BMW GROUP

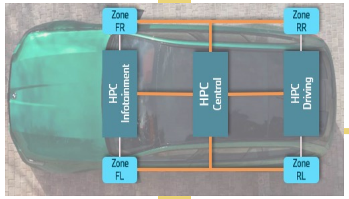
technica engineering
Member of KPIT Group

AUTOMOTIVE ETHERNET SWITCHING REBOOTED

2025 – AEC, MUNICH

STEFANY CHOURAKORN (BMW), IAGO ALVAREZ (TECHNICA), DR. LARS VÖLKER (TECHNICA)

AGENDA.



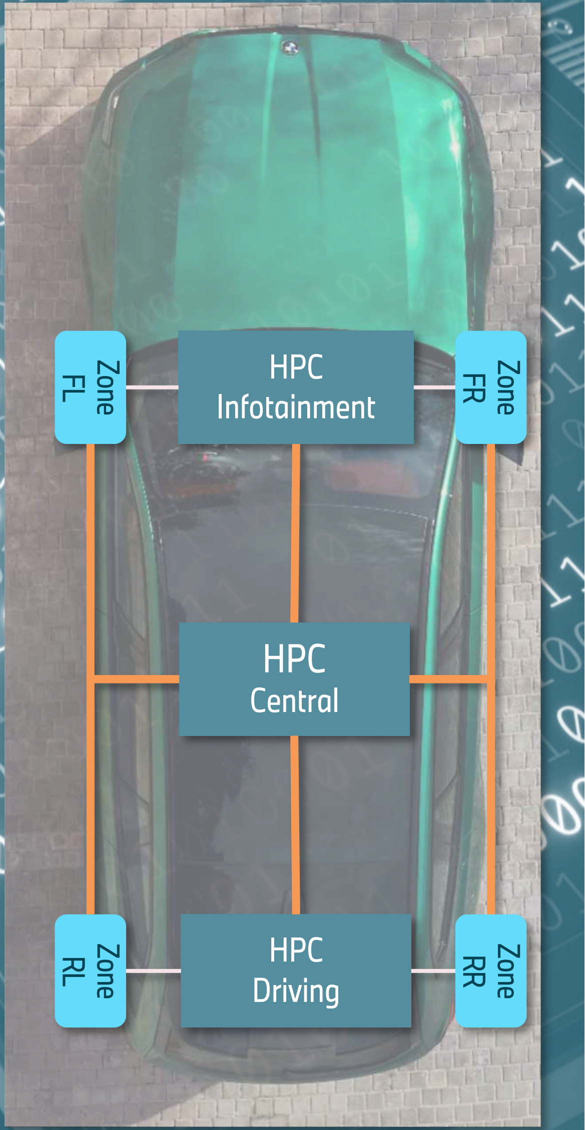
What's so complicated ?

Switch Configuration

Switch Management

Switch Software

What’s so complicated?



SWITCHES ARE KEY ELEMENTS IN OUR IN-VEHICLE NETWORK. THEIR INTEGRATION REQUIRES A LOT OF COORDINATION…

… and yet no uniformed guidelines how to manage a switch uniformly from SW perspective.

PAIN POINTS AND CHALLENGES.

Switch dependencies

HW design
Configuration depends on design.

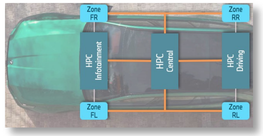
SW dependencies
Switch firmware and host drivers.

