SOFTWARE-DEFINED NETWORKING IN AUTOMOTIVE

MICHAEL DOERING (BOSCH, CR/AEX1)
JENS BIERSCHENK (BOSCH, AE-BE/EKE1)

BOSCH

Software-Defined Networking in Automotive

Information Technologies' influence on Automotive Technology

2007:

- **GBit-Ethernet and Computers in cars?**
  - Totally crazy idea!
  - Technologies were proven in IT,
    but not suitable for automotive use
  - Today they are.

2018:

- **Software Defined Networking in cars?**
  - Totally crazy idea?
  - Technology is proven in IT…

BOSCH

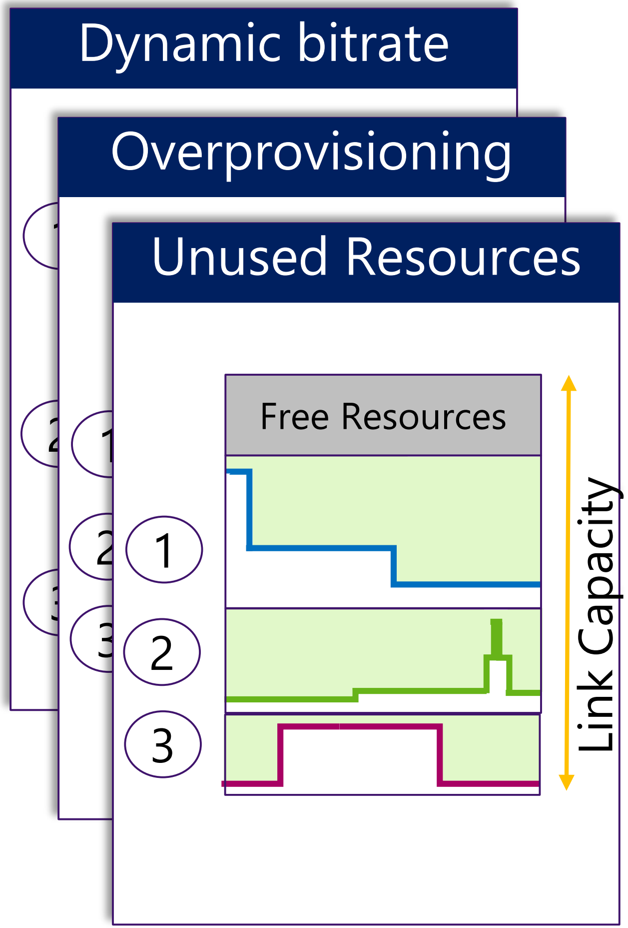
Software-Defined Networking in Automotive
Outline

- Motivation: Dynamic Communication Patterns
- Intro: Software-Defined Networking
  - Marketing Claims
  - Data Center Reality
- Overview
  - SDN Basics, Application Domains
  - Automotive Requirements
- Automotive Use Cases
- Architecture and Building Blocks

BOSCH

Introduction
Motivation

- **Today's in-vehicle communication is static, but there is a strong trend towards dynamic communication patterns**
  - Over-the-air updates, service orientation, changing or additional com. traffic
  - New use-cases cause variable or erratic traffic
    (e.g. field-data acquisition for validation, situation specific com. modes, …)
  - Changing user expectations, additional (post-production) features and 'apps'



Dilemma:
Overprovisioning is too wasteful (sum of peak traffic of *each* flow),
but uncoordinated reallocation of communication resources is also a no-go!

→ Increasing need for smart capacity utilization

Is Software-Defined Networking a solution?

