RTAutoSec

Foundational Security for Your Real-Time Autonomous System Network

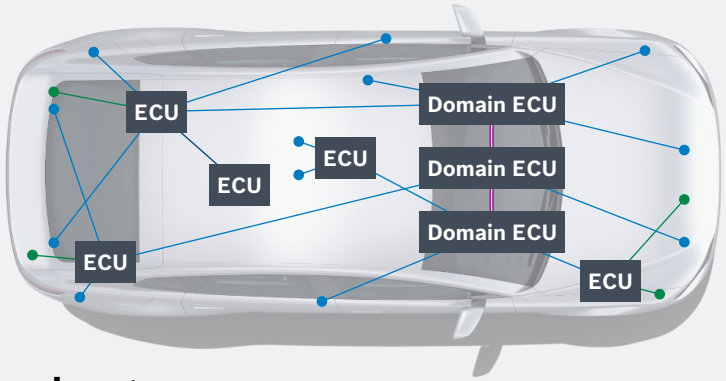
Friedrich Wiemer
Robert Bosch GmbH
XC-CE/ECS1
July 8th, 2025

BOSCH

Introduction
In-Vehicle Architectures

Domain-based Architecture

Ethernet CAN LIN

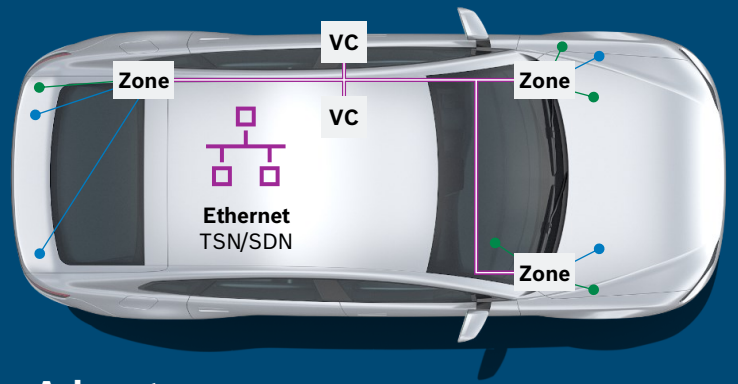


Disadvantages

- Vehicle consists of weakly coupled subsystems (domains)
- Deterministic communications between systems only exceptional with extra effort
- Specific individual solutions and software partitioning
- Limited systematic communication network design

Zonal-based Architecture

Ethernet TSN/SDN CAN LIN

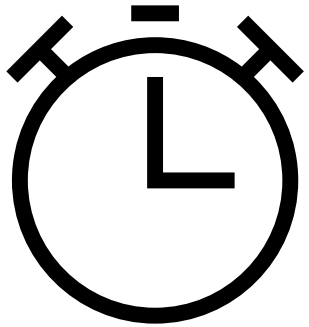


Advantages

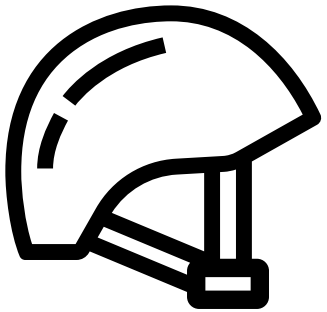
- One unified network with common time zone
- Synchronous actions in multiple ECUs with accuracy < 200 ns at vehicle level
- Systematic communication network design with simulation and validation

TSN Time Sensitive Network, **SDN** Software Defined Network, **VC** Vehicle Computer

Introduction
Foundational requirements for Real-Time Autonomous Systems

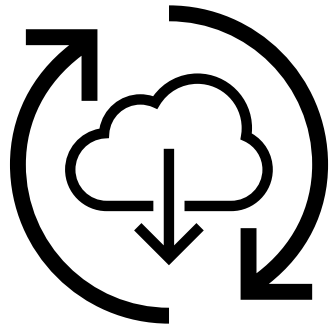


Hard real-time behavior



Safety critical

Immediate start up



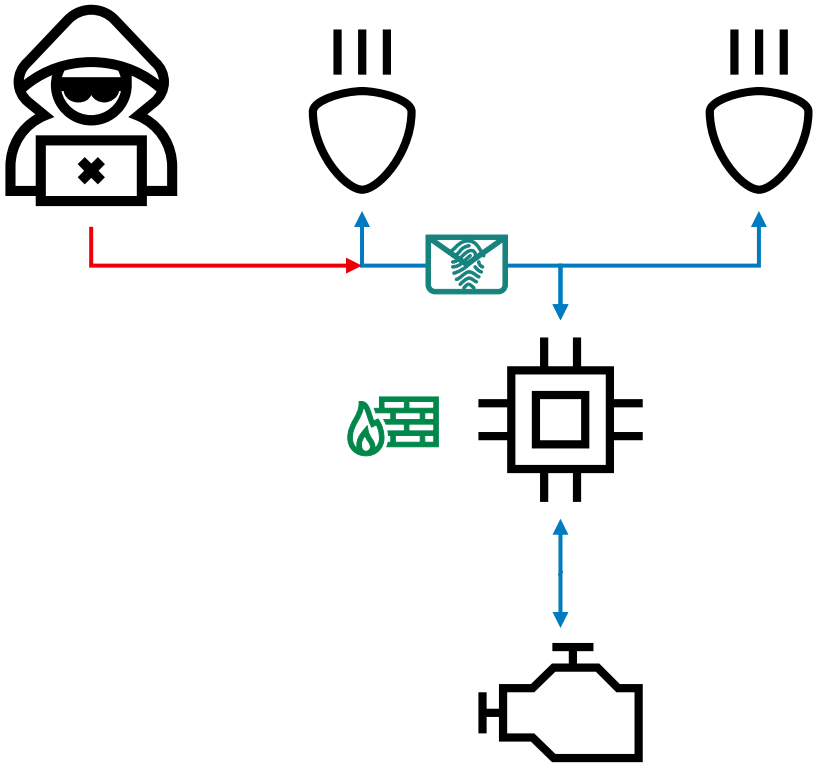
Updateability

Introduction
Why protect communication

- Attackers exploit most accessible parts
- Exploit chain:
  - Insufficient separation on architectural & network level
  - Insufficient authentication of messages
- Allows attacker to inject & spoof messages

  → start engine / open doors → steal car

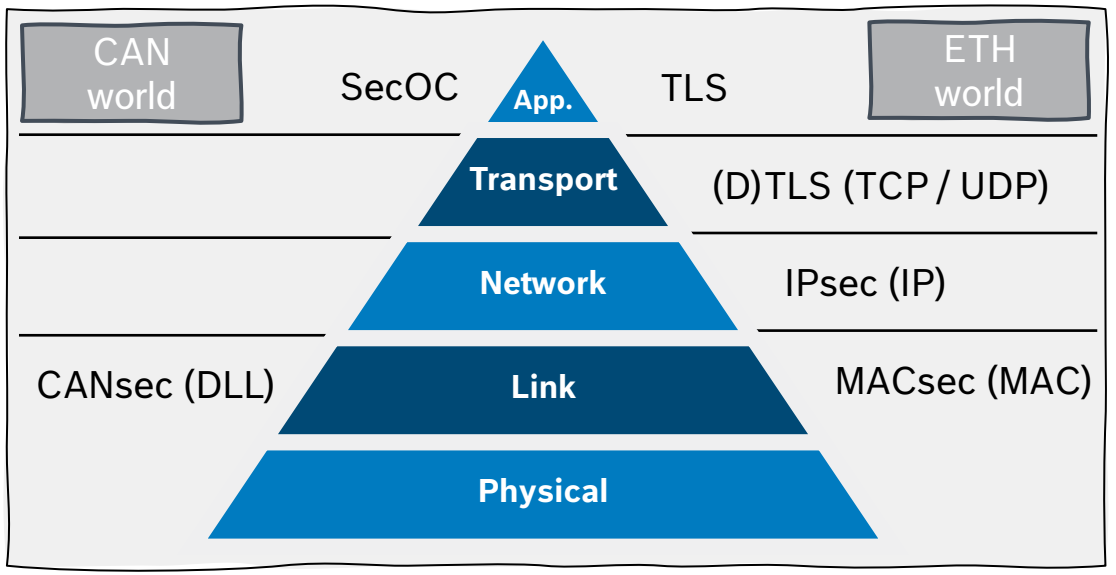
- Holistic security concept should
  - Separate communication with firewalls
  - Authenticate messages with security protocol
  - and more 😉