Proprietary **toolchains** and **management protocols.**

+

E/E System at OEM

Many actors
OEM, Tier 1, developers, testing house…



Multiple Systems
Linux, Autosar, …

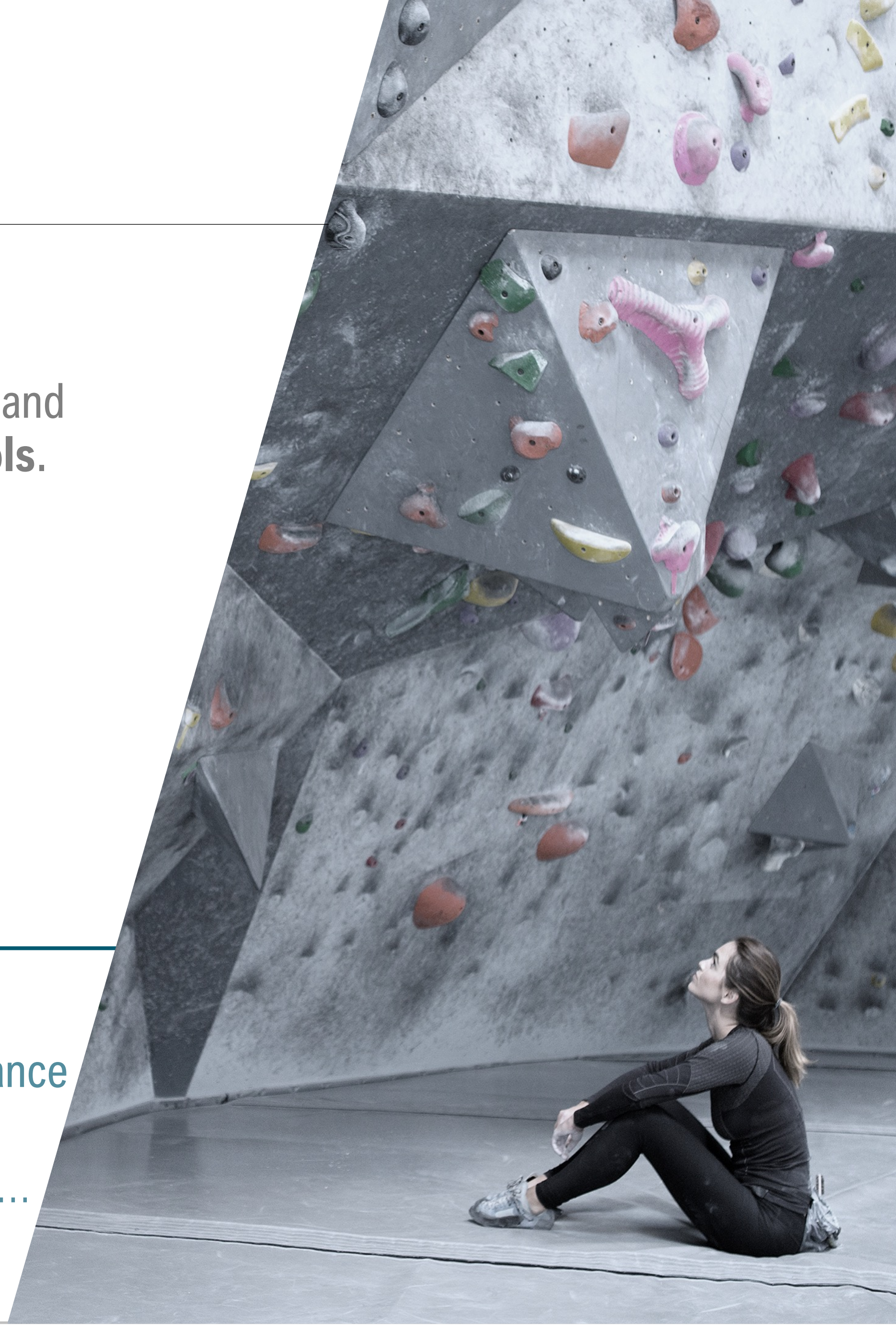
No fully **compatible solutions.**

Consequences

No systematic re-use.
Project specific.

Time consuming and costly.
Integration, bug fix, ...

Complex maintenance and compatibility.
Version, multi-vendors,…



STANDARDS: PATCHY AND LACK OF CONSENS AMONG INDUSTRIES.
SAME FUNCTIONS BUT DIFFERENT CHALLENGES.

Automotive

IEEE OPEN ALLIANCE AVNU AUTOSAR

Solutions in IT world
[vs. Automotive]

- Mechanism:
  SNMP, NETCONF, RESTCONF, CORECONF. [TBD, Prop.].
- Protocols:
  SNMP, RCP over SSH, HTTP. [SOME/IP, Prop.].
  ➢ Monitoring vs. management.
- Configuration model:
  MIB, YANG [AUTOSAR, MIB support].
- Description format:
  XML, JSON [AUTOSAR, MIB support].

Automotive

- Many standards, different working groups.
- Focus on HW and protocols.
- No generic SW definition for all systems.
- Not covering for all our needs.

Open points for IT solutions

- Adaptability for embedded solution: resources, complexity, safety, …
- Capability for extended management
- Security
- Does it solve all our problems ?

