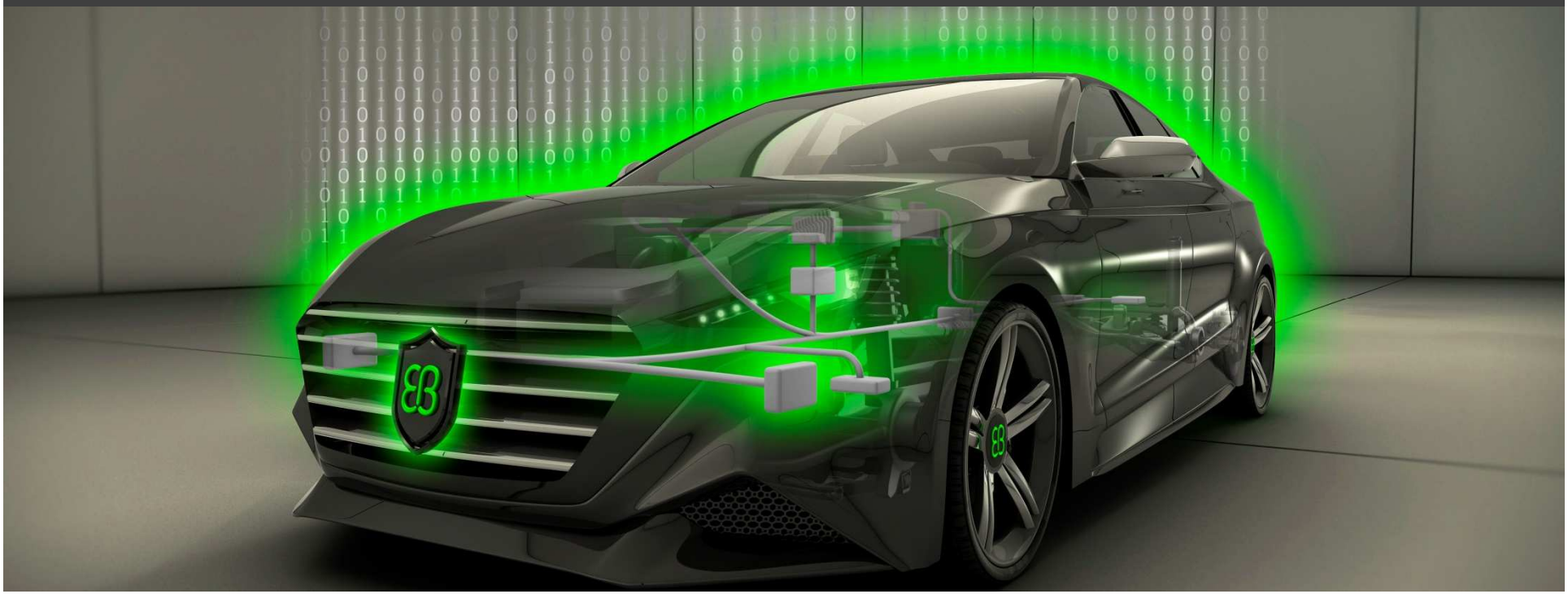
Software Architecture for Secure ECUs

Rudolf Grave

EB TechDay - June 2015



Agenda

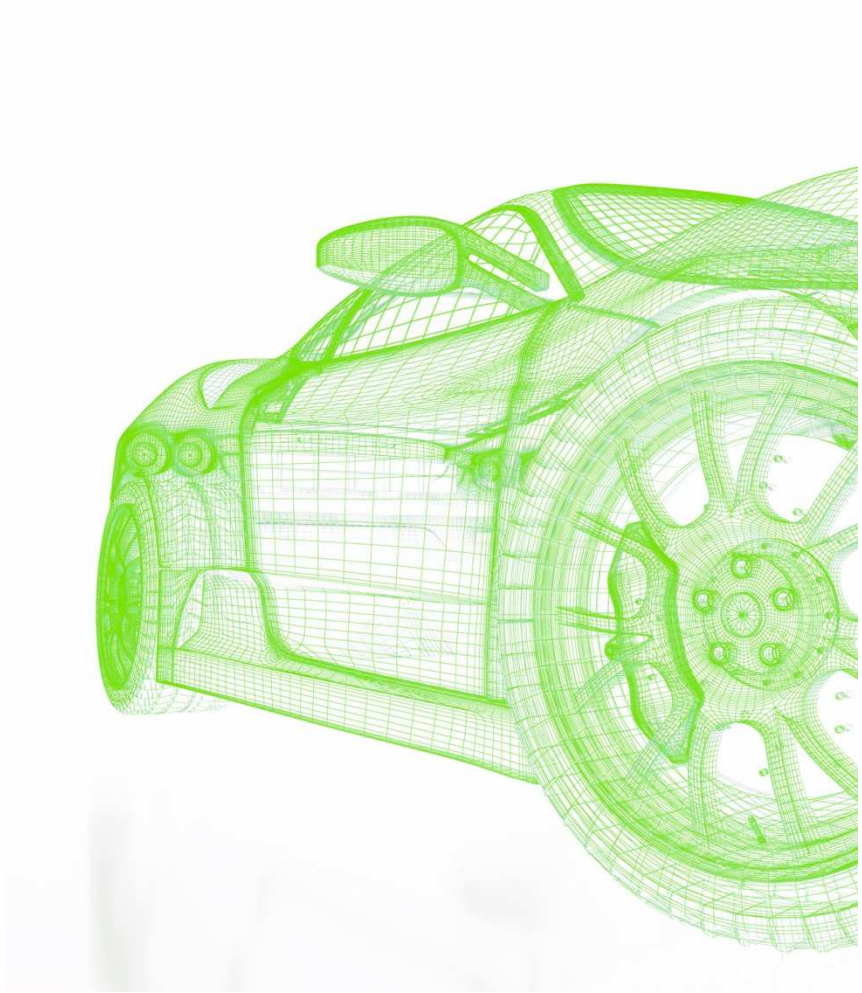
"No safety without security and vice versa"

Established Safety Concepts

Safety Analysis Methods for Security Analysis

Secure Software Architecture Extensions

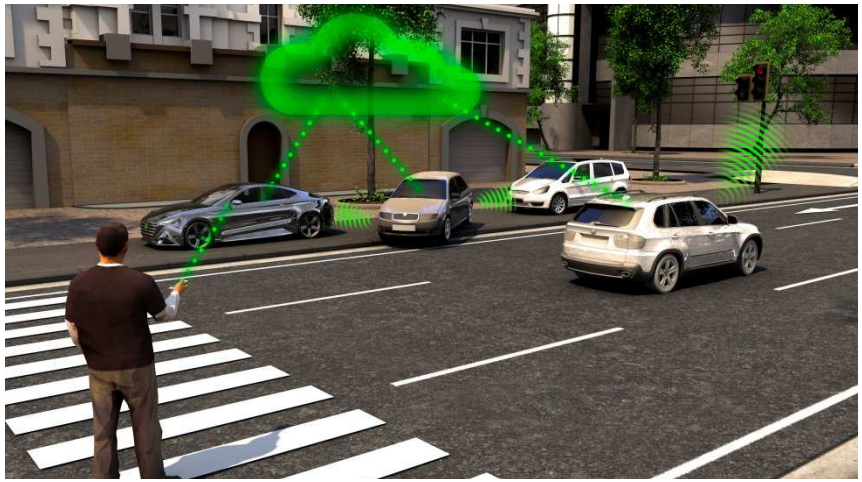
Summary



The Car of the Future:
Increased comfort – Increased potential for damage

- **Autonomous Driving**
  - Highly safety-critical
  - Requires latest data from the “cloud”