Introduction
How to protect communication

- Mechanisms for secure in-vehicle communication:
  - **SecOC** (CAN, FR, ETH),
  - **(D)TLS** (ETH) for TCP (UDP)
  - **IPsec** (ETH) for IP,
  - **MACsec** (ETH) for MAC / LLC
  - **CANsec** (CAN) on Layer 2
- Security protocols typically consists of:
  - Authentication
  - Key Agreement
  - Data protection

Communication and security protocols

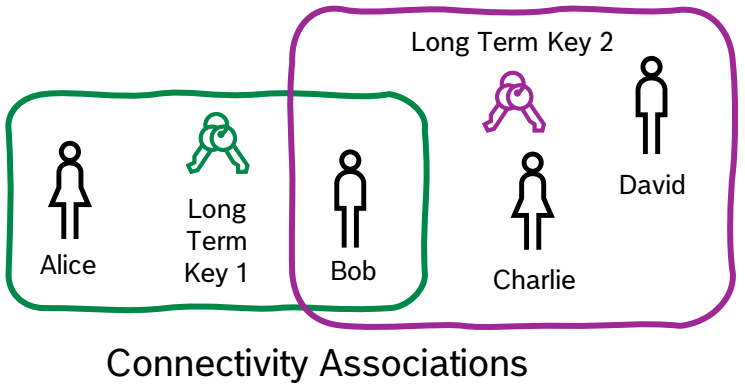


Confidentiality / Authenticity

Standardized / Custom Implementations

Security protocols implemented in HW (PHY or MAC) gives us additional crypto performance.

Introduction
Communication Architecture



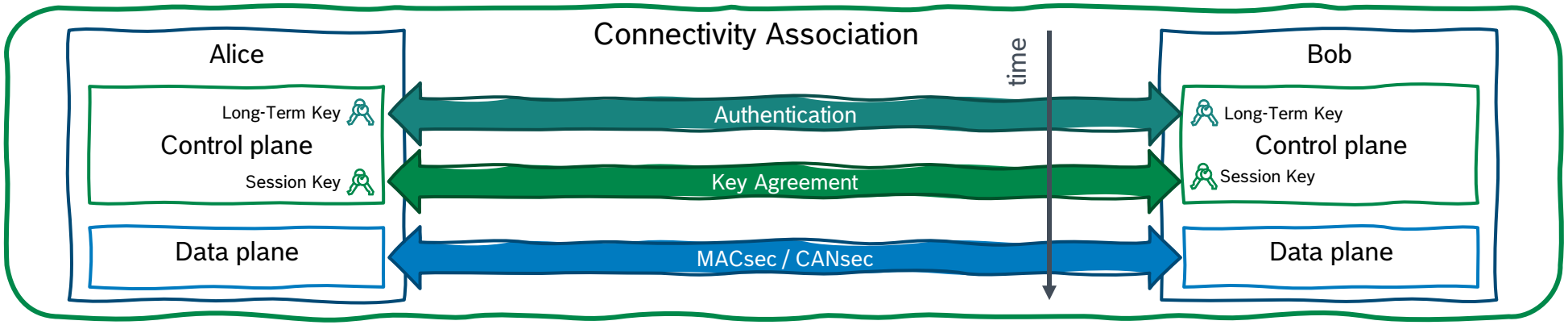
Three consecutive phases

- Authentication of peers
- Session key negotiation between peers

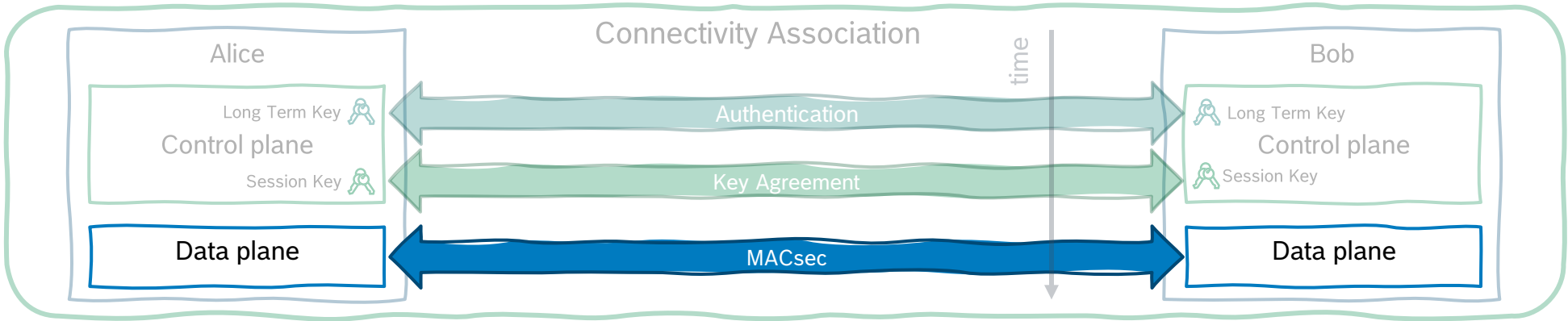
} Control plane

- Secure communication

} Data plane

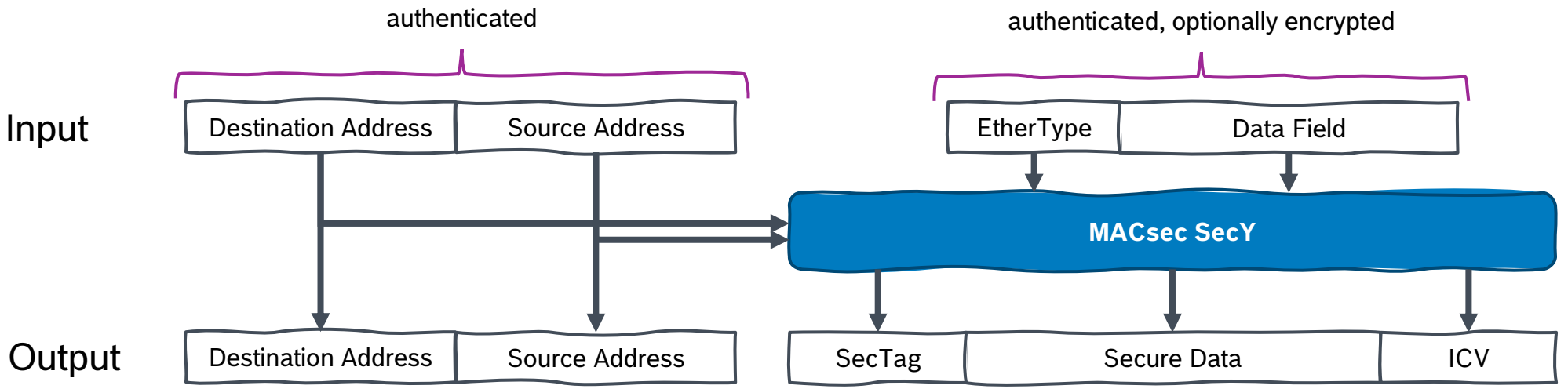


MACsec



BOSCH

MACsec
High-Level View



MACsec
Automotive MACsec relevant standards

2006 — **IEEE 802.1AE**
Initial MACsec standard

2011 — **IEEE 802.1AEbn**
Extension adding 256bit crypto

2013 — **IEEE 802.1AEbw**
Extension adding 64bit packet numbers (XPN modes)