Introduction
Software-Defined Networking: Marketing Claims…

Greater resource flexibility and utilization!
[Cisco]

More agile networks by abstracting the static architecture!
[Oracle]

Eliminates the complexity and static nature of traditional networks!
[Big Switch]

Make networks more flexible, programmable and scalable!
[Broadcom]

Introduction
Software-Defined Networking: Data Center Reality…

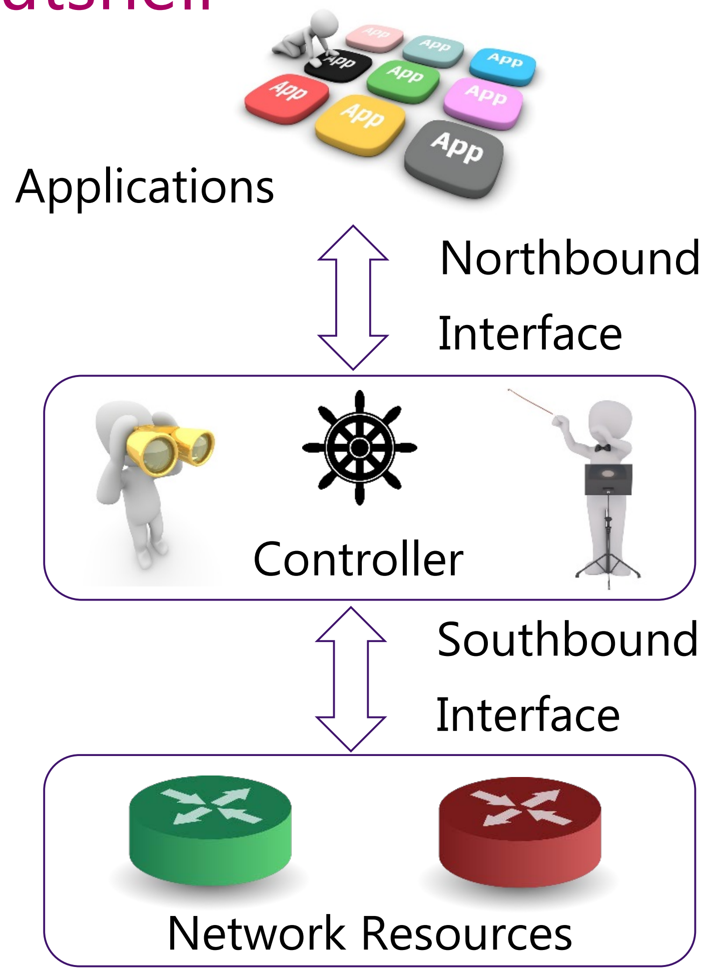
- SDN is enabling technology for cloud computing
  - Transparent overlays: Virtual networks and virtual machines (VMs) on top of physical infrastructure
  - Resource pooling and isolation: Many tenants share resources without interference
  - Elasticity and scalability: Start additional VMs on increasing load,
    SDN for load balancing and for interconnecting clients with the 'right' VM
  - Automation: No manual interaction for network planning and configuration deployment!

BOSCH

SDN OVERVIEW

SDN Overview
Software-Defined Networking: Basics in a Nutshell

- Software-Defined Networking is a system design paradigm:
  - 'Global' view on abstracted network elements, topology and state (= free & used resources!)
  - Logically centralized coordination of decentral communication devices
  - Controller is able to calculate global solution
  - Solution is translated into local configurations of devices (forwarding tables, rules, programs)
  - SDN terminology:
    - Flow: Sequence of Protocol Data Units (PDU) matching a filter (≈stream)
    - Switch: Network device that forwards and/or manipulates PDUs (≈bridge)



SDN Overview
SDN Application Domains: Use Cases and Maturity

Cloud Data Centers

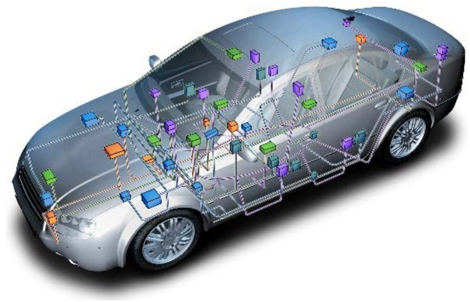
Use Cases
- Dynamic interconnection of VMs/containers
- Load balancing, availability, live migration

Maturity
- State of the art

Industrial Automation

Use Cases
- Reconfiguration of manufacturing equipment ("Industry 4.0")
- Update of TSN schedules