- **Car-2-X Communication**
  - Communication with
    - other cars
    - infrastructure
    - mobile phone or other devices
  - ECUs have access to the on-board network



Recent security breaches

OpenSSL “Heartbleed” vulnerability

- Sensitive data accessible via maintenance function
- Encryption and maintenance functions are technically unrelated
- Cause: implementation error

Remote door unlock

- Attacker could open cars with fake SMS
- Various vulnerabilities:
  - Partially unencrypted communication
  - Provision of sensitive data
  - Missing integrity checks
  - Weak or identical encryption keys
  - Replay attacks possible

Agenda

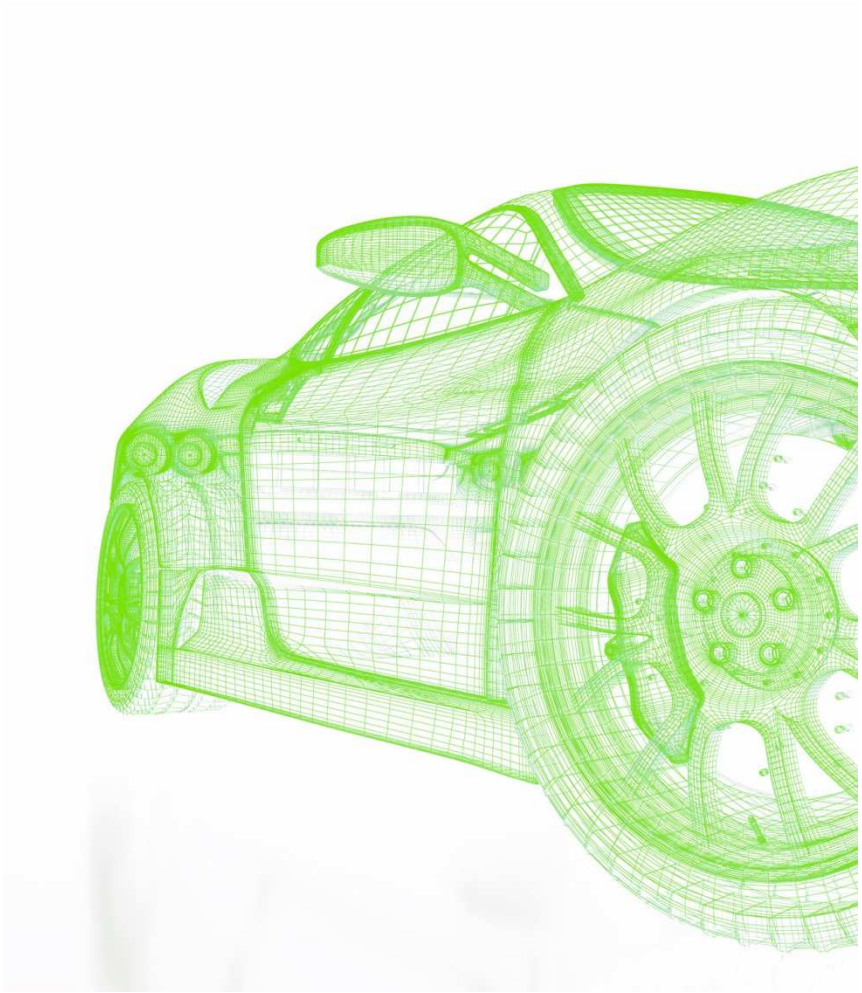
"No safety without security and vice versa"