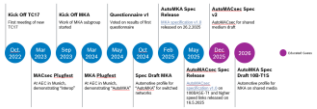
2017 — **IEEE 802.1AEcg**
Extension adding Ethernet Data Encryption devices

2018 — **IEEE 802.1AE**
Updated MACsec standard, incorporating previous extensions

2022 — **AUTOSAR R22-11**
Requirements on MACsec & Specification of MKA

2024 — **OA TC17**
Automotive profile for MKA finalized, published begin 2025

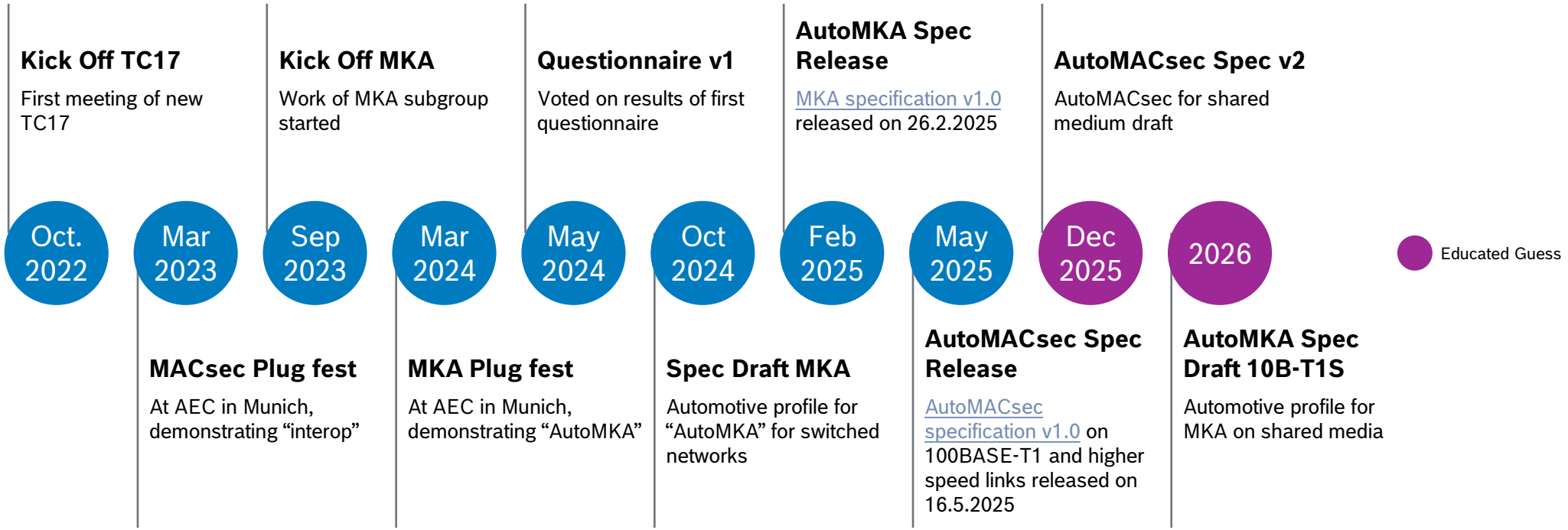
2025 — **OA TC17**



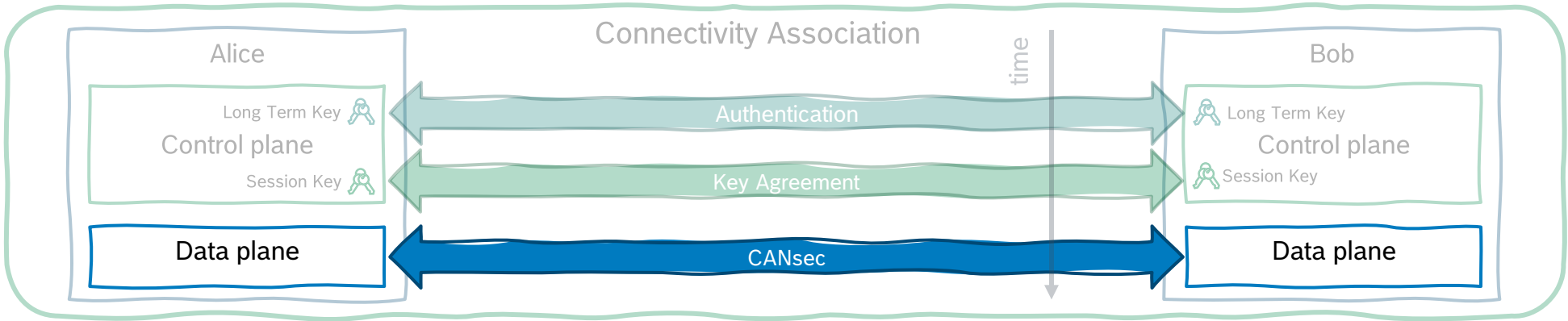
2026 (?) — **AUTOSAR**
Updates to MACsec and MKA components

OA TC17
Automotive profile for MKA on shared media

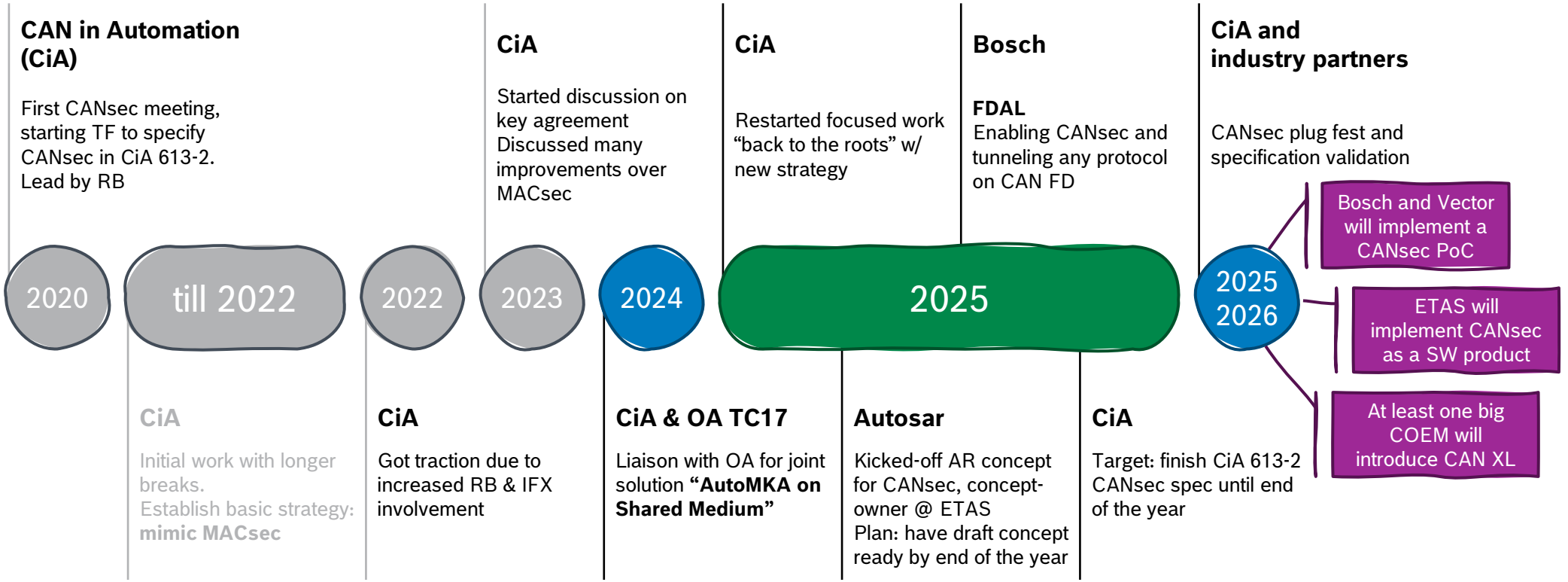
MACsec
Open Alliance TC17 Automotive Profile for MACsec



