THE ROAD TO SDV

HOW ARCHITECTURES LEAD THE WAY

#OneStepAhead

THE ROAD TO SDV

TABLE OF CONTENT

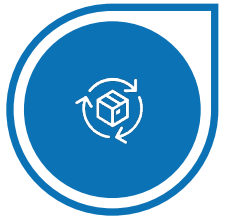
#1 | THE ROAD TO SDV
INTRODUCTION

THE ROAD TO SDV

WHAT IS THE GOAL?

- Adding features to vehicles via Software after production
  - Keeping products “fresh”
  - Fixing products without classical recalls
  - Potentially selling features to customers later
  - Important: Allowing to define the product after hardware is chosen

- Focus on Software
  - Fast and effective “Software Factories”



THE ROAD TO SDV

WHERE DID WE START FROM?

- In the last decades, a lot was pushed to the supply chain
  - Hardware design by Tier-1s, Software stacks by Tier-2s, etc.
  - Commodities improved pricing and quality
  - External partners could leverage on common solutions

- Typical process:
  - OEMs hand-over vast number of specifications
  - ARXML used to “configure” essential parameters
  - OEMs integrate vehicle

- Vehicles became cost-driven hardware software co-designs
  - New features often required new hardware

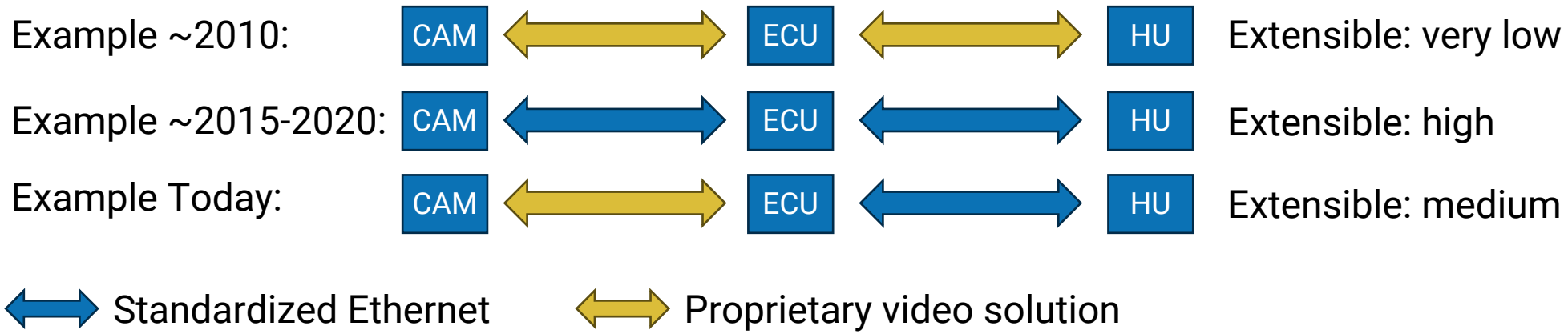
#2| THE ROAD TO SDV
COMMON MISTAKES

THE ROAD TO SDV

COMMON MISTAKE: COST OPTIMIZATION FIRST

- E/E Architecture often too cost-optimized
  - Enough CPU cycles, RAM, Bandwidth left?
  - Topology flexible enough?

- Proprietary solutions vs. future proofing
  - Which makes it easy to add more features later?