?

Why have switch vendors from other sectors still not introduced these mechanisms in automotive ?

WANTED!

System level

Expandable and sustainable solution for E/E evolution.

Simplified and **unified** workflow.

Adapted for multi-party projects and beneficial for all users (ecosystem).

Configuration

Universal configuration and description format.

Automation generation and **checks** in Continout Integration

Smooth **porting** and **migration independently from ECU platform**

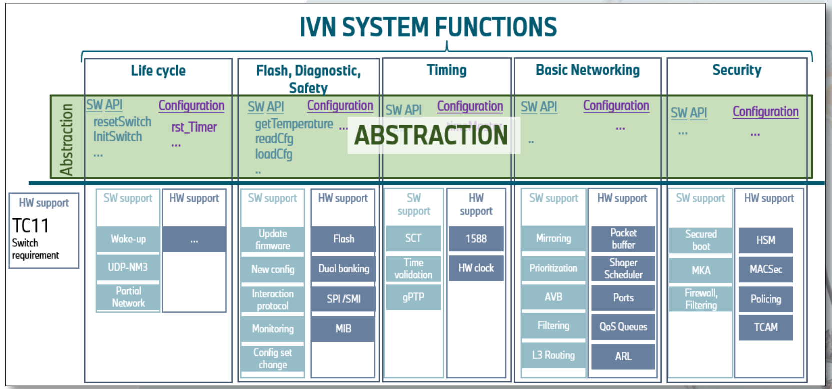
SW Interfaces

SW Reuse for synergy.

Maintenance and compatibility.

Efficient bug finding and fixing

HW abstraction in SW via **system-function** approach



- Common understanding on system-function
- Reduce **time** and **complexity** in ECU development

With the increase of IVN complexity, the need of a solution is now acute !

STANDARDIZATION!

- How can you contribute? Join TC19 today!

- OPEN TC19 got founded in 2025!
  - Software for management and configuration of Automotive Ethernet Switches.
  - More than 50 members joined in the first days alone!

- What do we want to do for Automotive Ethernet Switches?
  - Universal Configuration and APIs.
  - Management.
  - Extensible Software structure and abstractions.

SWITCH CONFIGURATION

SWITCH CONFIGURATION

CONFIGURATION CHALLENGES IN THE SDV LANDSCAPE

- ARXML is one of the core problems today!
  - ARXML is too complex and verbose – complexity introduces problems.
  - ARXML is too slow – authoring is hard, and parsing slows down processes.
  - ARXML is not compatible to today’s workflows (git, CI/CD, etc.) – due to size and complexity; managing diffs becomes nearly impossible.

- What SDV demands:
  - Modern, simple, fast, and SDV repository-compatible in a developer-friendly format.

- Others are also exploring “quick” solutions based on JSON.

SWITCH CONFIGURATION

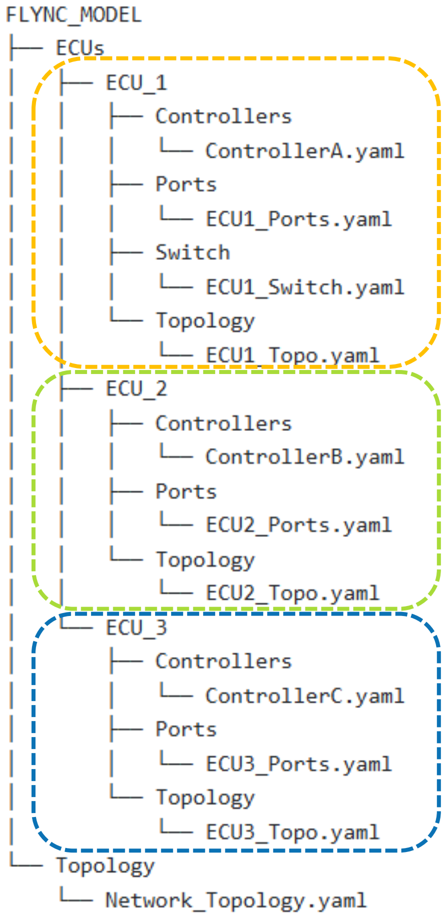
FLYNC: CONFIGURATION MODEL FOR SDV (NEW PROPOSAL)

- Flexible YAML-based Network Configuration (FLYNC) is
  - A network configuration model tailored for SDV processes.
  - A collection of YAML files.
- Key points and benefits:
  - Lightweight, human-readable and easy to use.
  - Faster parsing and execution compared to ARXML.
  - Repository-friendly and highly compatible with modern workflows.
- Main risks:
  - Migration complexity from legacy solutions to FLYNC.
  - Compatibility with existing tools, as well as standardization requirements.