CANsec


Connectivity Association
Alice
Bob
time
Long Term Key
Control plane
Session Key
Authentication
Key Agreement
CANsec
Data plane
Data plane

CANsec
Timeline

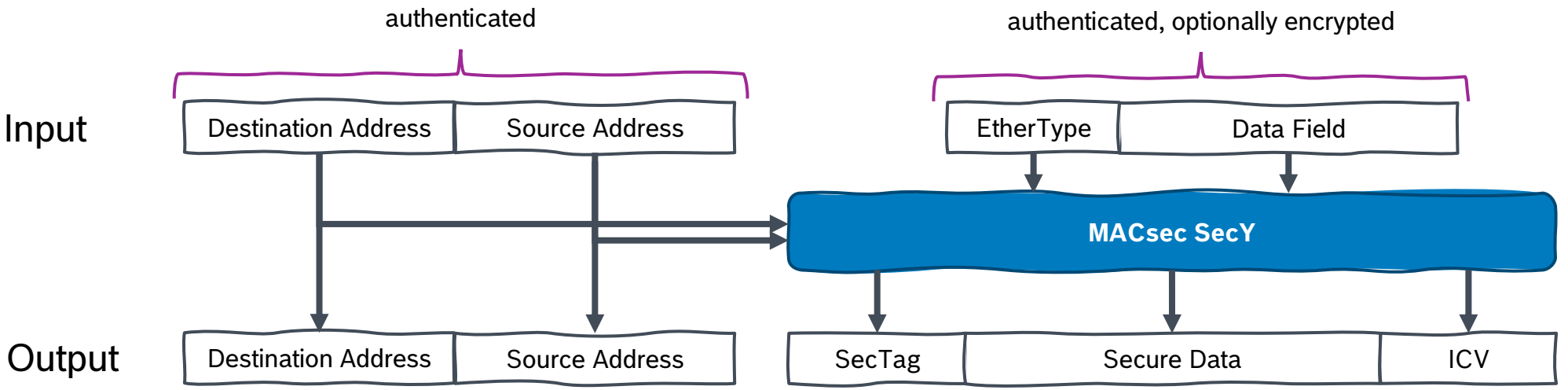


Supporters
for new strategy: **Reuse MACsec**

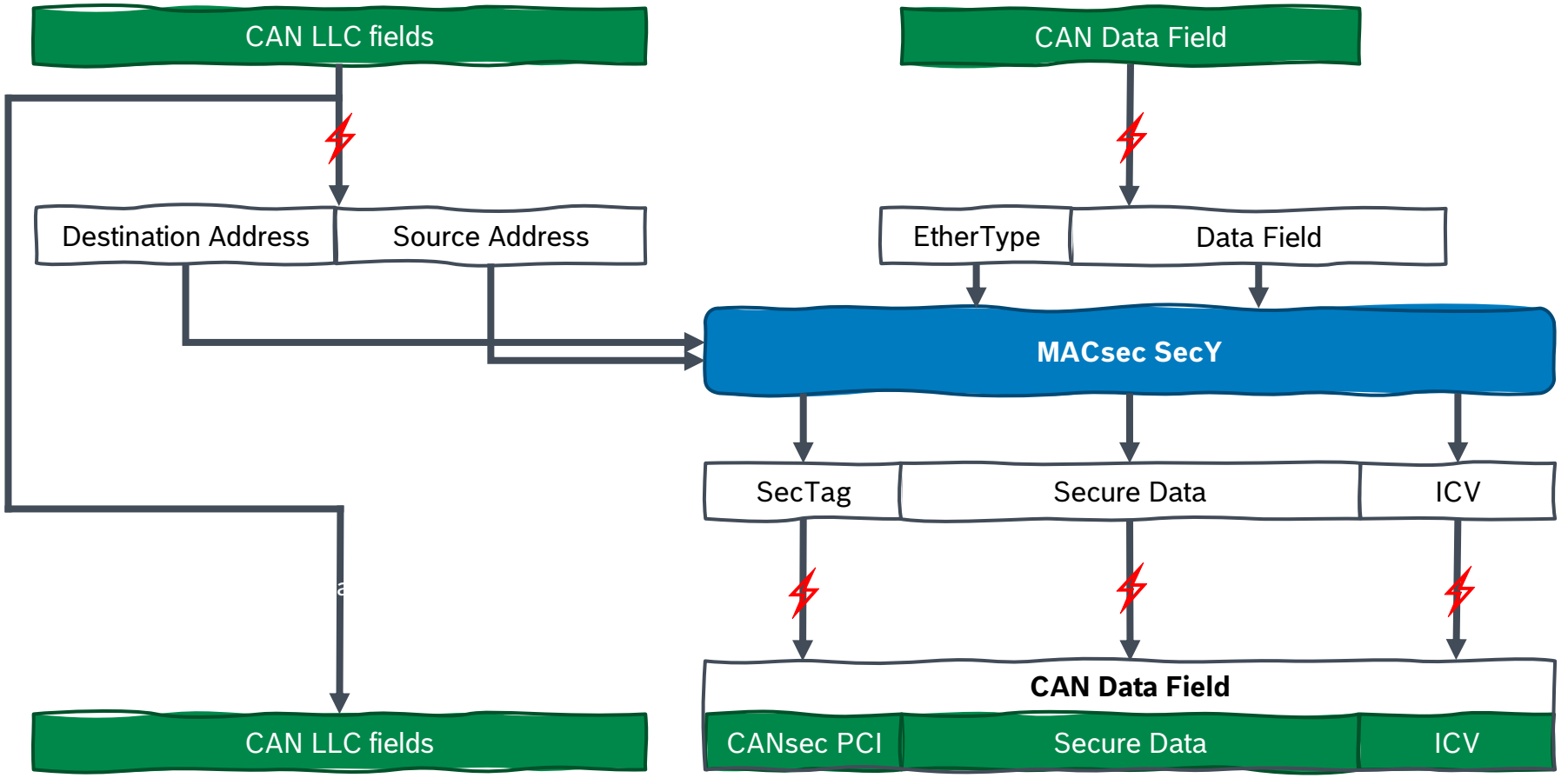
VECTOR RENESAS NXP life.augmented CARIAD