Maturity
- First concepts from SDN in standardization (CNC, 802.1Qcc)



In-vehicle Networks

Use Cases
- Coordinated update of communication matrix
- SW updates, QoS for SoA
- Update of TSN schedules

Maturity
- Research: studies and PoCs at OEMs and Tier1s

SDN Overview
SDN Application Domains: Properties and Requirements

Cloud Data Centers

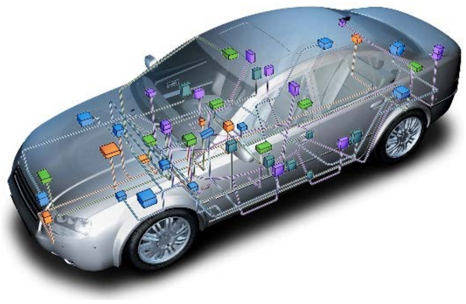
Properties
- Up to 100.000s of endpoints
- Very high dynamics (VM orchestration)

Require-ments
- Scalability, availability, isolation, security

Industrial Automation

Properties
- Up to 1000s endpoints
- Medium dynamics (mainly hardware components)

Require-ments
- Real time capabilities, safety, availability, security



In-vehicle Networks

Properties
- 10s to 100s of endpoints
- Low to high dynamics (depends on use cases)

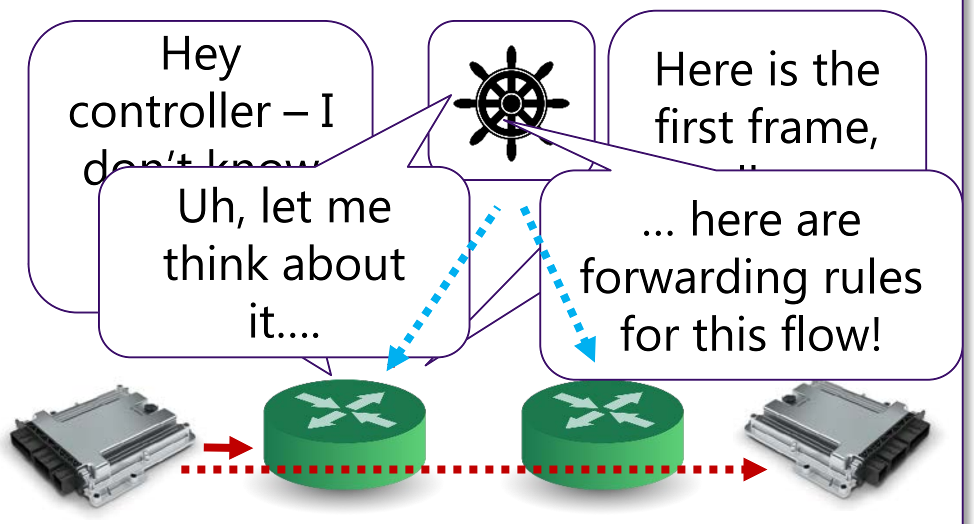
Require-ments
- Real time capabilities, safety, security, availability, cost-efficiency,…

BOSCH

SDN Overview
SDN in Automotive

- Special requirements
  - TSN integration
  - SDN must not cause increased latencies!
  - End-to-end streams across heterogeneous segments
    - E.g. different link speeds and technologies (Eth, CAN, …)
  - SDN Controller must not be single point of failure
    - Logically centralized, physically distributed
  - SDN-Switches must remain as independent as possible
    - E.g. autonomous error handling and backup strategies

Example: Reactive Forwarding in SDNs



Better: registration & persistent config.

Not all SDN modes are suitable for automotive use cases.

Some building blocks need adaptation.