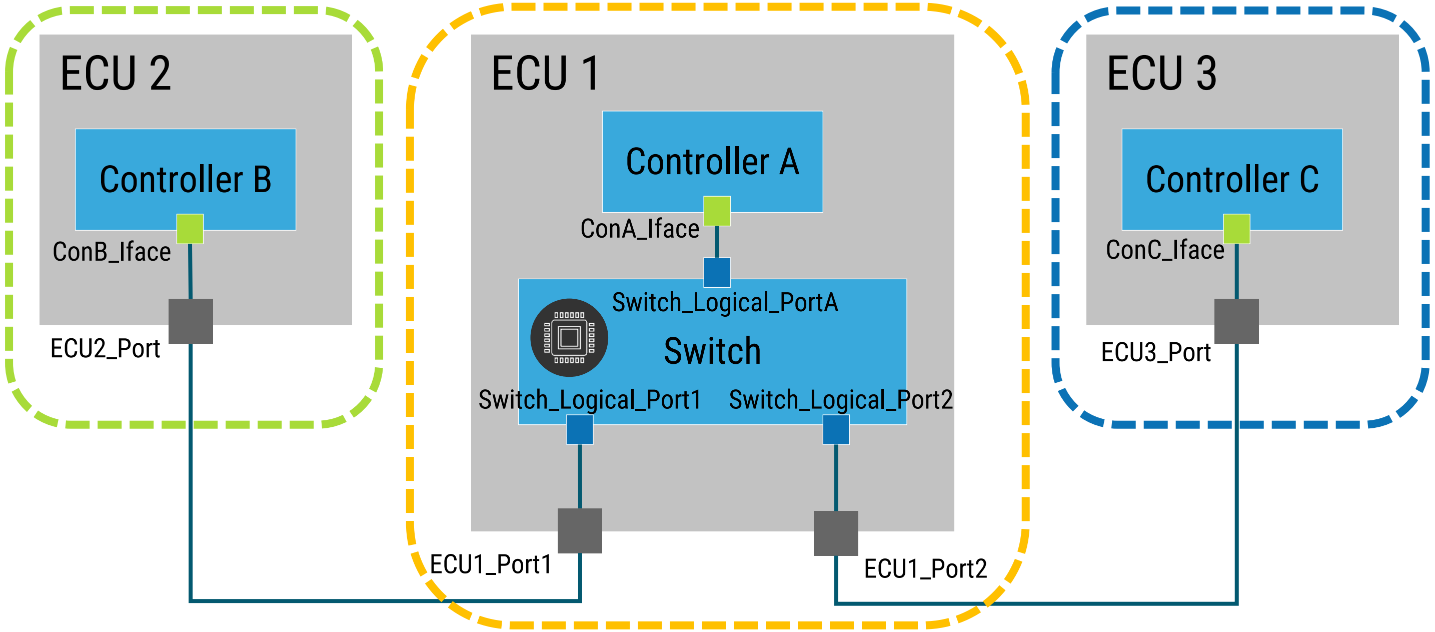
SWITCH CONFIGURATION

FLYNC IN ACTION: A VISUAL EXAMPLE

- FLYNC as enabler for SDV networks configuration:



```
FLYNC_MODEL
├── ECUs
│   ├── ECU_1
│   │   ├── Controllers
│   │   │   └── ControllerA.yaml
│   │   ├── Ports
│   │   │   └── ECU1_Ports.yaml
│   │   ├── Switch
│   │   │   └── ECU1_Switch.yaml
│   │   └── Topology
│   │       └── ECU1_Topo.yaml
│   ├── ECU_2
│   │   ├── Controllers
│   │   │   └── ControllerB.yaml
│   │   ├── Ports
│   │   │   └── ECU2_Ports.yaml
│   │   └── Topology
│   │       └── ECU2_Topo.yaml
│   └── ECU_3
│       ├── Controllers
│       │   └── ControllerC.yaml
│       ├── Ports
│       │   └── ECU3_Ports.yaml
│       └── Topology
│           └── ECU3_Topo.yaml
└── Topology
    └── Network_Topology.yaml
```