BOSCH

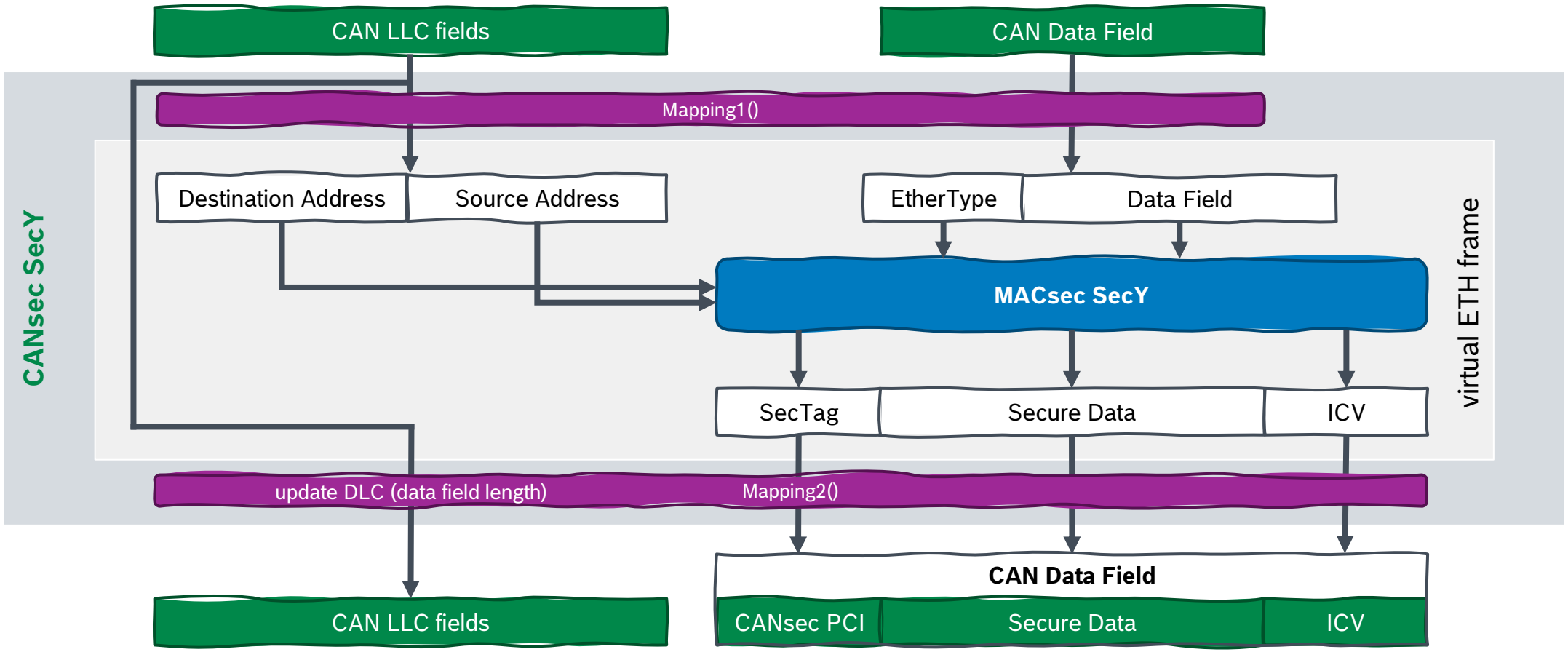
MACsec
High-Level View



CANsec
High-Level View

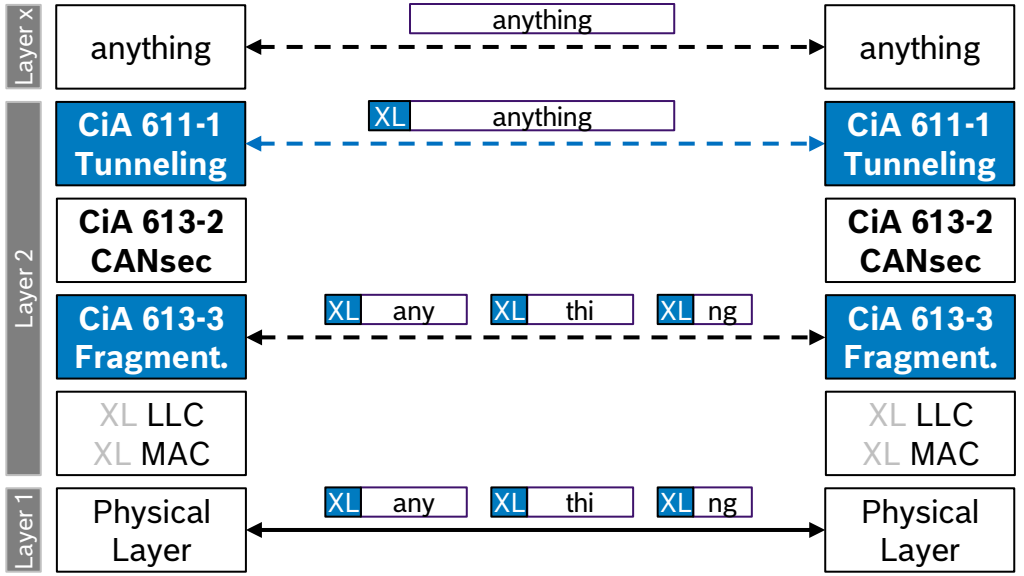


CANsec
High-Level View

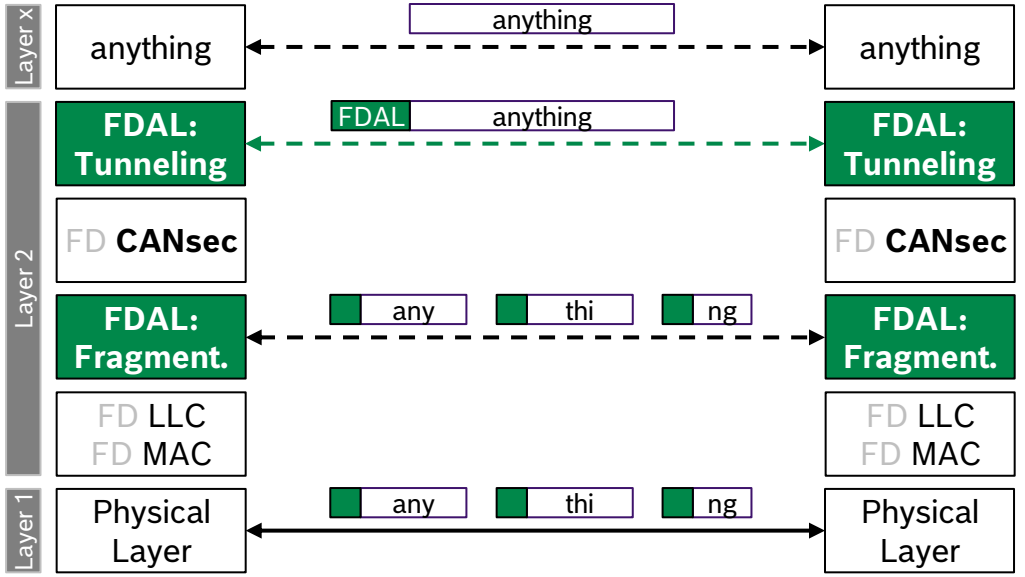


CANsec for FD
FD Adaptation Layer (FDAL)

