TeleTrusT
Pioneers in IT security.

Informationstag "Das Automobil als IT-Sicherheitsfall"

Berlin, 11.05.2012

Safety and Security for Automotive using Microkernel Technology

Dr.-Ing. Matthias Gerlach
OpenSynergy

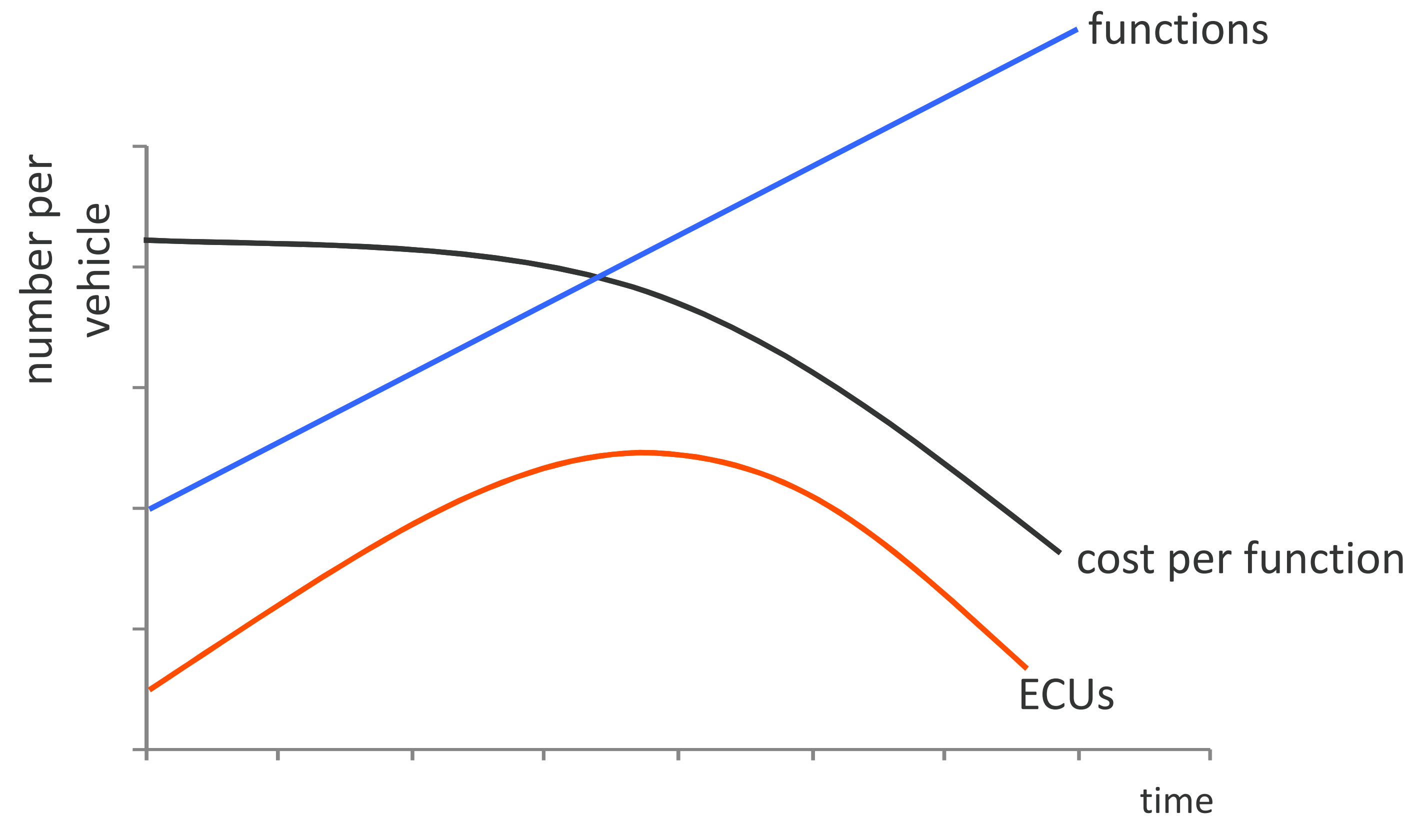
Two Birds with One Stone

Safety and Security for Automotive using Microkernel Technology

Dr.-Ing. Matthias Gerlach (OpenSynergy)