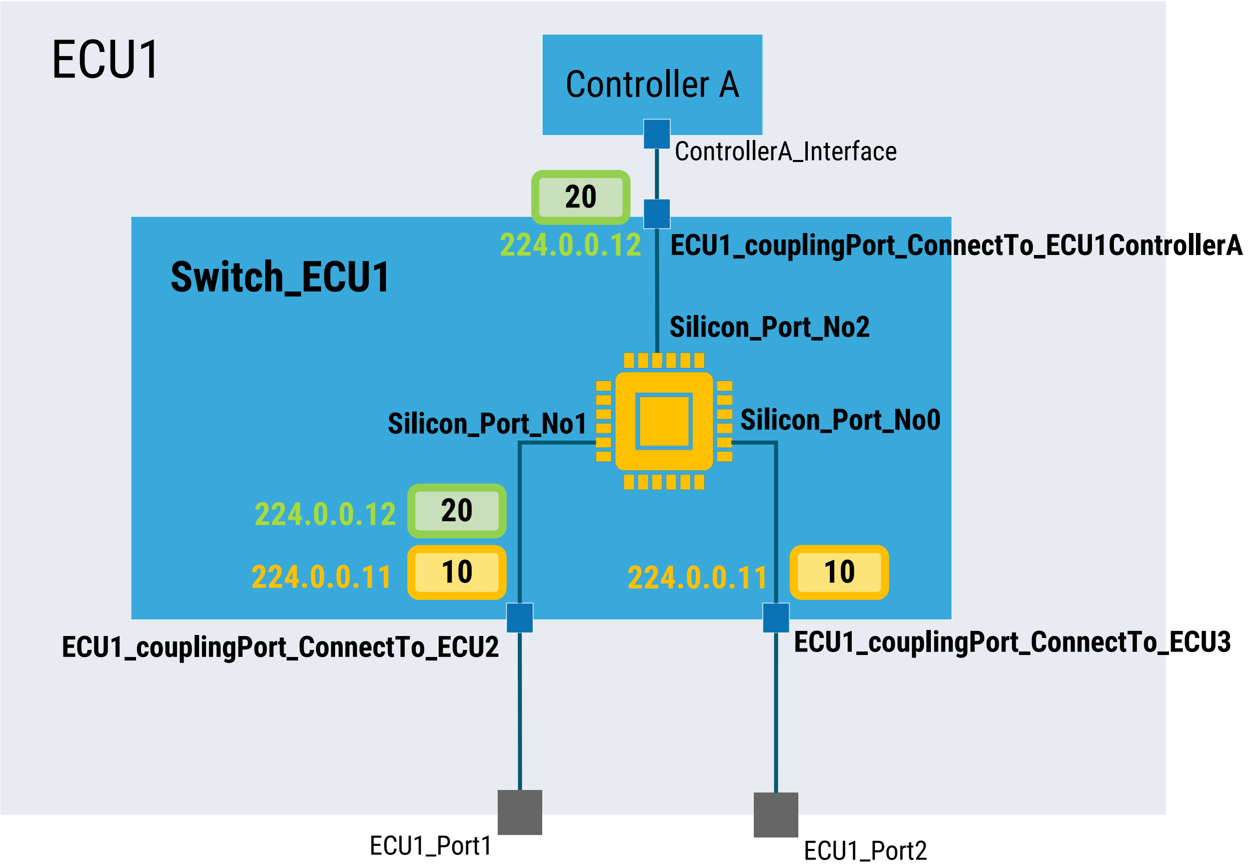


- **ECUx Topology** describes the ECU's (physical/logical) internal connections. For example: ECU_Ports to Switch_HW_Ports, Switch_HW_Ports to Controller_Ifaces.
- **Network Topology** describes the physical connections between ECUs. Ex: ECU1_Port1 to ECU2_Port.