Existing: CAN XL solution



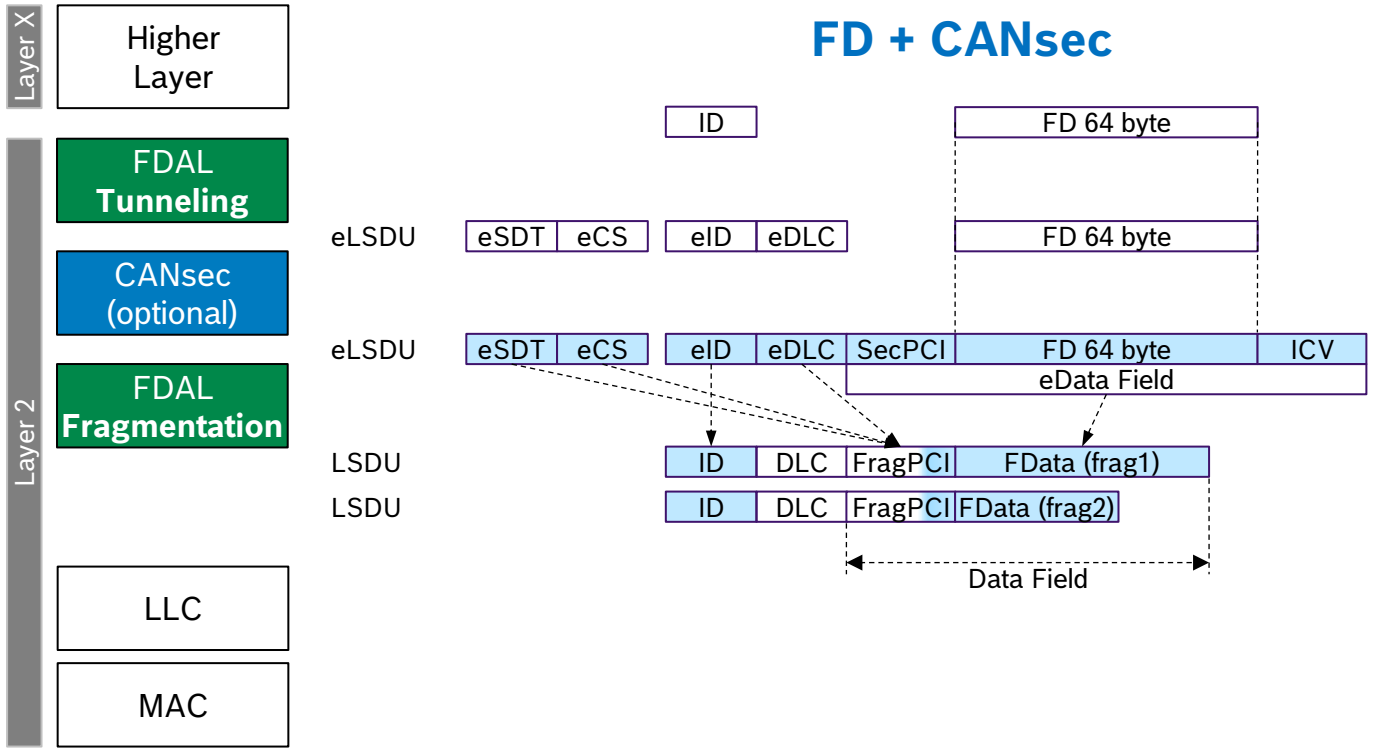
New: enable “anything over FD”



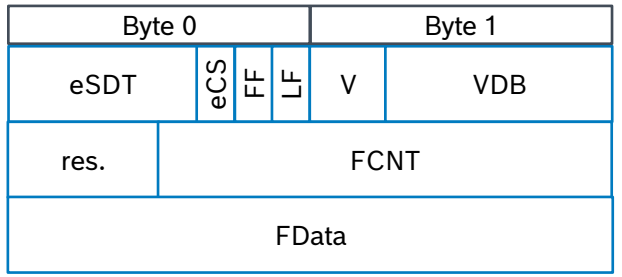
Key property: Unified/Similar FD and XL solution

- FDAL Tunneling **is like** CiA 611-1 (CAN XL Tunneling)
- FDAL Fragment. **is like** CiA 613-3 (CAN XL Fragment.)
- FD CANsec **is like** CiA 613-2 (CAN XL CANsec)

CANsec for FD
FDAL Example: FD tunneling

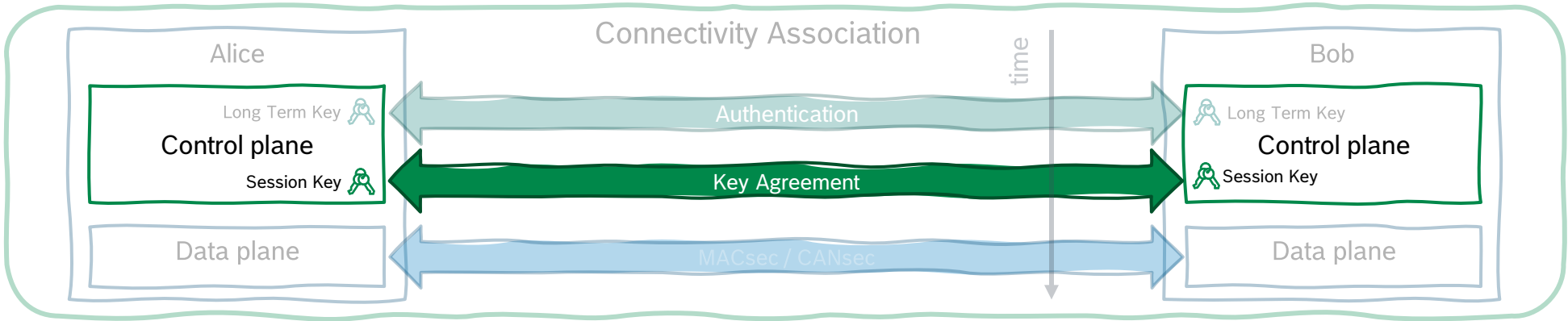


FDAL Frag. PDU



Field	Name	Description
eSDT	Extended SDU Type	The payload type
eCS	Extended CANsec	Whether CANsec was applied
First Frame	First Frame	If this is the first fragment of an extended LSDU
Last Frame	Last Frame	If this is the last fragment of an extended LSDU
V	Version	FDAL version
VDB	Valid Data Bytes	To detect padding bytes in the CAN frame
FCNT	Fragment Counter	Detects missing frames
FData	Fragment Data	The actual fragmented payload