Established Safety Concepts

Safety Analysis Methods for Security Analysis

Secure Software Architecture Extensions

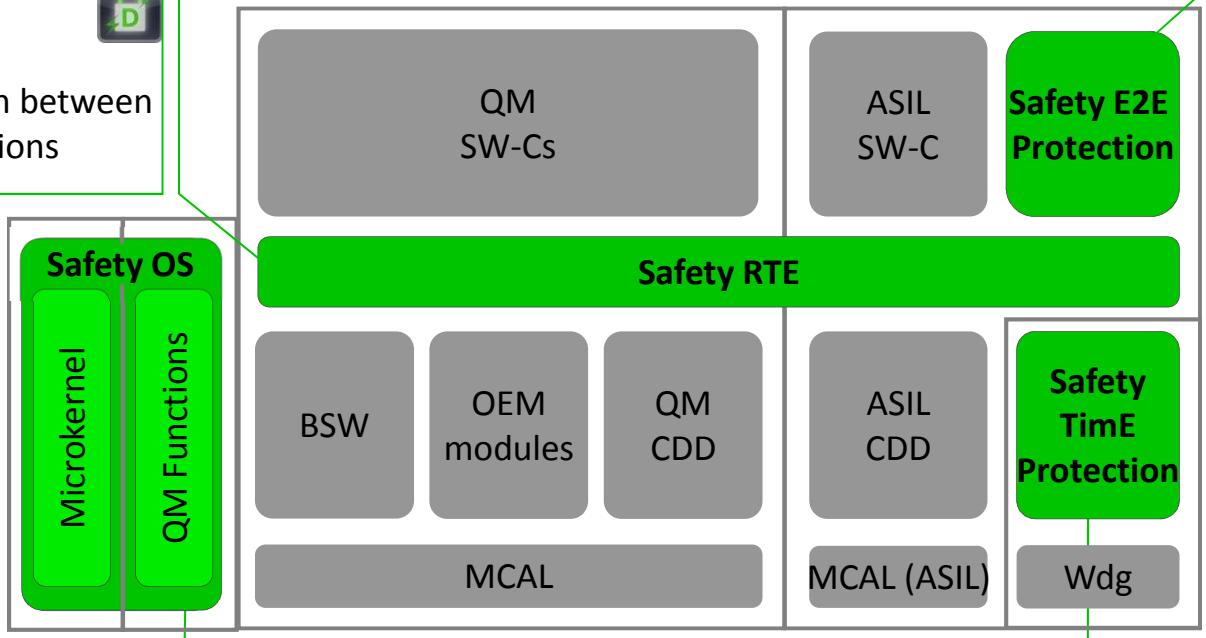
Summary



Memory Partitioning

Safety RTE
Protected communication between Memory Partitions

Safety E2E Protection
Safe communication to other ECUs

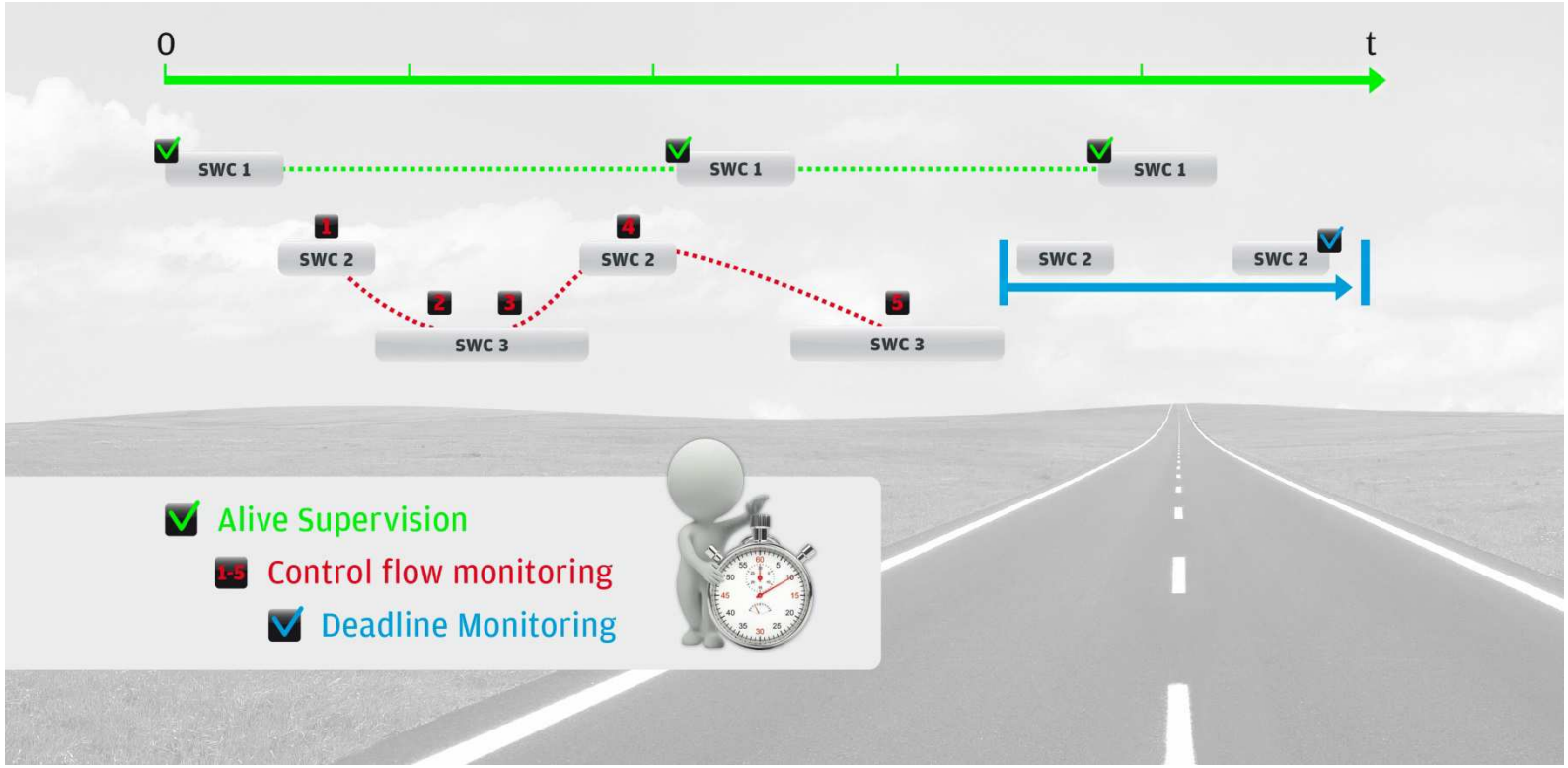


Memory Partitions

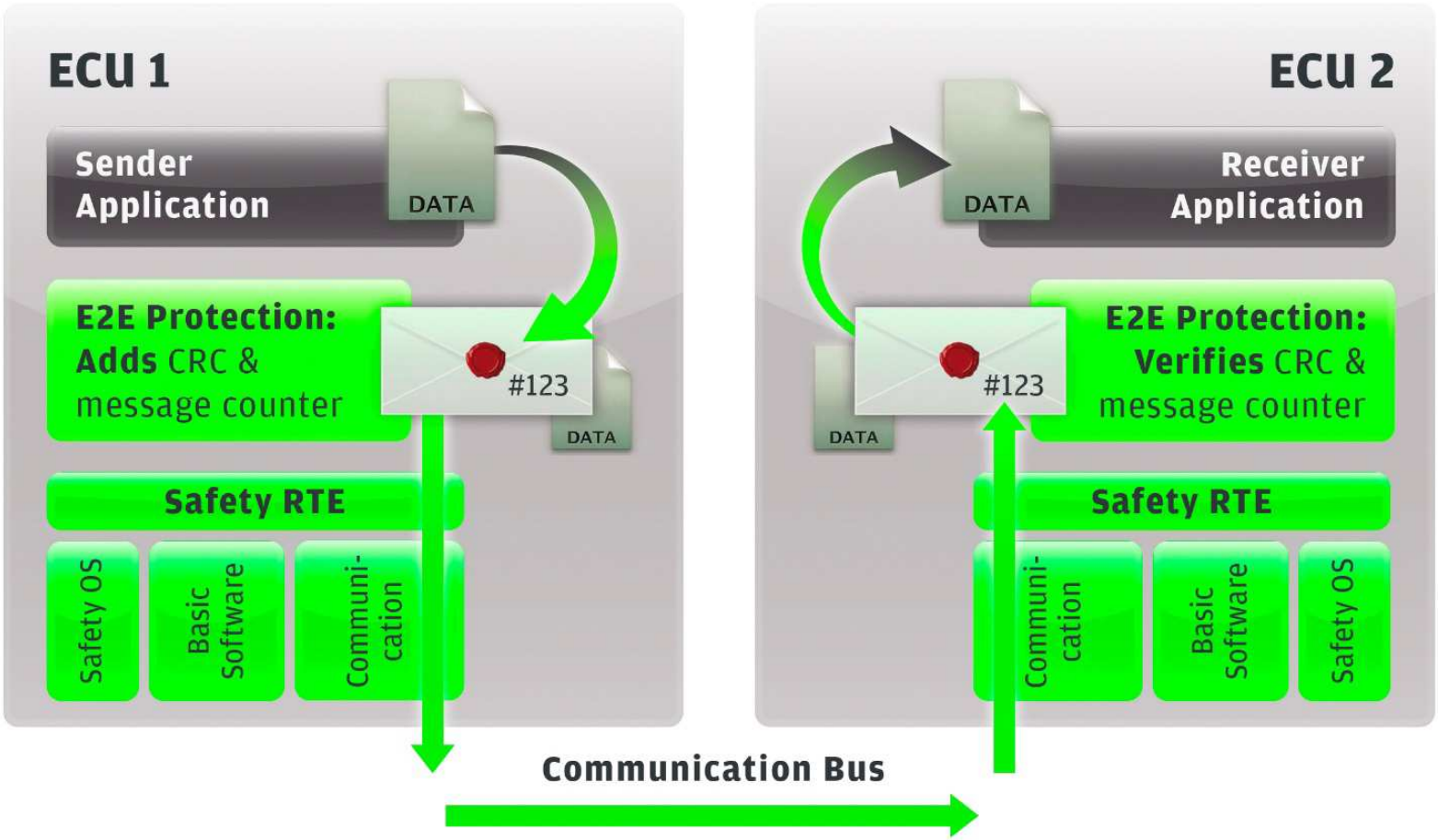
Safety OS
- Data Protection
- Stack Protection
- Context Protection
- OS Protection
- Hardware Error management