SWITCH CONFIGURATION

FLYNC IN ACTION: A VISUAL EXAMPLE

- FLYNC for Switch Configuration



```
Switch_Name: Switch_ECU1
Ports:
  - Port_Name: ECU1_couplingPort_ConnectTo_ECU2
    Silicon_Port_No: 0
  - Port_Name: ECU1_couplingPort_ConnectTo_ECU3
    Silicon_Port_No: 1
Internal_Ports:
  - Port_Name: ECU1_couplingPort_ConnectTo_ECU1ControllerA
    Silicon_Port_No: 2
VLANs:
  - VLAN_ID: 10
    Name: VLAN_10
    Default_Priority: 7
    Ports:
      - ECU1_couplingPort_ConnectTo_ECU2
      - ECU1_couplingPort_ConnectTo_ECU3
    Multicast:
      Addresses:
      - Address: 224.0.0.11
        Ports:
          - ECU1_couplingPort_ConnectTo_ECU2
          - ECU1_couplingPort_ConnectTo_ECU3
  - VLAN_ID: 20
    Name: VLAN_20
    Default_Priority: 0
    Ports:
      - ECU1_couplingPort_ConnectTo_ECU1ControllerA
      - ECU1_couplingPort_ConnectTo_ECU2
    Multicast:
      Addresses:
      - Address: 224.0.0.12
        Ports:
          - ECU1_couplingPort_ConnectTo_ECU1ControllerA
          - ECU1_couplingPort_ConnectTo_ECU3
```

SWITCH MANAGEMENT

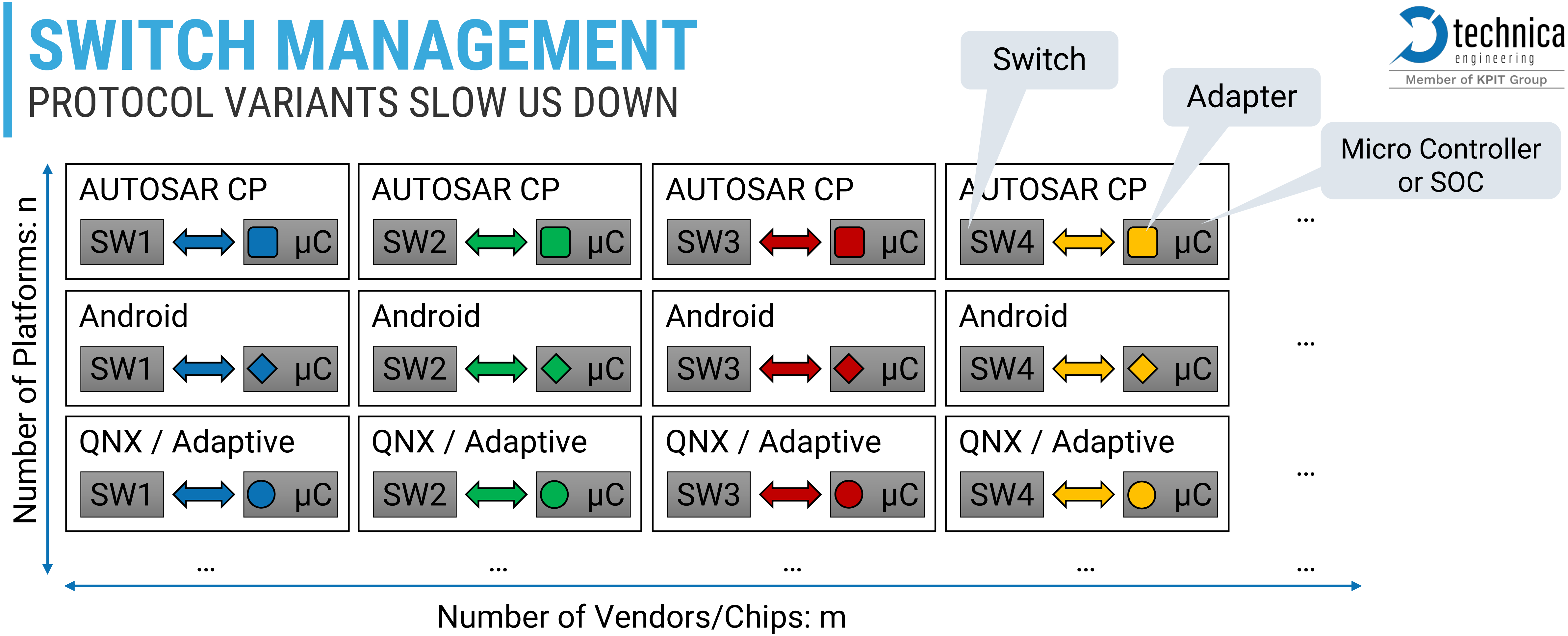
SWITCH MANAGEMENT

WHAT DO WE HAVE TO DO?