AUTOMOTIVE SDN:
USE CASES

Use Cases
Two classes of Use Cases

- **Network resource management**
  - Automatic reconfiguration of devices, e.g. VLANs, Schedules/ Gate Control Lists - GCLs (→Central Network Configuration - CNC)
  - Bandwidth management for dynamic communication patterns
  - Flexible communication matrices

  →These use cases are not so far in the future (e.g. CNC)

  →Evolutionary transition possible

- **Advanced flow/stream manipulation**
  - Filtering and rule based manipulation of flows/frames
  - Flow redirection, e.g. for function shifting
  - Self-healing, e.g. for 'limp home'

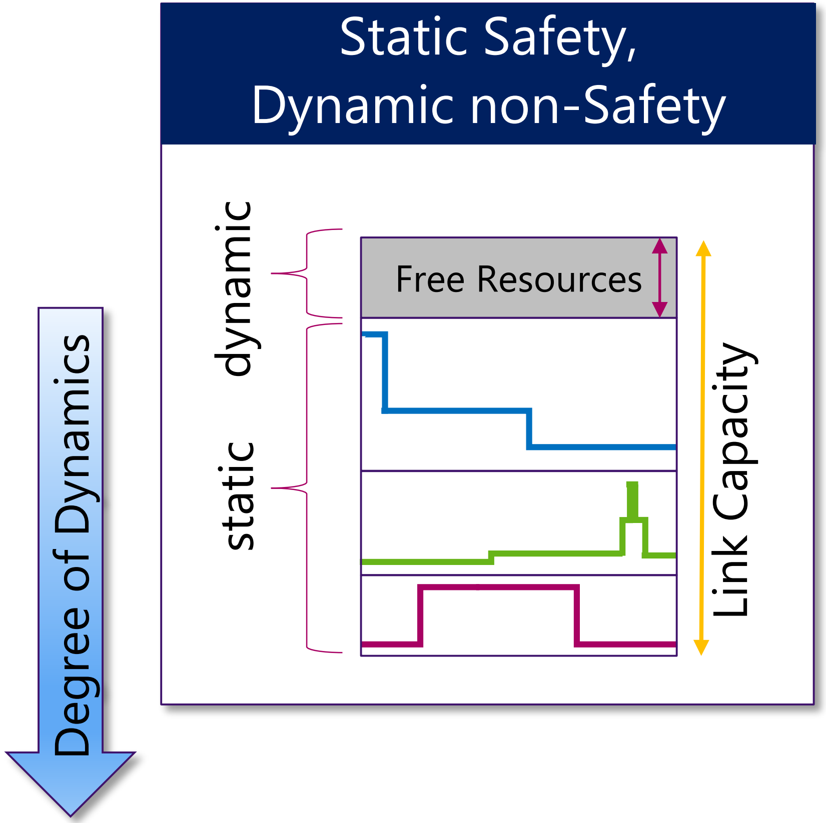
  → More challenging use cases. New safety concepts necessary.

BOSCH

Use Cases

Evolutionary Use Case Example: Resource Management / SW Update

- Over-the-air update, or deployment of additional function
  - Requires additional messages (e.g. sensor data, ESP friction)
    - Huge challenge today!
    - SOA/SOC will make this easier on upper layers
    - But no interface to lower layers (e.g. GCL configuration)
  - Several options with SDN architecture:
    - Controller only handles externally generated configuration roll-out (CNC)
    - Controller locally verifies configuration
    - Situation-aware controller triggers roll-out (in a ‘safe’ situation)
    - Controller independently switches between several fixed configurations
    - Controller locally calculates dynamic configuration within separate budget
    - Controller locally handles complete off-line reconfiguration (low dynamics)
    - Integration with SOA/SOC-middleware: full dynamics