Safety TimE Protection
- Alive supervision
- Deadline Monitoring
- Control flow monitoring

Time and Execution Protection



Communication Protection



Agenda

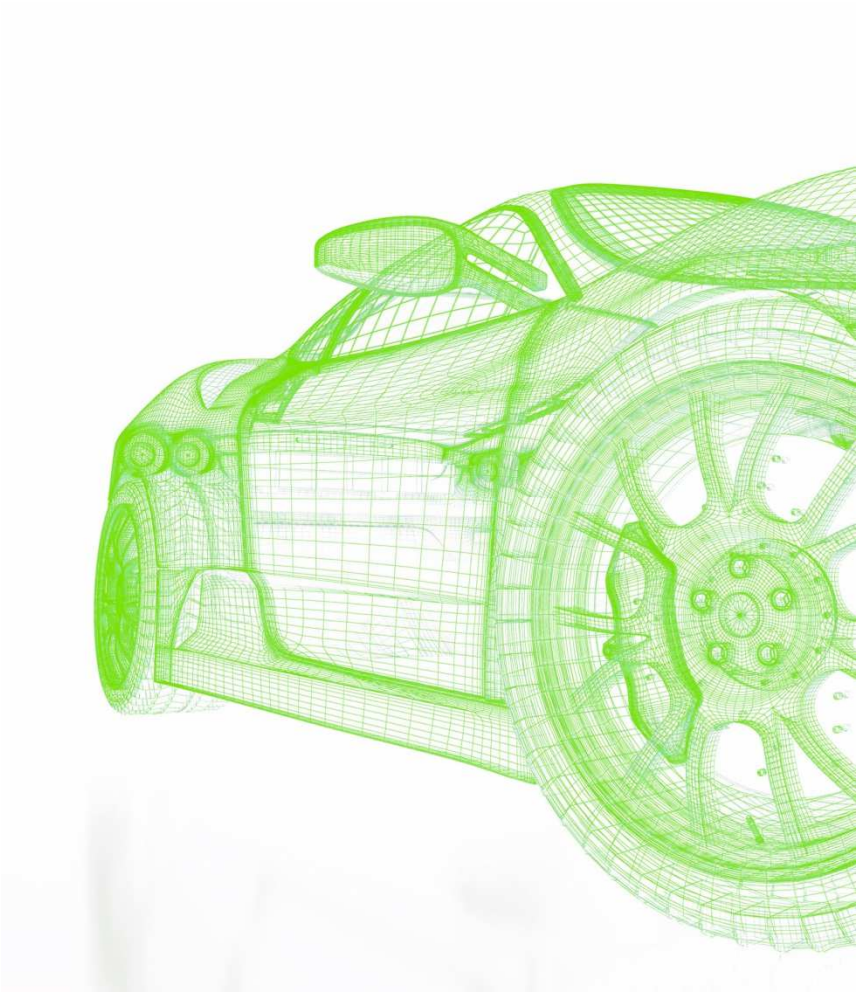
"No safety without security and vice versa"

Established Safety Concepts

Safety Analysis Methods for Security Analysis

Secure Software Architecture Extensions

Summary



Static vs. Dynamic Threat Model

Safety: static threat model

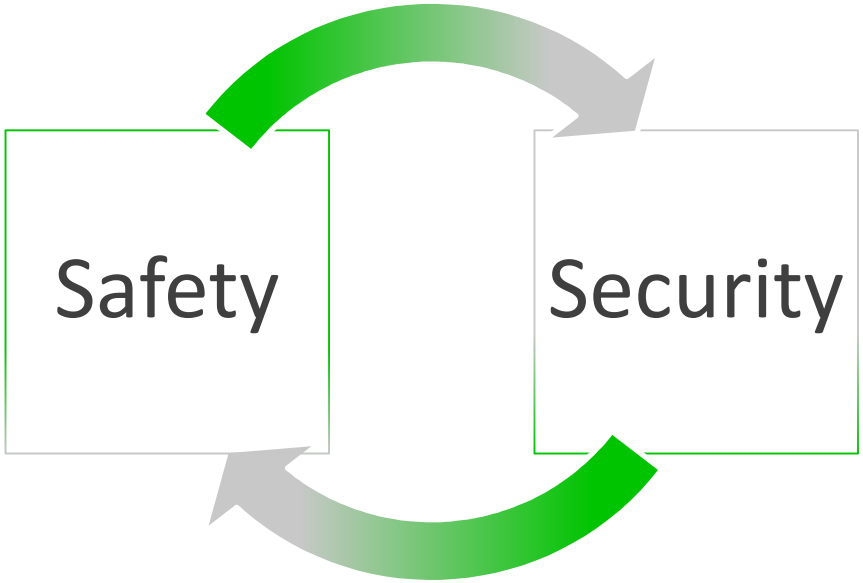
- Threats are known at system design
- Threats are internal, e.g. random or systematic faults
- Iterations improve existing model with new knowledge

Security: dynamic threat model

- New threats can emerge during system operation
- Threats are external
- Intelligent opponent has to be considered

Extending Safety Analysis to Security Analysis

- Safety and security rely on **risk models**
- It´s crucial to recognize and **use synergies**
- Extend hazard and risk analysis with **malicious attacker**
  - Attacker has access to all communication channels
  - Extend safety requirements with security requirements



Searching for security vulnerabilities brings new safety exposures to light and vice versa