- We are talking about Automotive Switch Management for about 14+ years:
  - Who is allowed to write code for internal ARM controllers?
  - License and cost?
  - Can we reuse IT standards and do we need AUTOSAR?
- Is anybody happy about the current situation?
  - Vendor-specific SDKs on Switches and Complex Device Drivers on AUTOSAR?
  - AUTOSAR solutions that are complex, years behind, and have unclear licenses?
  - IT-based solutions that assume tons of resources and startup time does not matter?
- Is this heterogenous approach working for us?

SWITCH MANAGEMENT

PROTOCOL VARIANTS SLOW US DOWN



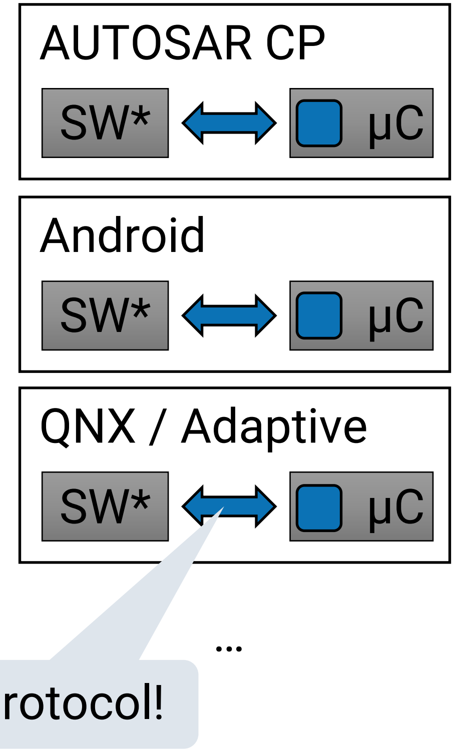
AUTOSAR CP SW1 ⟷ ■ µC	AUTOSAR CP SW2 ⟷ ■ µC	AUTOSAR CP SW3 ⟷ ■ µC	AUTOSAR CP SW4 ⟷ ■ µC	…
Android SW1 ⟷ ◆ µC	Android SW2 ⟷ ◆ µC	Android SW3 ⟷ ◆ µC	Android SW4 ⟷ ◆ µC	…
QNX / Adaptive SW1 ⟷ ● µC	QNX / Adaptive SW2 ⟷ ● µC	QNX / Adaptive SW3 ⟷ ● µC	QNX / Adaptive SW4 ⟷ ● µC	…
…	…	…	…	…

- A different adapter per vendor and operating system: m + n*m
  - Also, differences between semiconductors of a vendor are possible.
- We need to reduce this complexity to speed up development!

SWITCH MANAGEMENT

PROPOSAL FOR STANDARDIZED SOLUTION

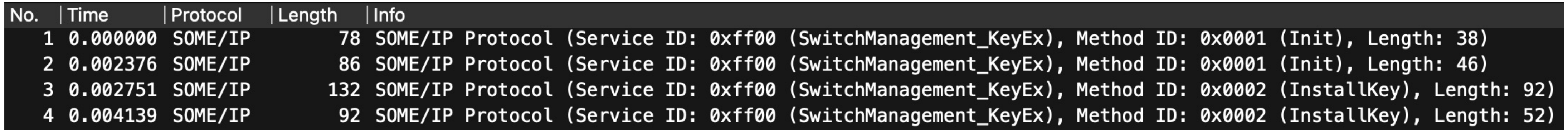
- Current standards for Network Management (in the IT sense) are not viable.
  - Too much overhead, not solving the correct problem, not fully supporting Automotive.
- We need a platform-agnostic protocol to transport our management messages.
  - Reducing required protocol implementations from **m+n*m** to **n+m**.
- Switch Management also includes installing keys into Switches and activating MACsec
  - Switch Management communication needs to be protected.
- Proposal:
  - Use a lean control protocol supported on all automotive platforms: e.g., SOME/IP.
  - Standardize high-level interfaces as services.
  - Create standardized and/or open-source adapters.
  - Built-in security solution.