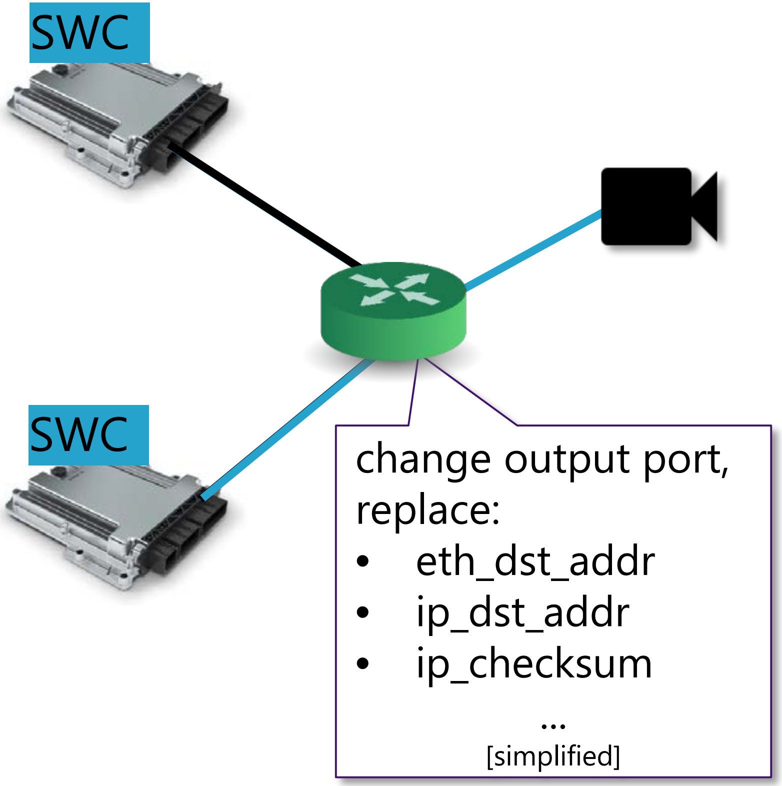


CR/AEX1-Doering | 2018-07-04
2753

Use Cases

Revolutionary Use Case Example: Advanced Flow Manipulation / Failover

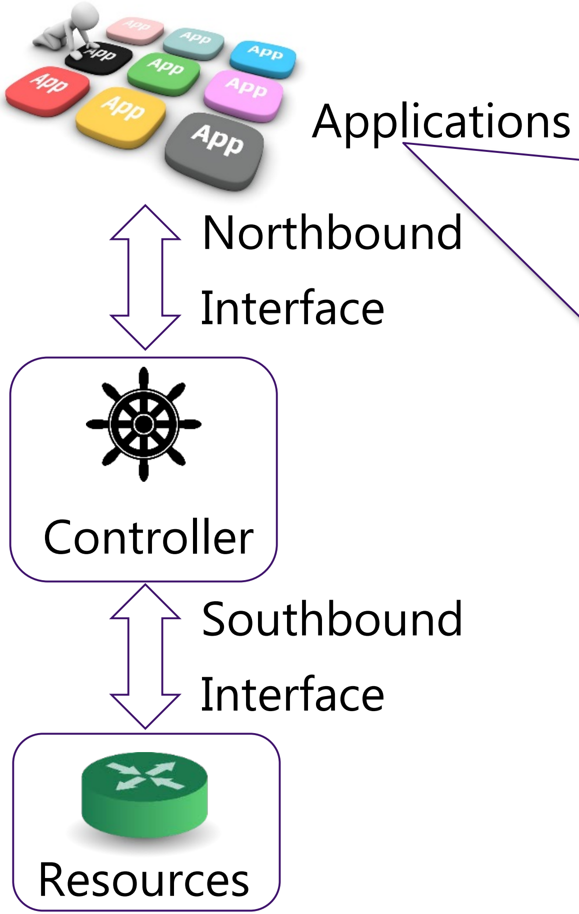
- Fast failover: switch from one processing instance to another
  - In-network redirection of frames, faster than handling at endpoints
  - Advantage over parallel/seamless redundancy:
    - much fewer network resources needed, but almost as good
    - additionally: solution to source/destination (talker/listener) redundancy
  - SDN-enabled switches
    - Filters, rules and programs for frame processing (e.g. OpenFlow, P4)
      Configured by controller, but run locally → very fast
    - Basic idea: redirect in-transit frames on failover
    - Some modifications required (e.g. address fields, checksums)
      → On-the-fly replacement of address fields
  - Paradigm-shift: frame processing and modification in switches