Agenda

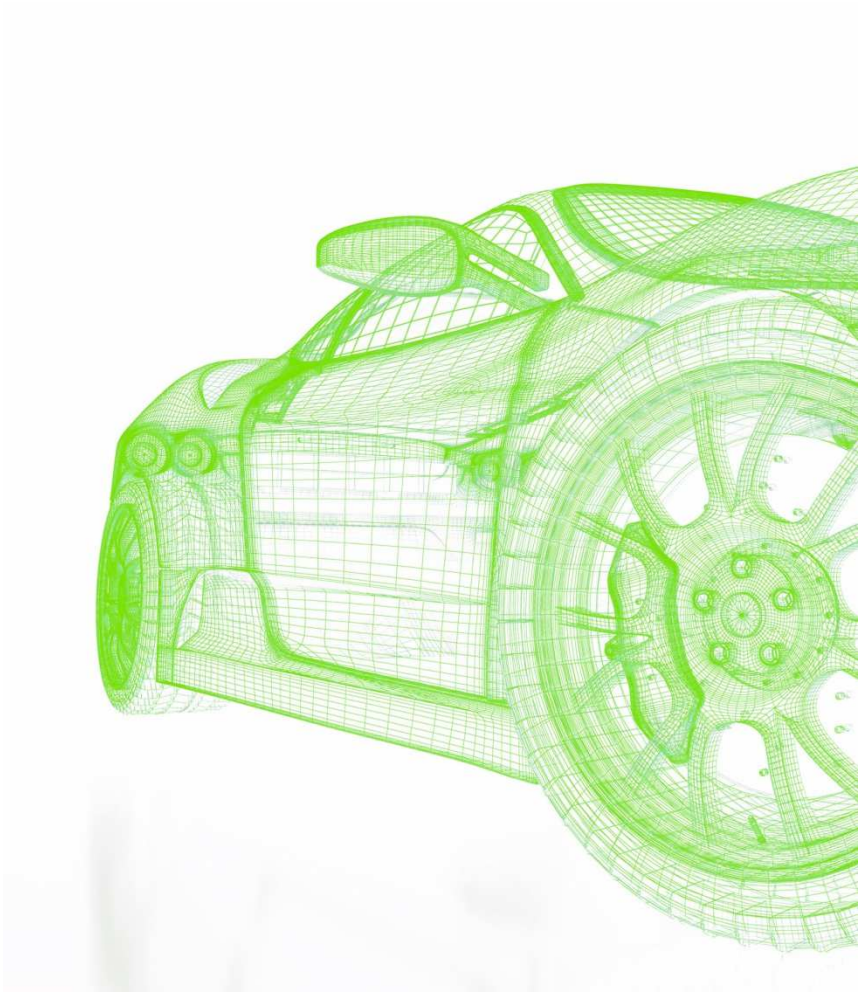
"No safety without security and vice versa"

Established Safety Concepts

Safety Analysis Methods for Security Analysis

Secure Software Architecture Extensions

Summary



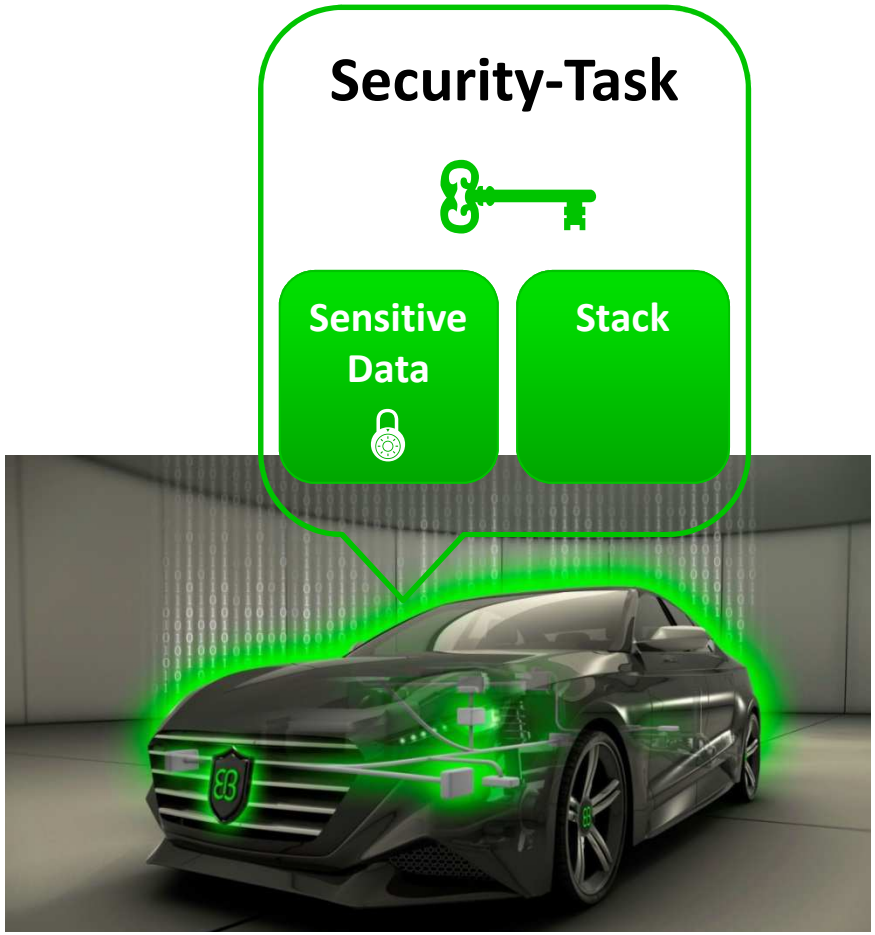
Using Partitioning to Protect Data

- **Memory Partitioning**
  - Read protection
  - Execution protection

⇒ Allow access to sensitive data only for authorized tasks

- **Stack Protection**
  - Stack protected via MPU

⇒ Prevention against stack-overflow attacks



Message Integrity

Threats → **Countermeasures**

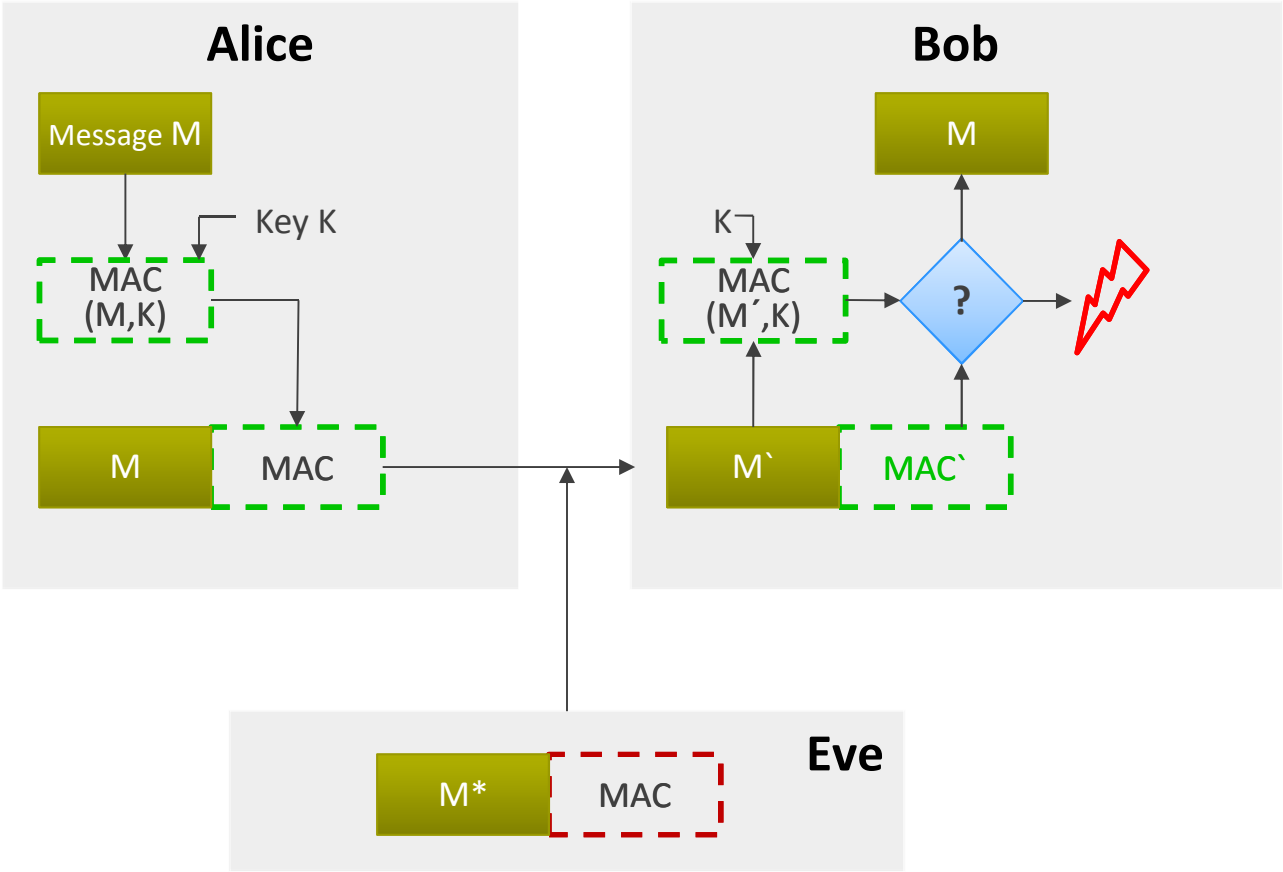
1. **Unauthorized message access**
   - Read, Modify, Delete
2. **Impersonate other user**
   - Initiate communication
3. **Temporal attacks**
   - Replay, Delay