SWITCH MANAGEMENT

SERVICE-BASED SWITCH MANAGEMENT

- Proposal:
  - Step 1: Establish session key.



```
No.   |Time      |Protocol  |Length  |Info
    1 0.000000  SOME/IP         78  SOME/IP Protocol (Service ID: 0xff00 (SwitchManagement_KeyEx), Method ID: 0x0001 (Init), Length: 38)
    2 0.002376  SOME/IP         86  SOME/IP Protocol (Service ID: 0xff00 (SwitchManagement_KeyEx), Method ID: 0x0001 (Init), Length: 46)
    3 0.002751  SOME/IP        132  SOME/IP Protocol (Service ID: 0xff00 (SwitchManagement_KeyEx), Method ID: 0x0002 (InstallKey), Length: 92)
    4 0.004139  SOME/IP         92  SOME/IP Protocol (Service ID: 0xff00 (SwitchManagement_KeyEx), Method ID: 0x0002 (InstallKey), Length: 52)
```

  - Step 2: Allow secure communication via SOME/IP – think SecOC.
    - MACsec: Install CAKs, activate/deactivate CAs, manage MACsec, and others.
    - Ethernet: Counters, SQI, Cable Diagnostics, Link State, CRCs, Addresses, etc.
    - gPTP + Safety: messages received.
    - Others: e.g., TCAM filters.

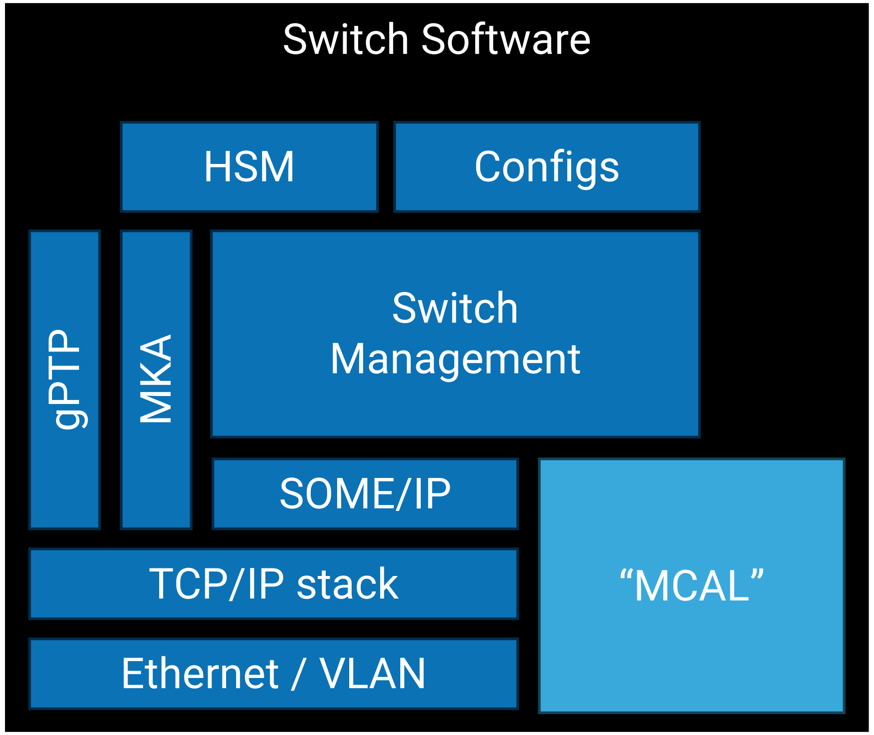
SWITCH SOFTWARE

SWITCH SOFTWARE

OVERVIEW

- What do we want to achieve?
  - Shift left development and validation.
  - Minimize the need for chip vendor specific code.
  - Simple and small software stack (SDV ready).
  - Code reuse via open-source.
  - Open standard and open license.

- Existing open-source building blocks are considered!
  - e.g., FreeRTOS, OpenBSW, lwIP



Simplified Architecture

SUMMARY

SUMMARY

- Switches are key elements in the E/E Architecture.
- Current approaches lead to multiple pain points.
- It is essential to find a solution right now as systems getting too complex.

- We propose that an open, SDV-capable solutions with:
  - An open and modern configuration format, like FLYNC
  - An efficient and effective Switch Management
  - An open, extendable software stack
  - An open and living eco system

- Do you want to join our journey?

THANK YOU FOR YOUR ATTENTION