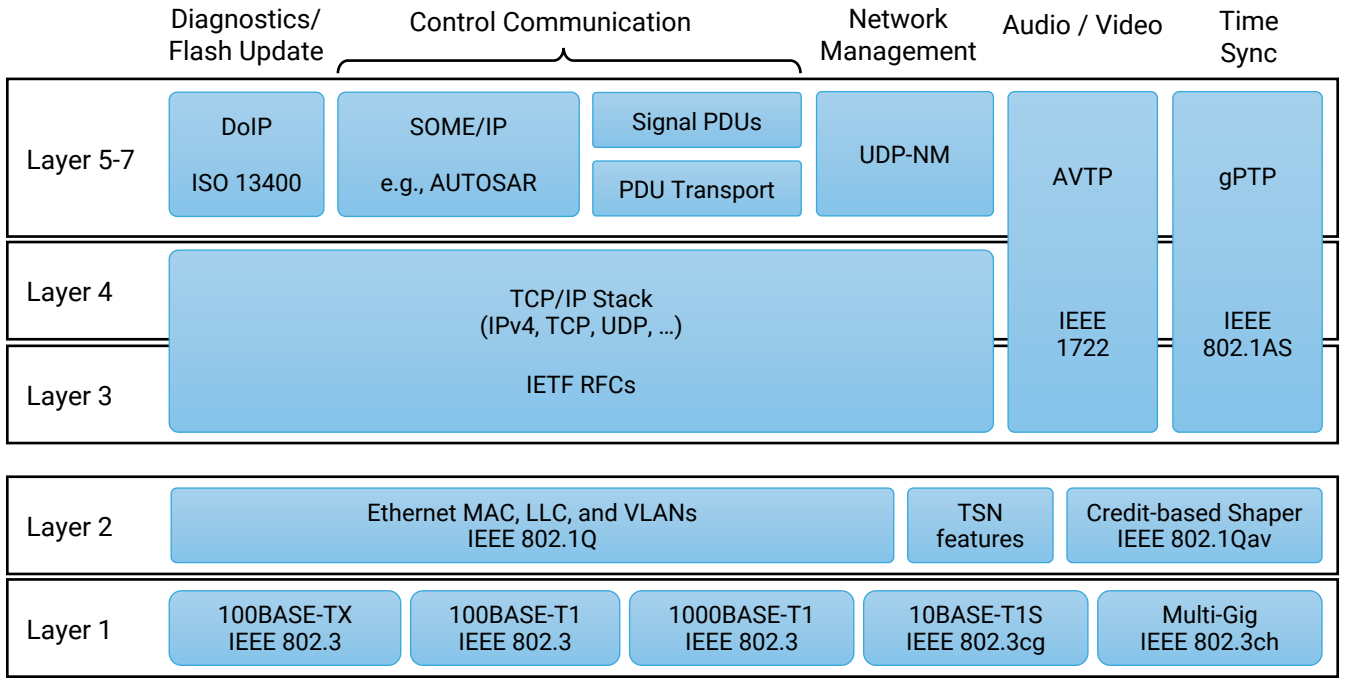
THE ROAD TO SDV

COMMON MISTAKE: COMPLEXITY EXPLOSION

- Ethernet Protocol Stack

- Red flags:
  - Proprietary extensions
  - Proprietary optimizations
  - Proprietary protocols
  - Complex protocol conversion
  - Domain-specific solutions
  - Diversion from state-of-the-art

- Compare and benchmark!