1. **Encryption**
   - Unauthorized read, modification, Impersonation of other user
2. **Signatures**
   - Modification, Impersonation of other user
3. **Integrity Checksums**
   - Modification
4. **Message counters and timestamps**
   - Replay, Delay

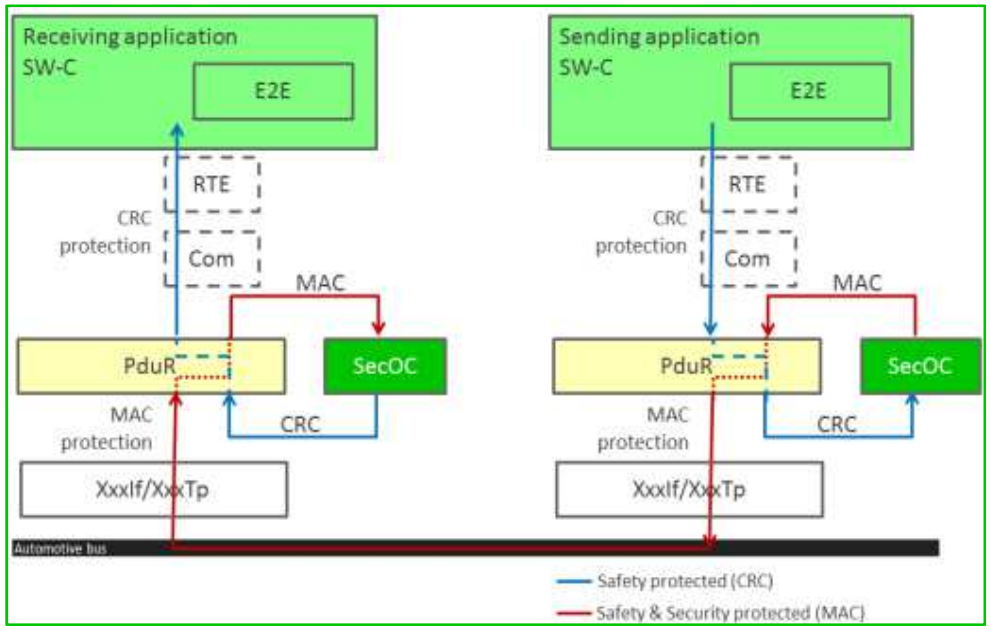
MACs containing signatures and freshness values eliminate most threads

Message Authentication

- MAC with key K is appended to message M
- Key is known to sender and receiver
- Alterations from Eve are detected by Bob



AUTOSAR: E2E and SecOC



Sender

- Unsafe channel between application and SecOC module
  - Protect message with CRC from E2E in application
  - Safe transport between ECUs
- SecOC transforms CRC to MAC
  - Safe and secure transport between ECUs