ARCHITECTURE AND ADAPTATION TO AUTOMOTIVE

Architecture and Adaptation

SDN in Automotive: Architectural Adaptations

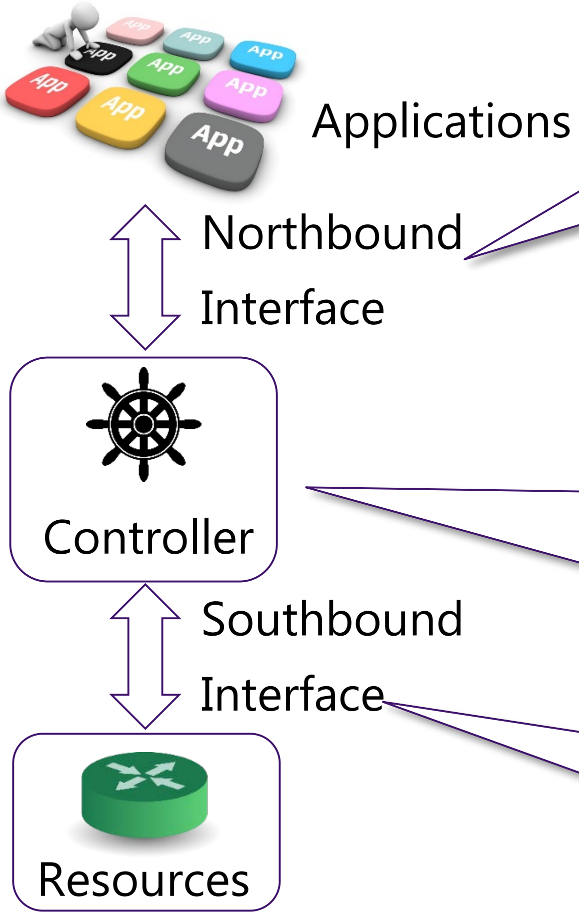


IT: Fuzzy and ambiguous definition of "Application", often used for network functions (e.g. Load Balancer)

Automotive:

- **Controller Module:** separate programs like CNC or Resource Manager that closely interact with Controller
- **Software Component:** piece of code running on an ECU/VC
  - Communication endpoints, interact with other endpoints → results in one or more **flows**
  - SWC has communication requirements that should match flows' characteristics:
    - End-to-End Latency & Jitter
    - Bitrate
    - Availability
    - ....

Architecture and Adaptation
SDN in Automotive: Architectural Adaptations



IT: Wide variety of implementations of NB-I, no consolidation yet, some standardization efforts (e.g. Open Networking Foundation)
Automotive:
- Tailor to automotive requirements (lack of established solutions)

IT: Several production-grade Controllers with failover capability available. Huge memory/CPU-footprints.
Automotive:
- Small footprint needed, but also fewer features required
- Scalability requirements bounded
- Domain specific safety and security requirements

IT: OpenFlow established standard, but only for Ethernet&IP
Automotive:
- TSN integration, CAN&FlexRay (migration path)
- Autonomous mode of operation for switches (latency, resilience)

Architecture and Adaptation
Important Interfaces 1/2: Communication Requirements

- Remember example use case “OTA Software update / dynamic com. Matrix”
  - New version causes additional traffic
  - Controller checks available capacity
  - If not enough: no reconfiguration!