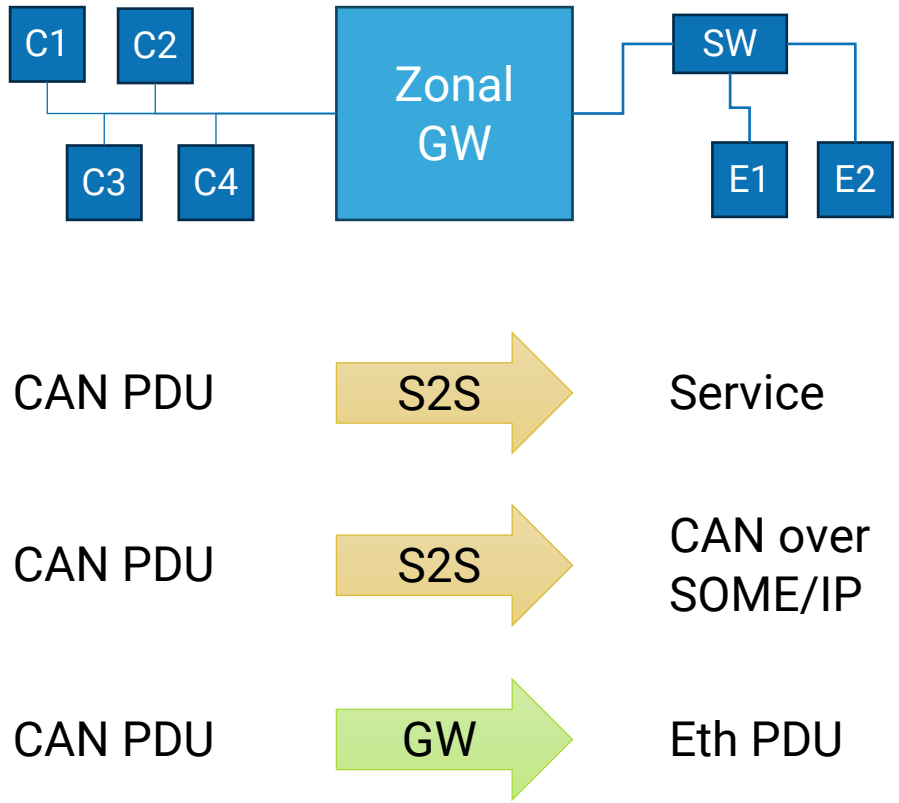


Technica Example Protocol Stack for Ethernet-based Communication
(Simplified, without Security and other aspects)

THE ROAD TO SDV

COMMON MISTAKE: LIMITED SCALABILITY

- Legacy Communication to Ethernet
  - Common option: Signal2Service (S2S)
- Issues:
  - Limited Scalability (high overhead in GW)
  - Breaks E2E Safety, breaks Security
  - Resampling even worse
  - Usage prediction challenging
- Better:
  - Use simple PDU Gatewaying only

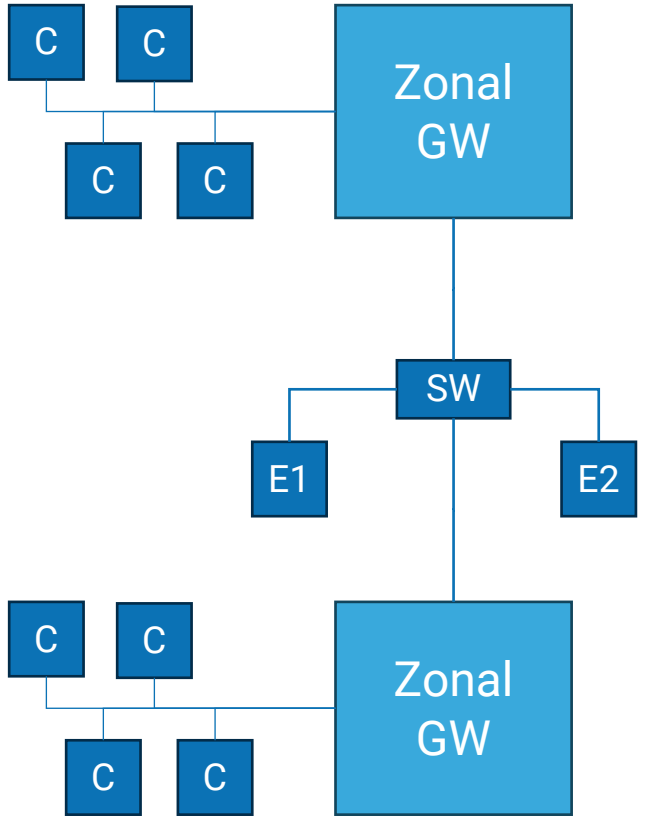


THE ROAD TO SDV

COMMON MISTAKE: SECURITY TOO LATE

- Traditional Security Process (reactive):
  - Define Use Cases
  - Determine what to protect
  - Implement Security mechanisms