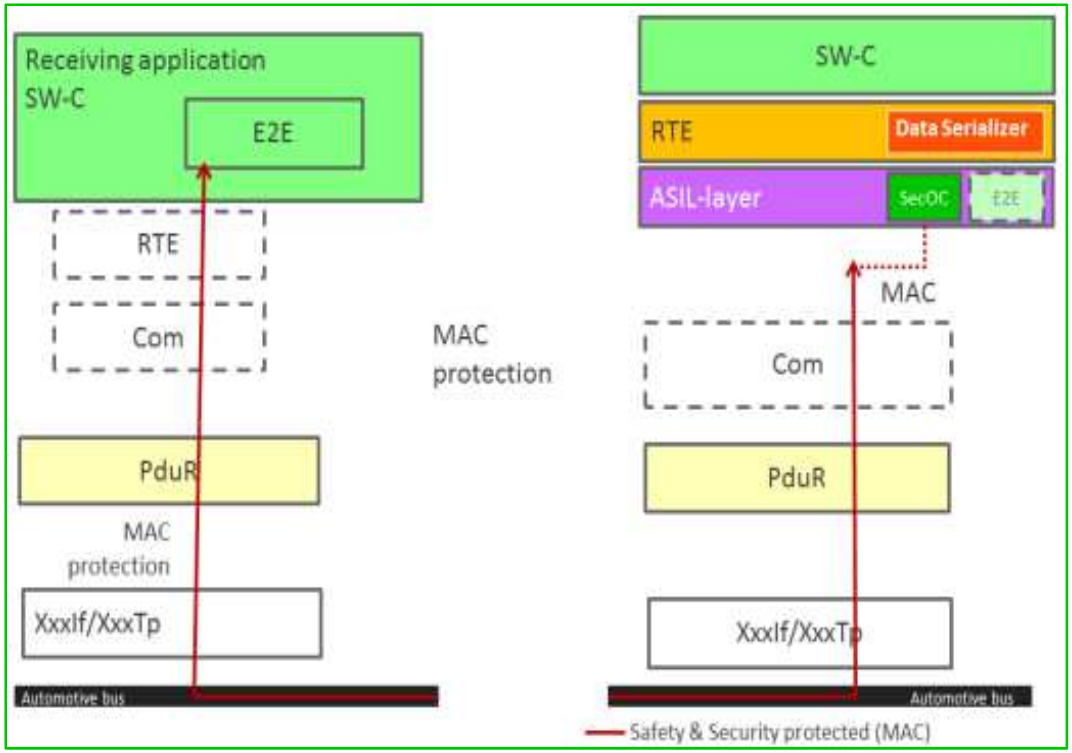
Receiver

- SecOC transforms MAC to CRC
- Application checks CRC

End-to-end protection with SecOC

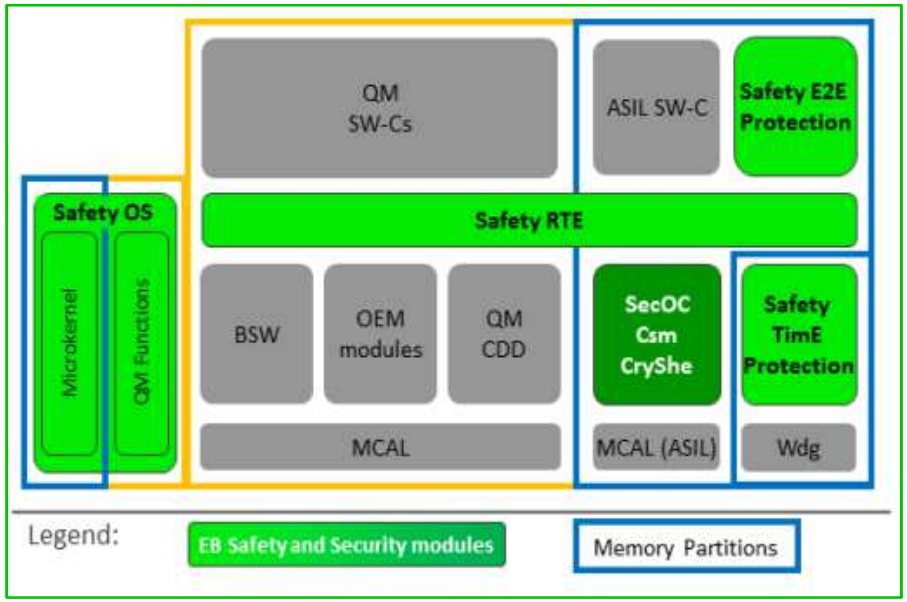
Single SecOC approach

- Use MAC algorithm also for message integrity
- Place ASIL developed SecOC in highest AUTOSAR layer
- Omit overhead for additional end-to-end protection



Elektrobit

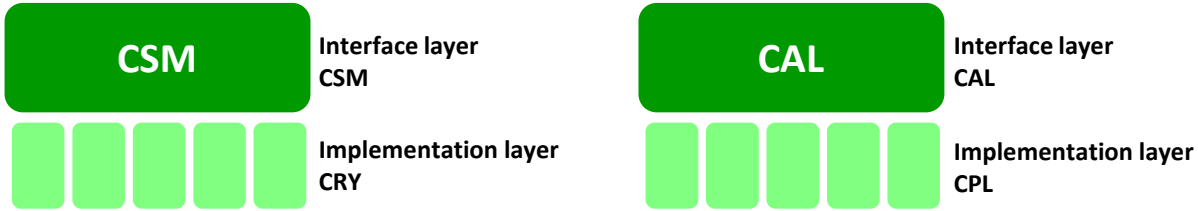
AUTOSAR safety and security architecture



- **Safety OS**
  - Memory write (**safety**), read and execution (**security**) protection
- **TimE Protection**
  - Control flow monitoring (**safety**)
- **E2E protection and SecOC**
  - Data integrity (**safety**, **security**)
  - Authentication (**security**)
- **Csm, CryShe**
  - Data encryption (**security**)

Csm, Cry, CryShe, Cal, Cpl,…

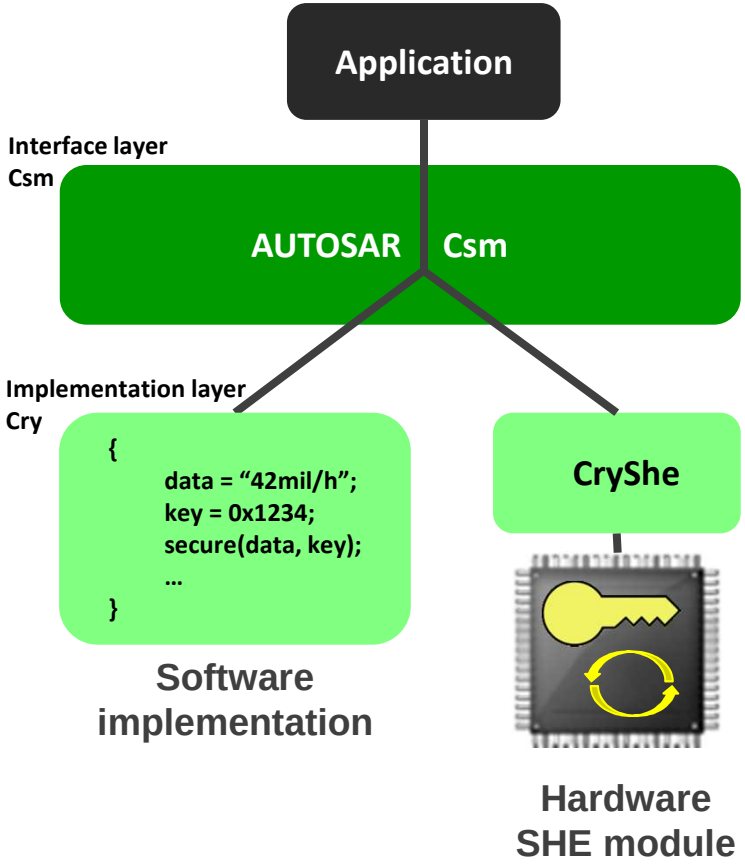
- AUTOSAR defines two sets of crypto routines
  - Crypto **service** manager (CSM/CRY)
  - Crypto abstraction **library** (CAL/CPL)
- Both AUTOSAR specifications subdivide crypto modules into two layers
  - Interface layer
    SERVICES
  - Implementation layer
    PRIMITIVES