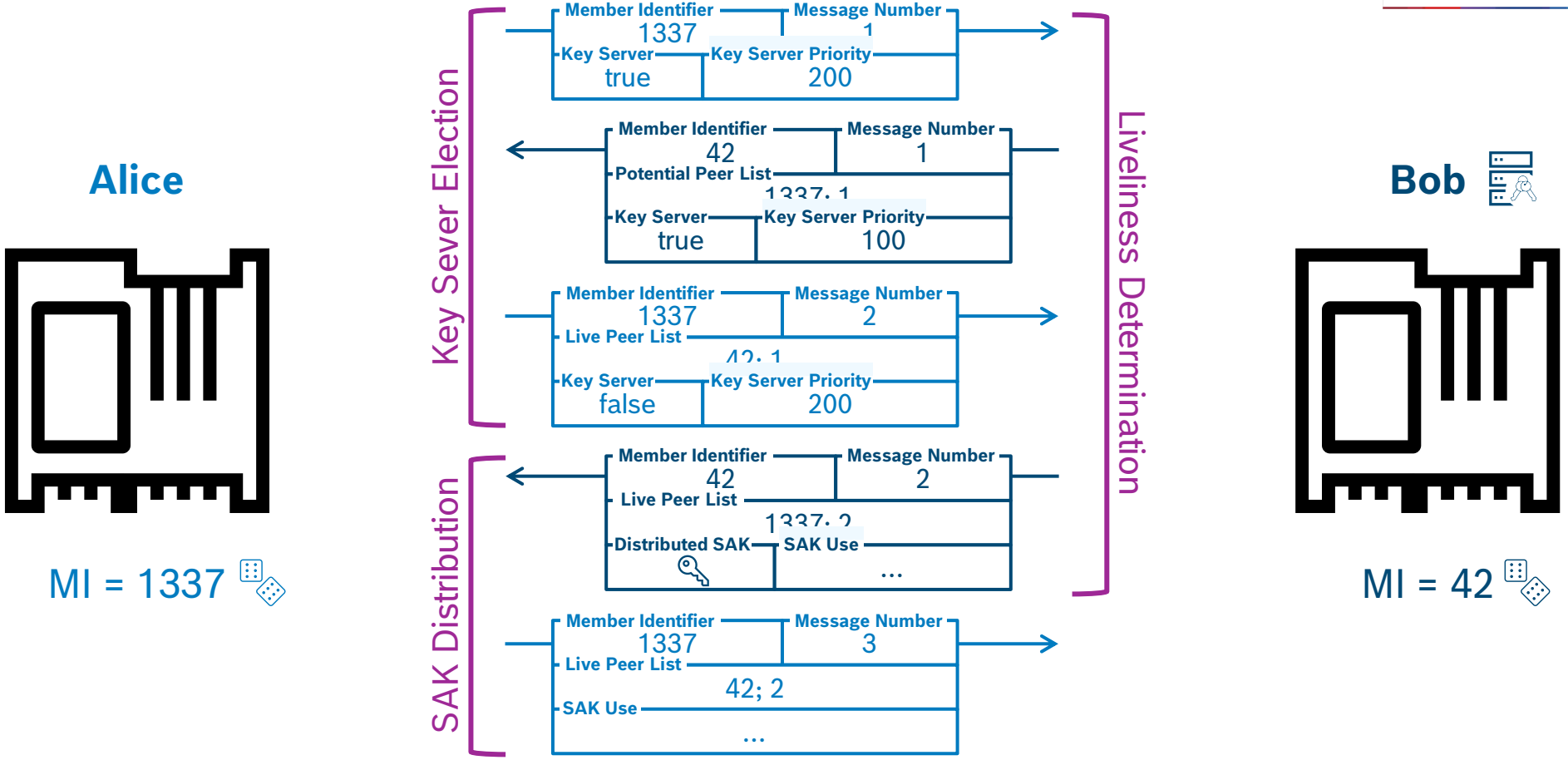
Key Agreement for MACsec & CANsec



BOSCH

MACsec Key Agreement (MKA)
Overview

Backup: MKA Details

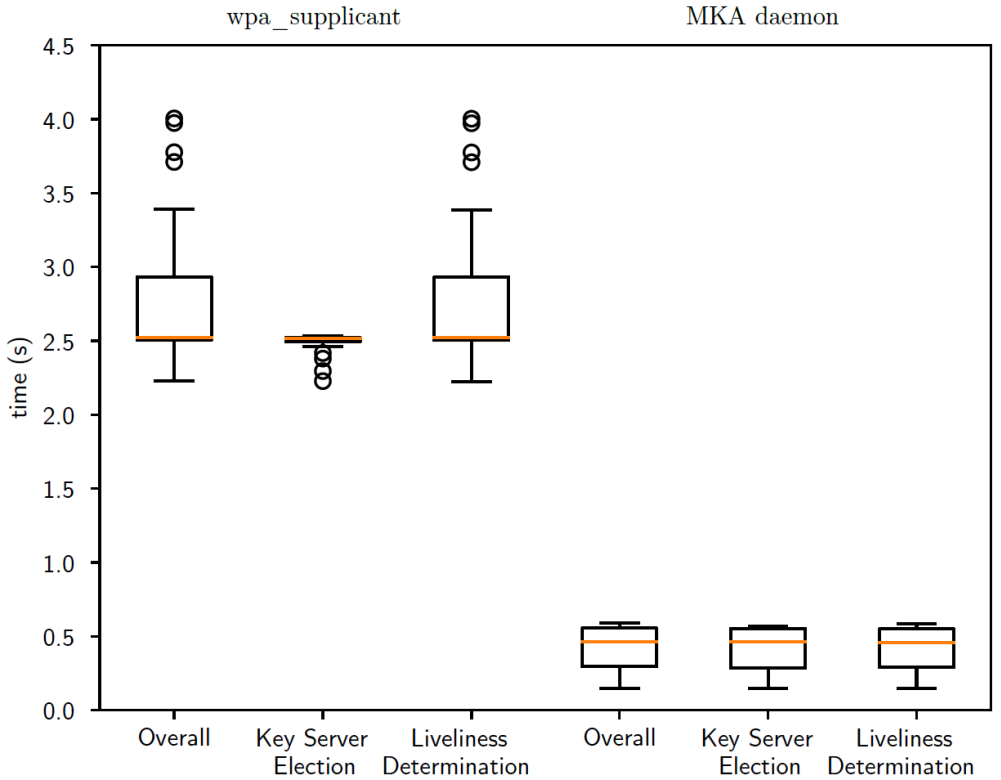


MACsec Key Agreement (MKA)
Automotive MKA for switched Ethernet

Automotive MKA Specification

Automotive MKA Specification for Automotive Ethernet and MACsec

https://opensig.org/wp-content/uploads/2025/03/OA_MACsec_Automotive-MKA-v1.pdf



Pre-Selected Key Server

Immediate Responses and Reset Detection

Hello Time Ramp Up

MACsec Key Agreement (MKA)
Automotive MKA for shared medium

- Open Alliance TC17 works on proposals for MKAv4
  - Optimized for shared medium / groups
- Already supported by MKA, but
  - AutoMKA optimizations lead to high bus load for bigger groups
  - Time-to-Key-Agreement (TTKA) is not good enough
- Currently two suggestions in discussion
- Timeline: submit to IEEE in July or November