OPENSYNERGY

Trend in Automotive: Integration



Pricing pressure requires multiple functions integrated on a single ECU

OPENSYNERGY

Trend in Automotive: Consumer Electronics Integration

Consumer electronics

Automotive

Driver assistance

Body control

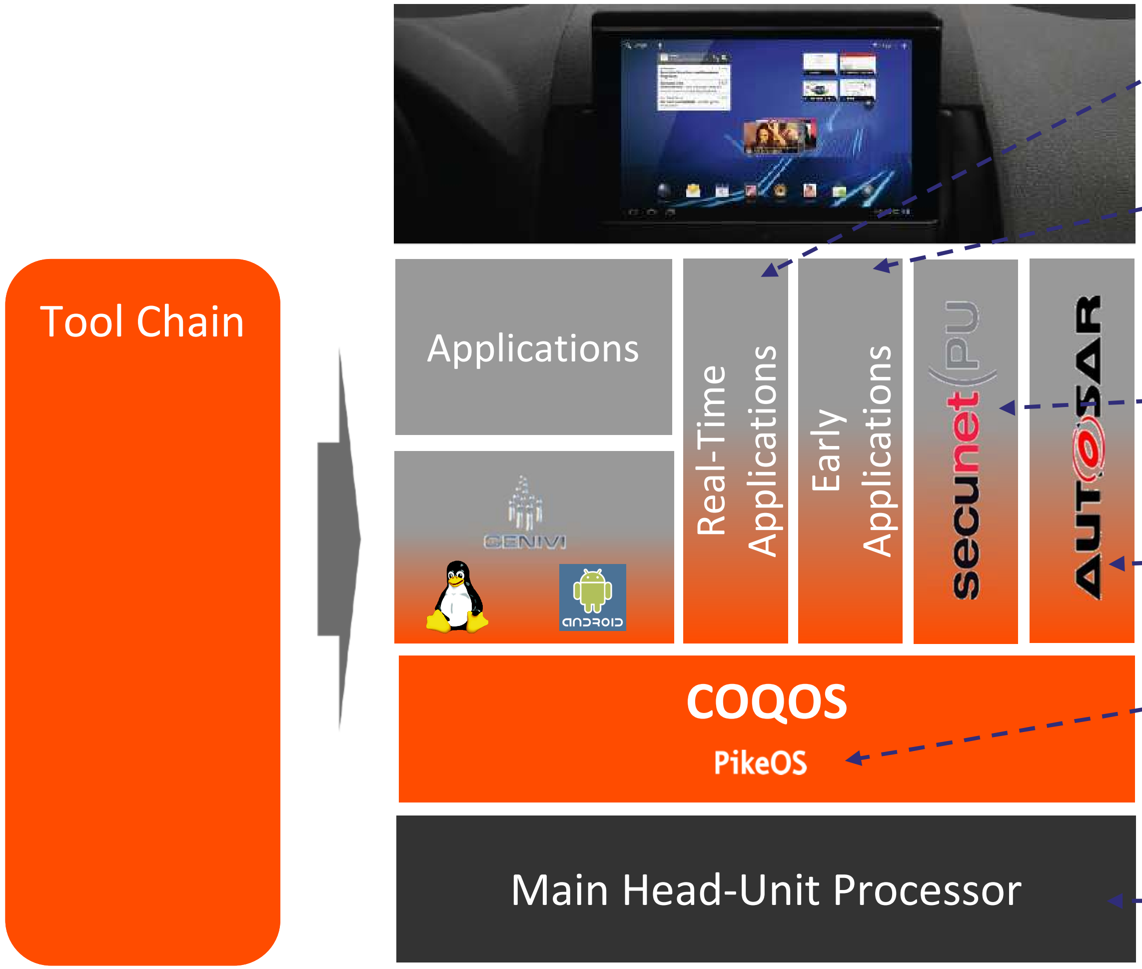
Chassis control

Comfort electronic

OPENSYNERGY

COQOS for Linux and Android Head-Units

COQOS is the best way to take advantage of Linux and still satisfy automotive requirements!



