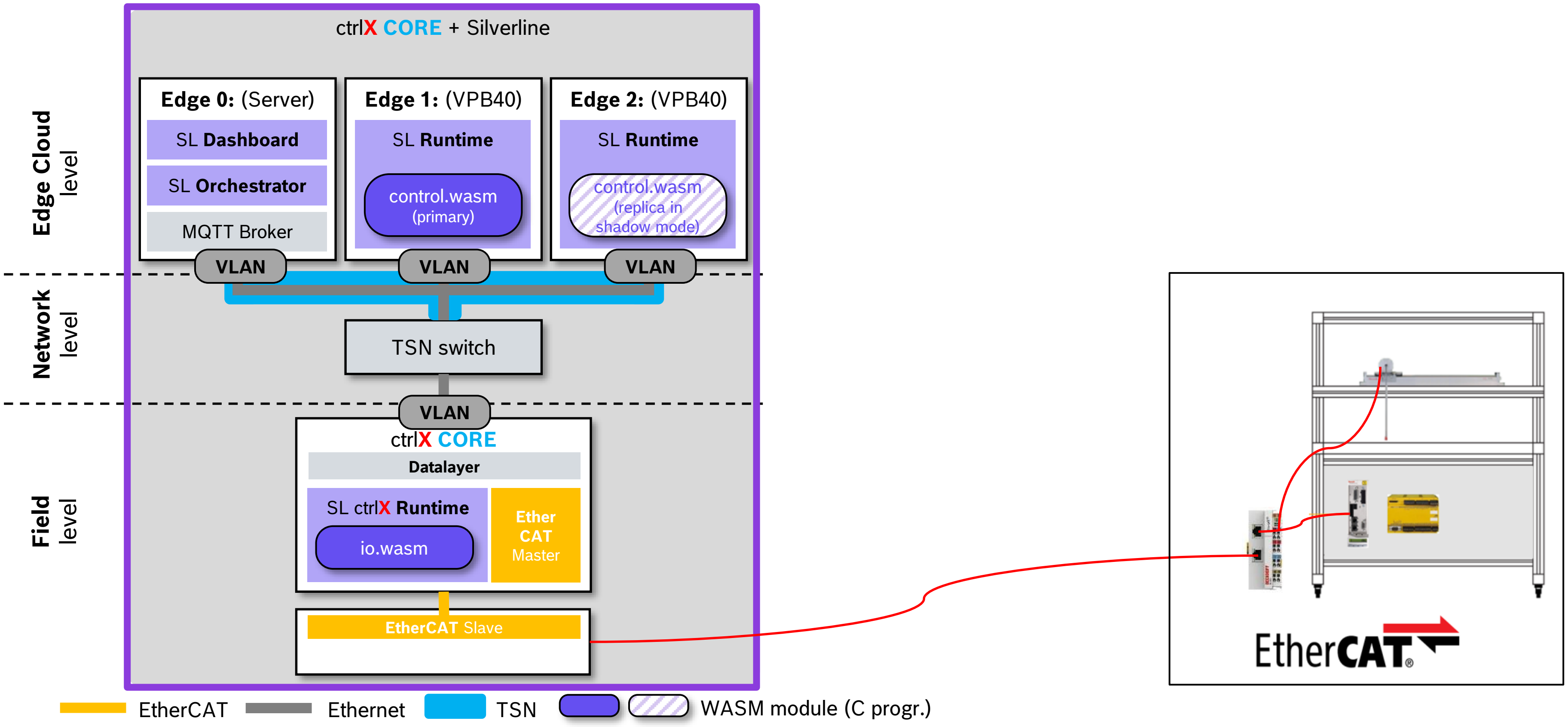
DEMO TIME

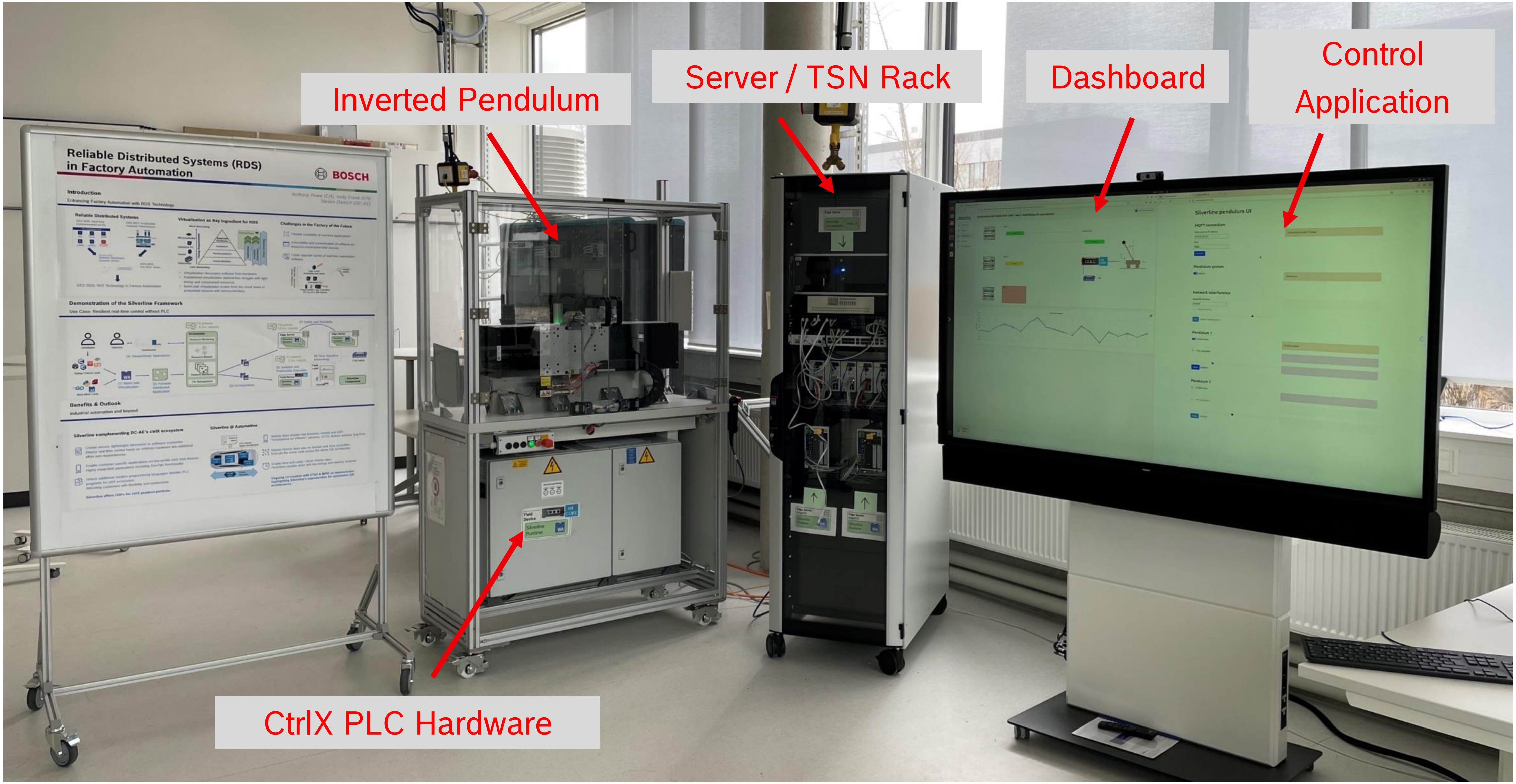
RESOURCE AWARE ORCHESTRATION

Silverline: A Framework for Reliable Distributed Systems
Demo: Edge Cloud Control Over Silverline

- Inverted Pendulum
  - Original Application: Monolithic Application Original application
    - Written in Structured Text (PLC), Needs Codesys proprietary PLC runtime
    - Inflexible deployment
  - Distributed a monolithic application into an I/O and Control Module
  - Ported application to C-> Wasm modules and introduced communication interfaces
    - MQTT in (also achieved via OPC-UA Pub Sub mechanisms in another version)
- Key goals
  - Distributed Control
  - Advanced Features
    - Transparent Failover, Zero DownTime Updates
    - Resource Aware Flexible Deployment
  - Network Isolation

Silverline in Action
Setup of Inverted Pendulum Demonstrator





Inverted Pendulum
Server / TSN Rack
Dashboard
Control Application
CtrlX PLC Hardware
Reliable Distributed Systems (RDS) in Factory Automation
BOSCH
Introduction
Silverline pendulum UI