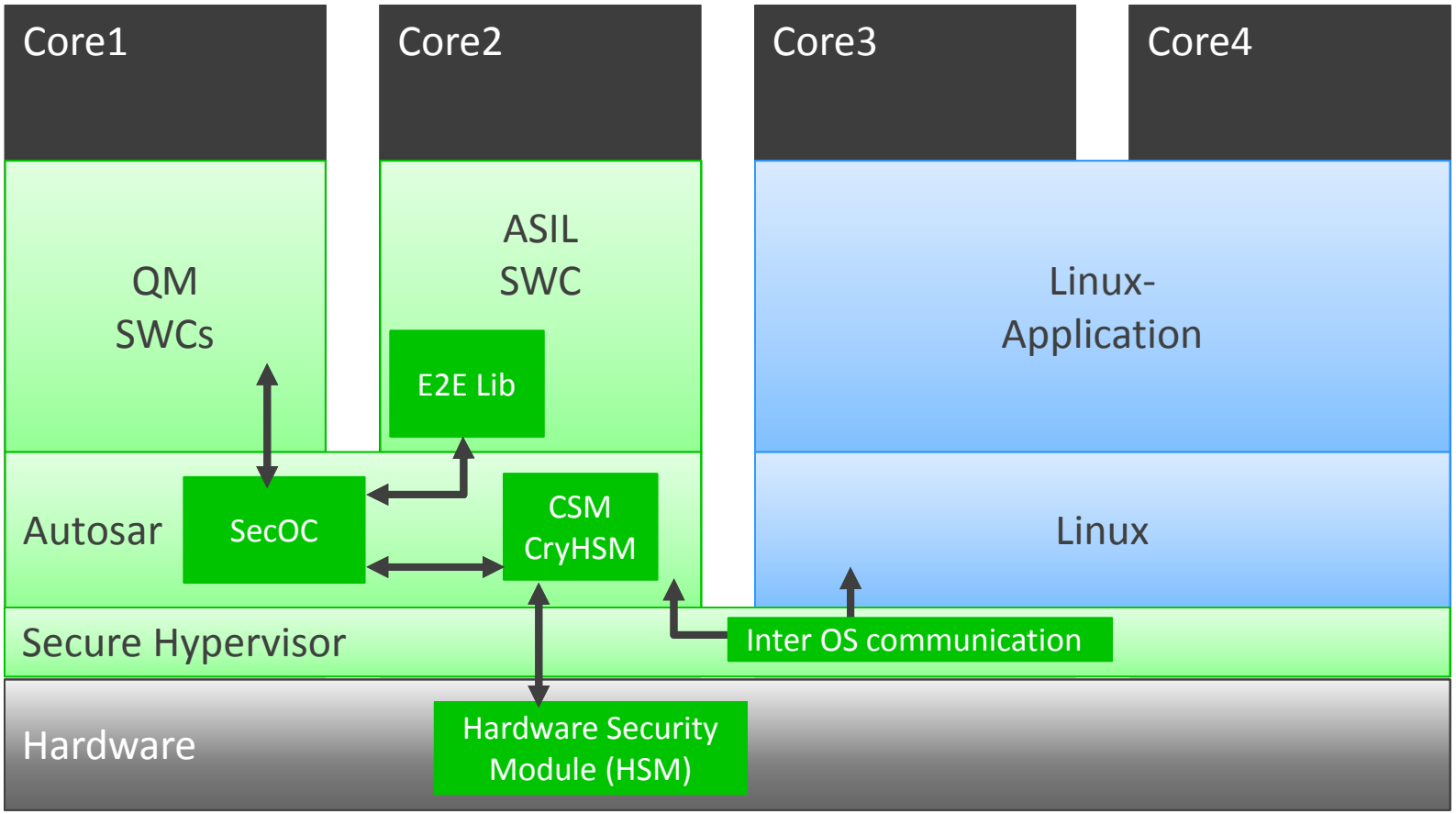
- Only the interface layer is properly specified in AUTOSAR
  - This layer is completely standardized
- Contents of the implementation layer are left open for customer options
  - This layer implements customer specific solutions with a standardized interface to the interface layer

Use case examples – Secure Hardware Extension

- Attainable security level in software is limited.
- New automotive ECUs offer a 'Secure Hardware Extension (SHE)' module.
  - E.g. freescale Bolero 3M/Calypso, Infineon TC179x, Fujitsu Atlas-L family, Renesas RH850
- EB integrates the new hardware module with standard software.
- Development of drivers for SHE.
- Integration with AUTOSAR cryptographic module (Csm/CryShe).
- Tool driven configuration.
- We enable customers to easily switch between cryptographic routines in software and hardware.



Outlook: Hypervisor setup



Agenda

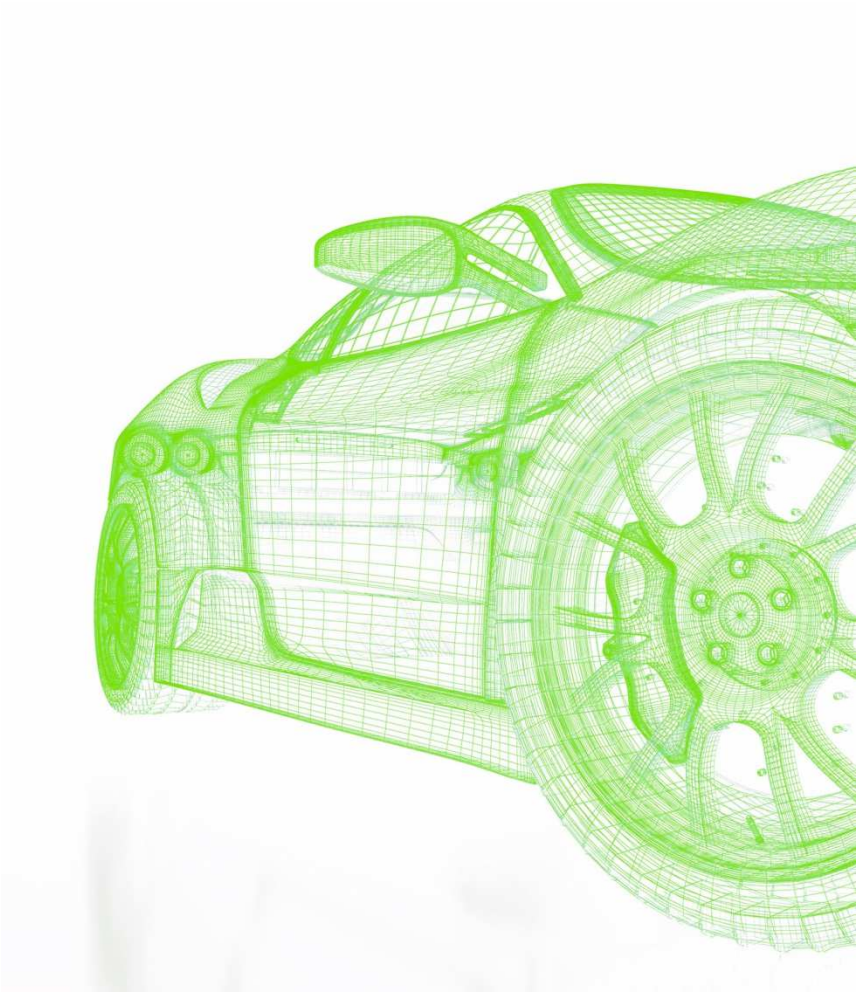
"No safety without security and vice versa"

Established Safety Concepts

Safety Analysis Methods for Security Analysis

Secure Software Architecture Extensions

Summary



Summary

- **Extend safety analyses with security aspects**
  - Safety and security complement each other
  - Employed methods are quite similar
- **Safe and secure software architectures**
  - Use partitioning mechanisms for protection mechanisms
  - Use secure authentication and integrity mechanisms for safe communication
- **Hypervisors combines two worlds**
  - Access to board net via AUTOSAR
  - Applications on e.g. Linux
  - Protected communication through Firewall

Thank you!

Elektrobit

Rudolf.Grave@elektrobit.com
automotive.elektrobit.com