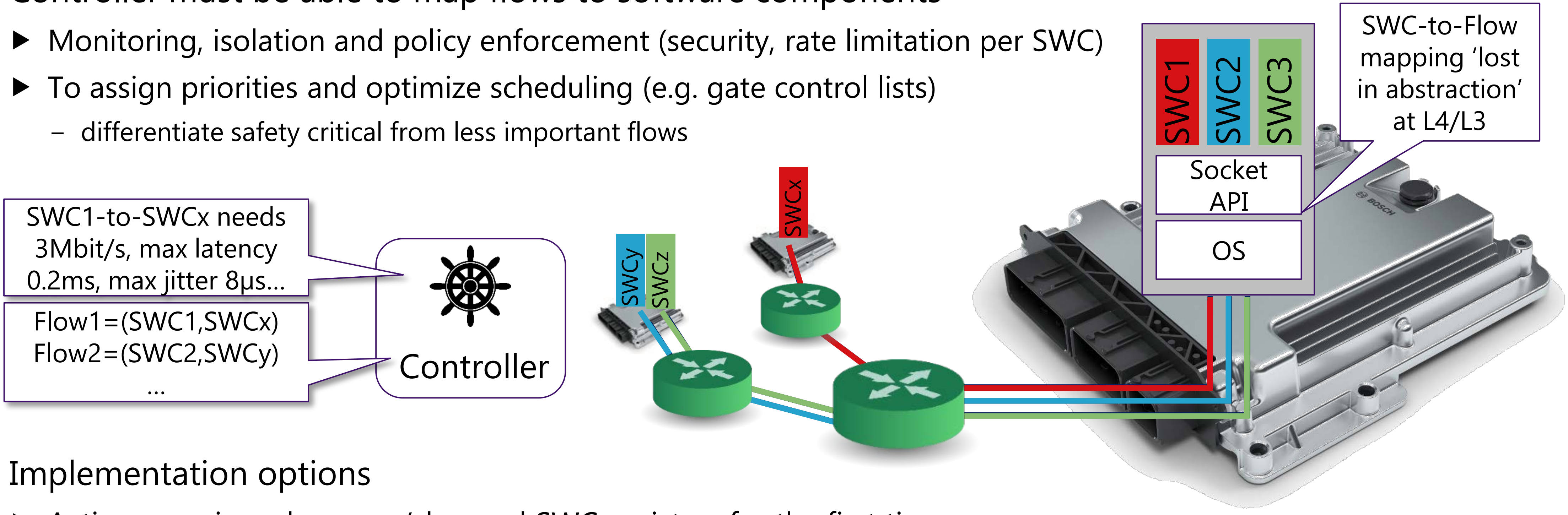
- → End-to-end communication requirements of software components must be known
  - Example: “SWC-1 needs flow to SWC-2 with <5ms latency, <1ms jitter and 500kbit/s bandwidth”
  - Could also include need for redundancy or more detailed model of communication pattern

- Implementation options
  - Direct registration of SWC at SDN controller, e.g. based on manifest or northbound interface
  - Use Adaptive AUTOSAR ara::com / QoS-API

BOSCH

Architecture and Adaptation
Important Interfaces 2/2: Flow Identification and Mapping

- Controller must be able to map flows to software components
  - Monitoring, isolation and policy enforcement (security, rate limitation per SWC)
  - To assign priorities and optimize scheduling (e.g. gate control lists)
    - differentiate safety critical from less important flows



- Implementation options
  - Active mapping when new/changed SWC registers for the first time
  - Use unique combination of known properties (e.g. src/dst addresses and transport layer ports)

Architecture and Adaptation
Building Blocks: Which are available, developing, challenging?

Available

- Architecture, Concepts, various implementations of Controllers
- Broad experience in highly available productive installations
- Ethernet integration, standardized Interfaces
- Off-the-shelf Data-Center 'SDN-Switches'

Developing

- TSN integration
  - CNC
  - SDN&TSN-Switches (Development Platforms)
- CAN integration
- Interfaces for End-to-End SWC requirements
- Proof-of-Concept Implementation
- Security in automotive SDN

Challenging

- 'Compatibility' with established safety concepts
- Mapping E2E-requirements to resources across heterogeneous technologies
- Resilience and failover mechanisms
- Migration strategies

BOSCH

SUMMARY

Software-Defined Networking in Automotive
Summary

- Strong trend towards dynamic in-vehicle communication patterns
  - Increasing need for smart capacity utilization

- Thanks to SDN, automatic network resource reconfiguration is a painless process in data centers
  - Would be a huge advancement in automotive environment
    - Imagine automated resource management and network configuration!
  - Use cases: OTA updates, SOA, self-healing, …
  - Some building blocks available, TSN integration coming up
  - Challenges: Safety concepts and migration strategies

- Automotive SDN is still some years ahead of series production
  - Just like “Ethernet in Cars” some years ago

Thanks! Questions?

{Michael.Doering2|Jens.Bierschenk}@de.bosch.com