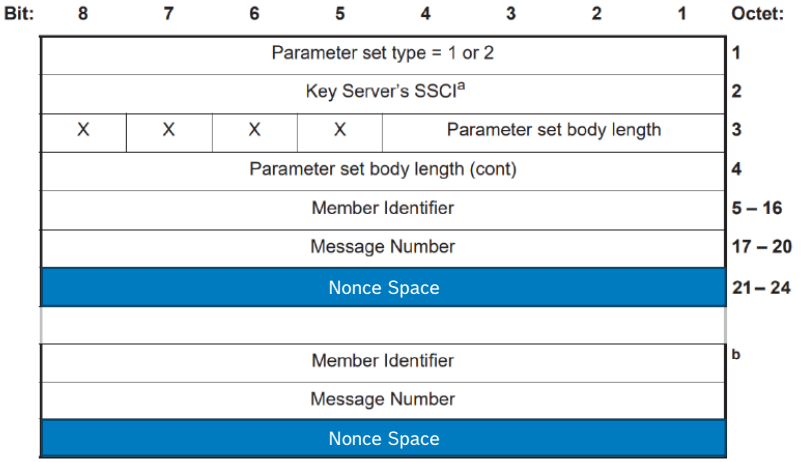
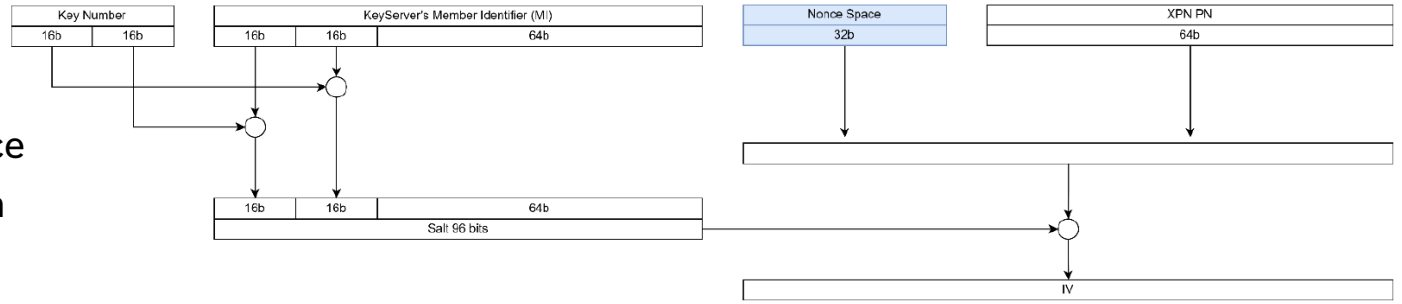
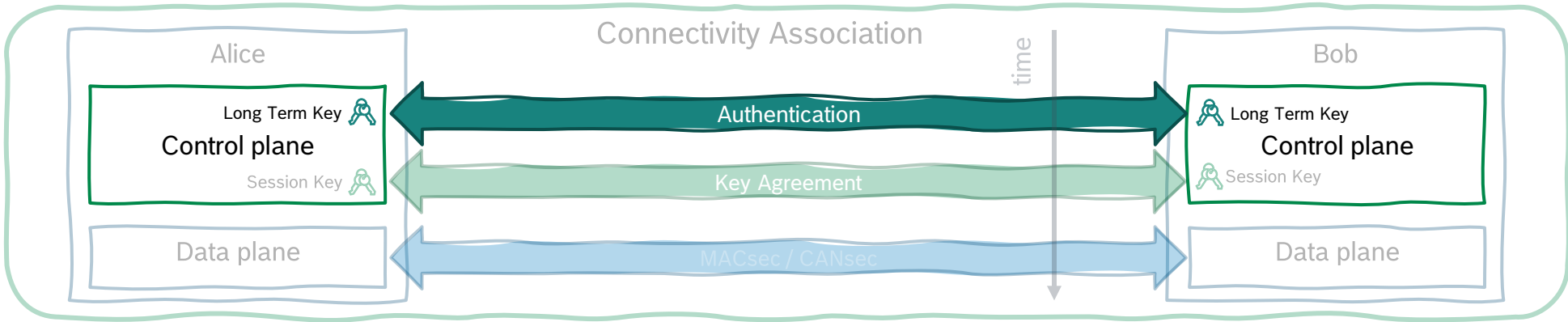


Figure 11-9—Live Peer List and Potential Peer List parameter sets

- Proposal 1: Nonce Spaces
  - Extends Peer List Parameter Set
  - Key Server manages nodes nonce space
  - Nonce space is included in IV derivation

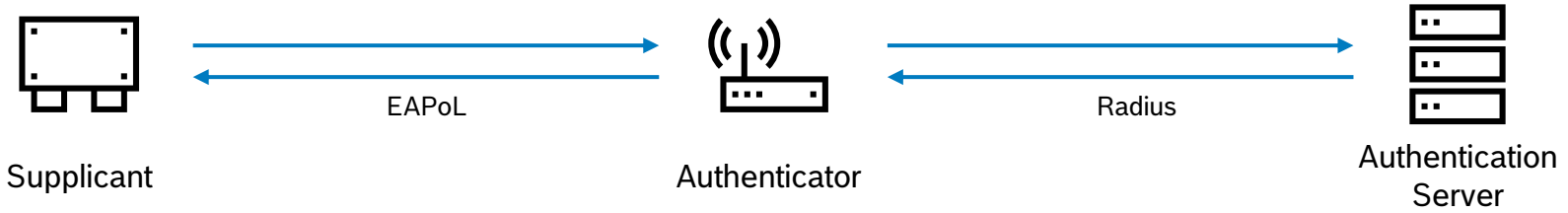


Authentication



BOSCH

Authentication
Extensible Authentication Protocol



Authentication

EAP-TLS

- Best security properties
- Complex
- Slow
- Additional nodes (authentication server)
- Additional crypto (asymmetric & certificates)

Pre-Shared-Keys

- Requires trusted environment for setup
- Most simple solution
- OEMs have proprietary solutions in place for key distribution in field (from SecOC)
- No additional crypto required
- Replacing ECUs is not straight forward
- No standardized way for pre-sharing-keys

Pre-shared keys look like 80’s techn. – but IVNs are engineered networks and thus much simpler than complex, dynamic IT networks.

Open Problems

Efficient Group Key Agreements

- Meet very tight timing constraints for TTKA
- Efficiently implementable
  - Simple state-machine for HW-only impl.
  - based on symmetric crypto

Efficient Key Injection Methods

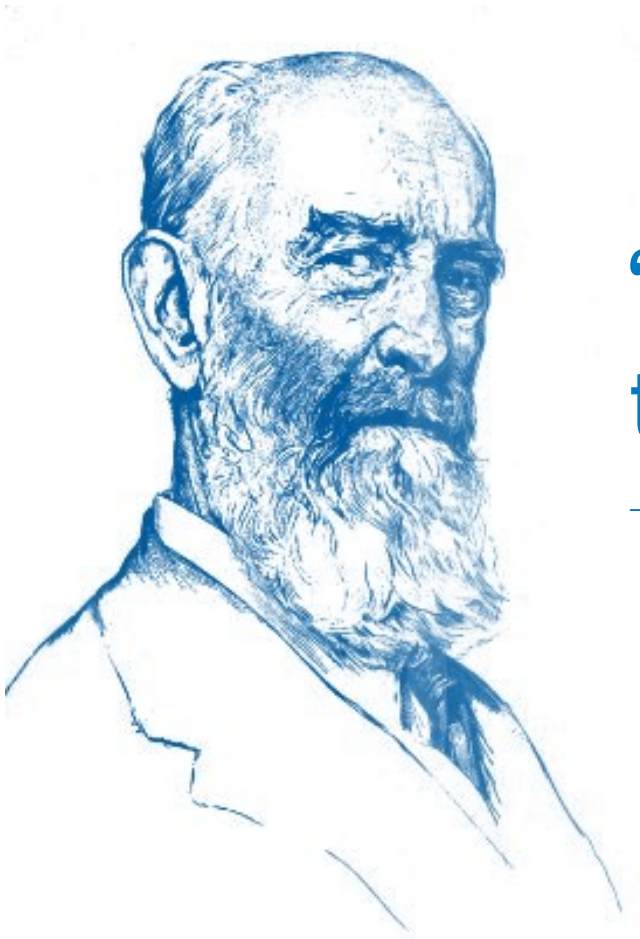
- Efficiently inject keys in SW-less devices
- Example: MACsec on SW-less edge node
  - How to get keys in there?
  - Standardized approach?

Efficient Authentication Protocols

- Efficiently implementable in constrained devices
- Optimized for engineered networks
  - Less dynamic
  - More known static configuration

BOSCH

Wrap up
Questions?



“Secure, or not secure, that is the question”

—freely adapted from Hamlet

Efficient Group Key Agreements

- Meet very tight timing constraints for TTKA
- Efficiently implementable
  - Simple state-machine for HW-only impl.
  - based on symmetric crypto

Efficient Authentication Protocols

- Efficiently implementable in constrained devices
- Optimized for engineered networks
  - Less dynamic
  - More known static configuration

Efficient Key Injection Methods

- Efficiently inject keys in SW-less devices
- Example: MACsec on SW-less edge node
  - How to get keys in there?
  - Standardized approach?