- Red flags:
  - Proprietary Security Solutions
  - Many different crypto algorithms/protocols
  - Multiple similar solutions (e.g., key setup)
  - More application-based than generic solutions

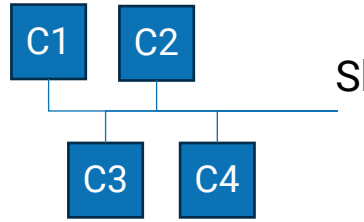


#3 | THE ROAD TO SDV
ESSENTIAL TECHNOLOGY & APPROACH

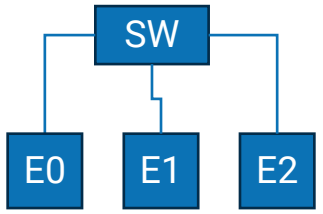
ESSENTIAL TECHNOLOGY & APPROACH

ETHERNET FOR FLEXIBLE PLATFORMS

- Automotive Ethernet allows for extensible platforms
  - Allows for high bandwidth (e.g., 100 and 1000Mbit/s)
- Switches operate in line-speed
  - Data in Ethernet Networks can be fully handled by hardware
  - Scalability limited by line-speeds and receiver capacity
- Additional benefits:
  - Virtualization with VLANs allow separation of domains, if needed
  - Switched Ethernet provides best scalability of all alternatives



Share Medium
(10Mbit/s)

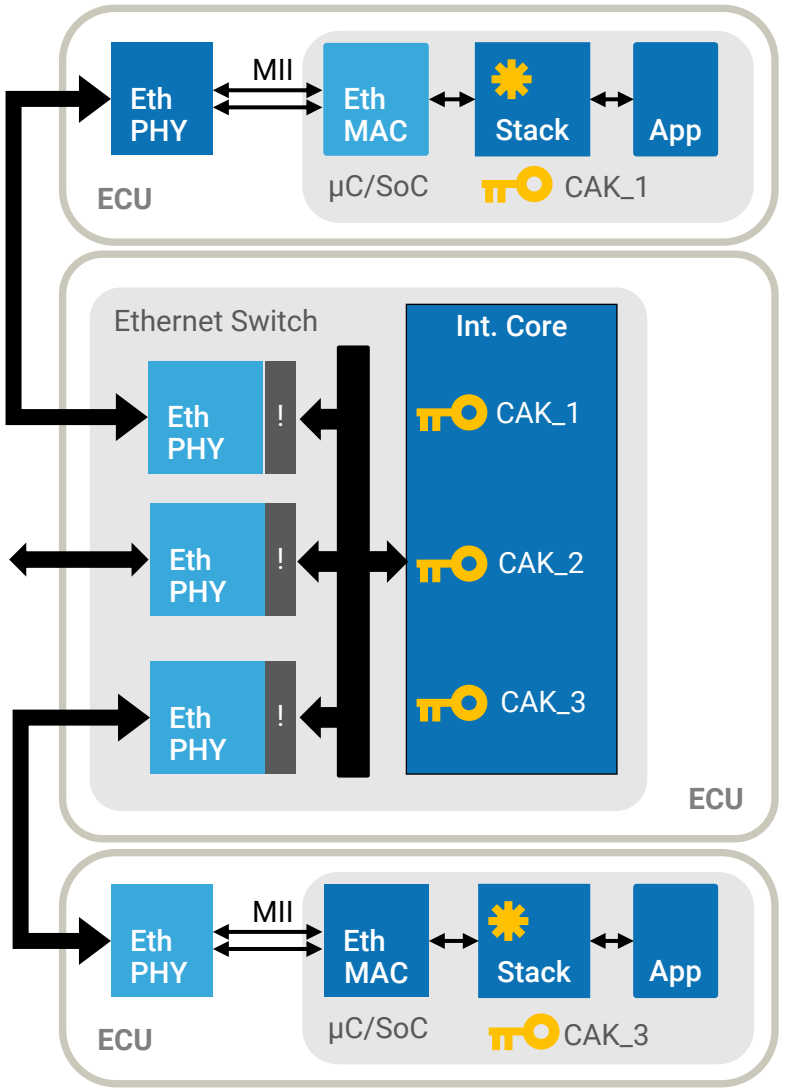


Switched Ethernet
100 Mbit/s,
1000 Mbit/s,
2500 Mbit/s,
5000 Mbit/s,
10000 Mbit/s,
...