Reliable Distributed Systems for Factory Automation

Bosch Research 2023

Silverline in Action
Timing Setup

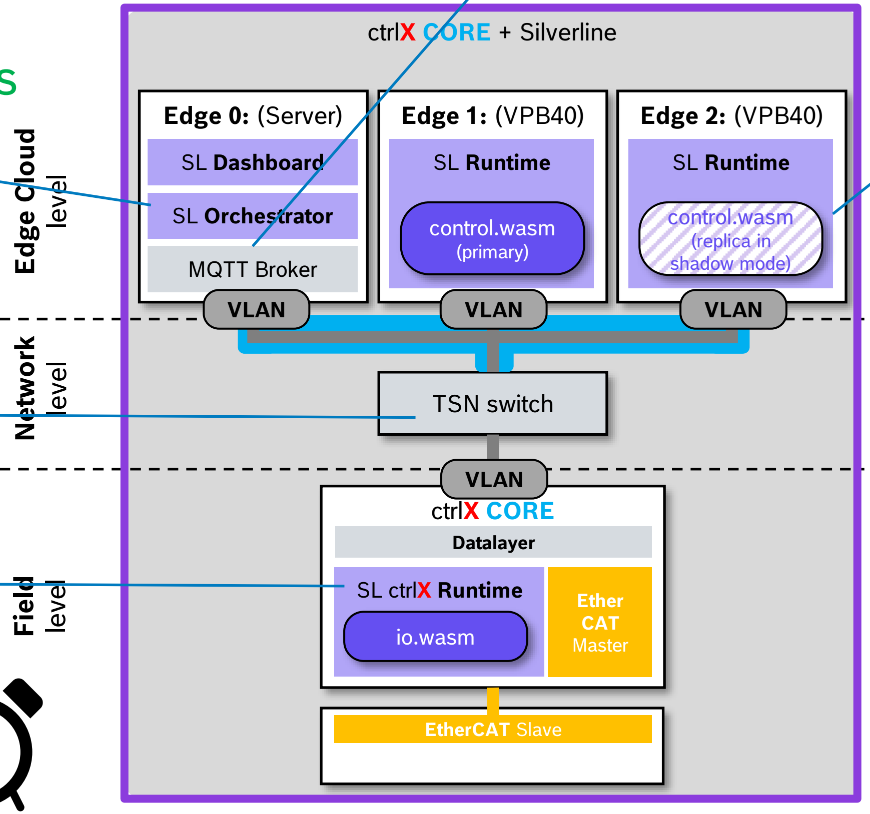
PubSub Mechanism MQTT for Tx, with real-time configuration

Orchestration sidecars for timely detection of runtime disconnects

TSN for network isolation

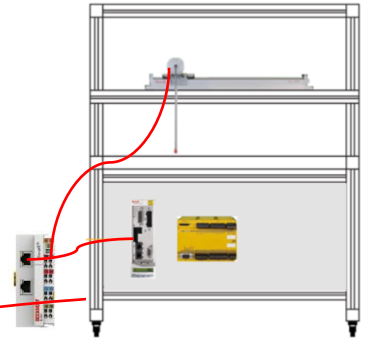
ctrlX Runtime:

Extensions for efficient datalayer access via shared memory



SilverLine Runtime: Resource reservations, Lightweight communication abstractions

Noise reduction: Core pinning, disable frequency scaling



EtherCAT

Ethercat FieldBus:

2ms periodic sensing/actuation

EtherCAT | Ethernet | TSN | WASM module (C progr.)

Internal | 2024-03-24

BOSCH

Reliable Distributed Systems

Nobody can build RDS alone – many competencies are needed

Players

SIEMENS
BOSCH

OEMs SIEMENS
rexroth
B/S/H/

Telecommunications

Hyperscalers

Assets & Competencies

Domain Knowledge

Communications

Cloud Technologies

Real-Time

Systems Integration

IT Security

Safety

Embedded Security

▶ The complexity of RDS can only be dealt with through partnerships, standardizations

BOSCH

Meeting the Timing Challenge in Edge Computing
Summary: Plethora of Challenges and Opportunities

- What are the right programming abstractions for the Edge ?
- Which virtualization mechanisms are best suited for deploying and migrating applications
  - We considered Webassembly –Byte code format –Polyglot, platform independent.
  - Not the only player in the game !
- Functional Decomposition/Deployment: When, how and where to offload applications
  - How to manage state ?
  - How to decompose legacy applications
- How to build Resilience and enable Edge Autonomy
  - Dealing with Timing Issues (Asynchronous message arrivals, etc)..
- Which mechanisms can be used to guarantee across the edge-cloud continuum
  - Resource reservations in a multi-tenant setup
  - Resource reservations on the network ?

▶ Lots of avenues for future researchers to change the game!

Meeting the Timing Challenge in Edge Computing
References

- Zuepke, A., Bastoni, A., Chen, W., Caccamo, M., & Mancuso, R. (2023). MemPol: Policing Core Memory Bandwidth from Outside of the Cores
- H. Yun, G. Yao, R. Pellizzoni, M. Caccamo and L. Sha, "MemGuard: Memory bandwidth reservation system for efficient performance isolation in multi-core platforms," *2013 IEEE 19th RTAS*
- https://docs.kernel.org/scheduler/sched-deadline.html

- *Thanks to my collaborators:*
  - Franz-Josef Grosch, Arne Hamann, Dirk Ziegenbein, Anthony Rowe, Marco Giani, Patrick Wiener, Fedor Smirnov, Behnaz Pourmohseni

Thanks!!

Discussion Time!

BOSCH