ESSENTIAL TECHNOLOGY & APPROACH

MACSEC AS PLATFORM SECURITY

- Security needs to be built into the platform
- MACsec is the best solution for protecting Ethernet
  - Hardware acceleration to achieve full through-put
  - Only hop-by-hop MACsec can protect everything
  - Automotive MACsec can start in about 10ms
- Combine with at least VLANs, filters, and ACLs
- Platform security must come very early!



MACsec presentations at https://automotive-macsec.com/papers.shtml

#4| THE ROAD TO SDV
HOW TO SPEED UP DEVELOPMENT?

HOW TO SPEED UP DEVELOPMENT?

AUTOSAR APPROACH

- AUTOSAR
  - Created Architectures, Formats, and Terminology
  - Very good for OEM delegating work to Tier-1s

- AUTOSAR Config Format ARXML
  - XML-based, huge → very hard to debug or ensure quality
  - Generation and Integration often not fully automated and slow
  - Every version different, vendor and OEM differences
  - Not repo compatible → hard to support modern processes
  - Stack view instead of vehicle view

- For SDV, improvements needed!

Schema 23-11
→ 9.8 MB

Single files often multiple 100 MB!

HOW TO SPEED UP DEVELOPMENT?

TESTING AND VALIDATION

technica engineering
Member of KPIT Group

- Automated testing of software units becoming popular

- But: ECU testing and Integration Testing often too manual
  - ECU software needs to automatically be tested via CI/CD
  - Easily 50 – 500k test cases needed for ECU validation

- Do not forget to use your data:
  - Automated analysis of traces from vehicles and integration setups
  - Record all communication with meta data and high-res timestamps
  - Analysis: Link load, load for different Traffic Aggregates, etc.

#5 | THE ROAD TO SDV
SUMMARY

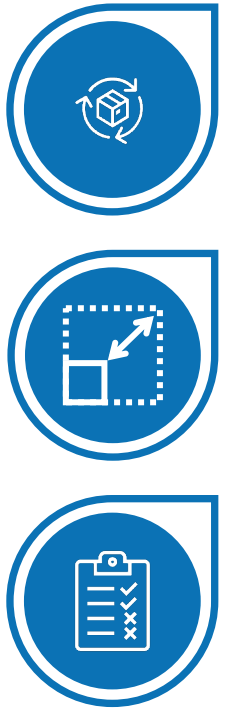
THE ROAD TO SDV

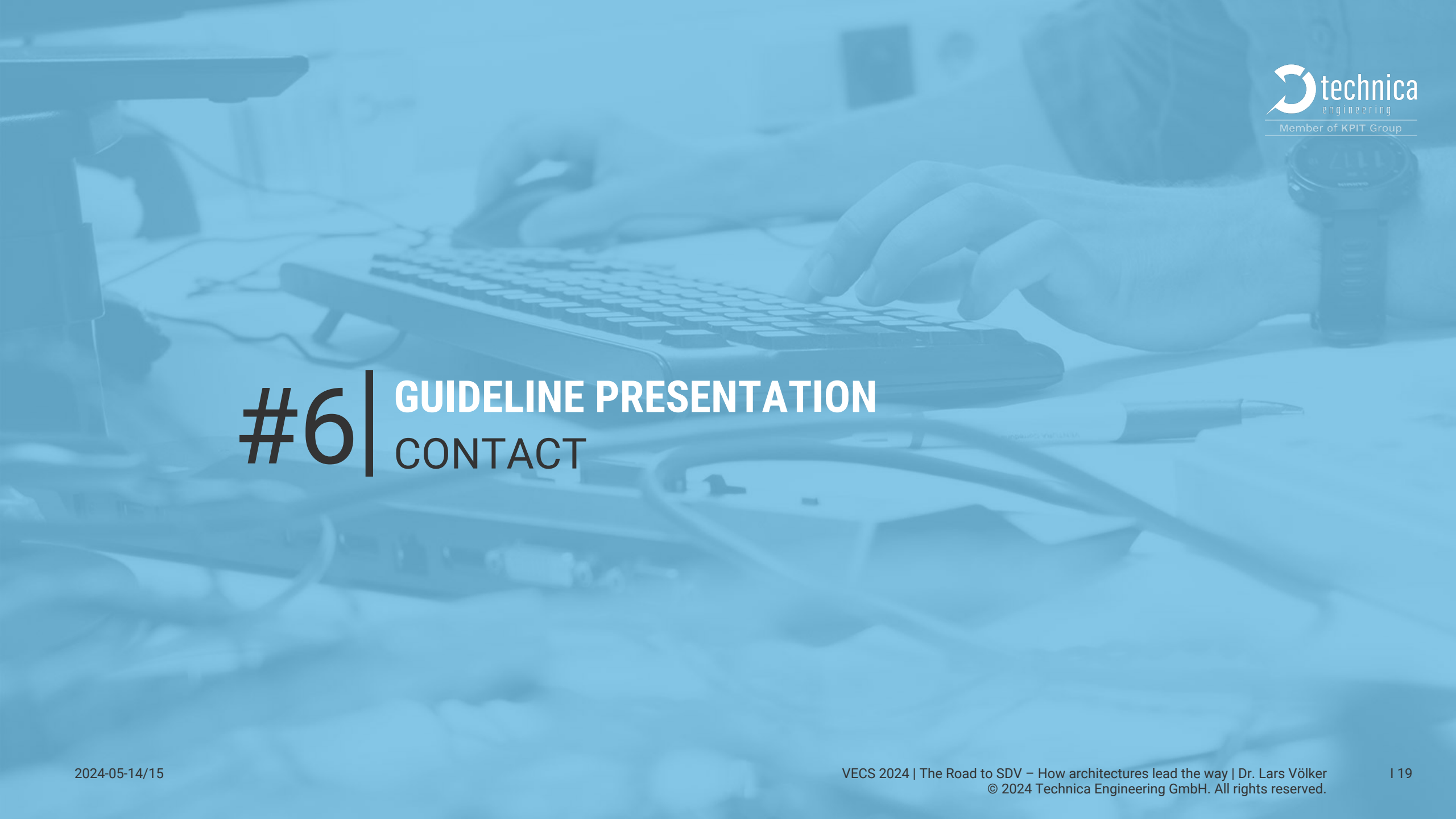
SUMMARY

- SDV allows for faster and later software delivery

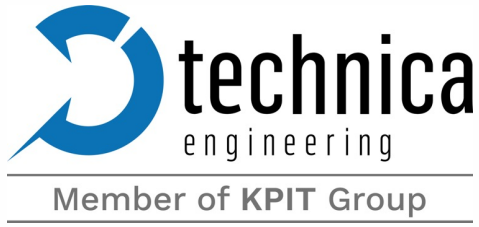
- Architecture needs to be designed for SDV
- Lower complexity and avoid proprietary technology
- Platform security instead of late security concept
- ARXML and development toolchain should be improved
- Automated ECU and integration testing is a must

- Please avoid common mistakes





#6 | GUIDELINE PRESENTATION
CONTACT



Technica Engineering GmbH

Leopoldstraße 236
80807 Munich
Germany

DR. LARS VÖLKER
Technical Fellow
lars.voelker@technica-engineering.de
+49 175 11 40 982