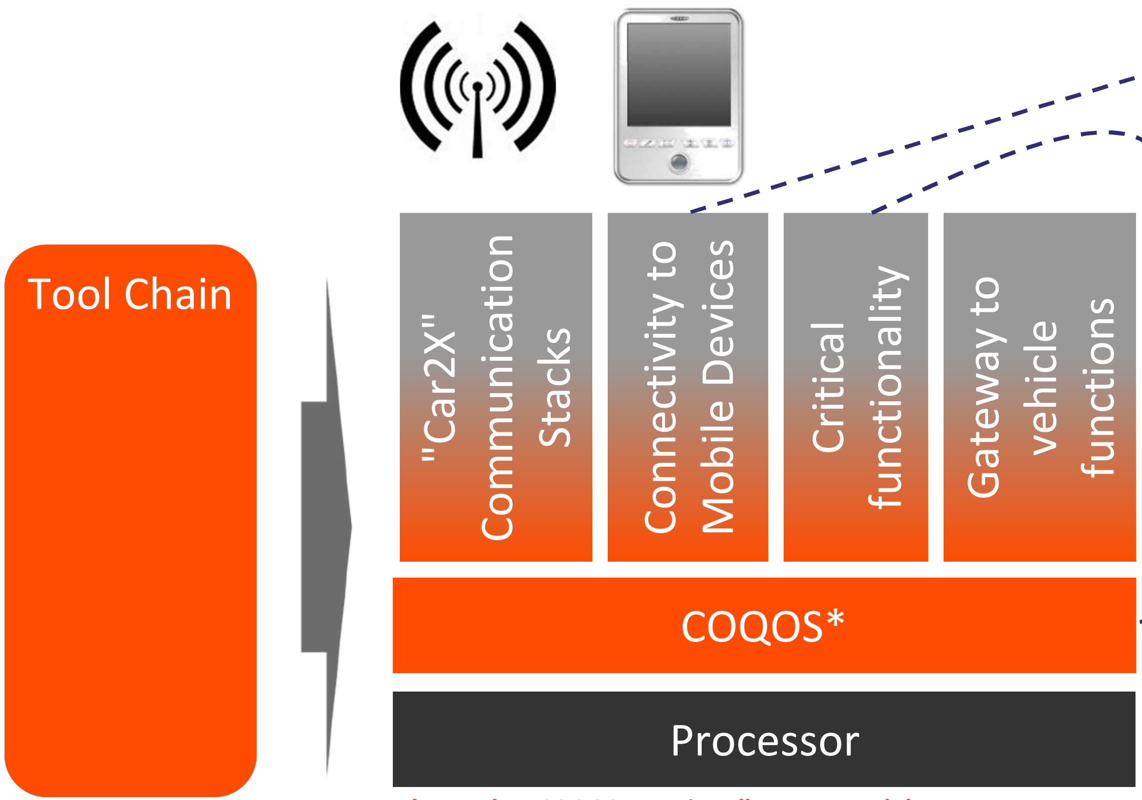
The product COQOS contains all orange modules

- **Real Time:** applications with high timing requirements run independently from Linux
- **Fast Boot**: applications are available extremely quickly, well before Linux has been started (e.g. Rear View)
- **Firewall**: all communication towards the vehicle network is controlled via policy
- **AUTOSAR**: plug in AUTOSAR basic software and SW-Cs
- **Certified**: the PikeOS microkernel has been certified according to the highest safety and security standards
- **Cost Reduction:** COQOS integrates diverse functionality on a single processor

OPENSYNERGY

COQOS for Connectivity

COQOS is well-suited to build devices connecting the vehicle systems with the outside world!



The product COQOS contains all orange modules

* Based on PikeOS Micro-Kernel from SYSGO AG

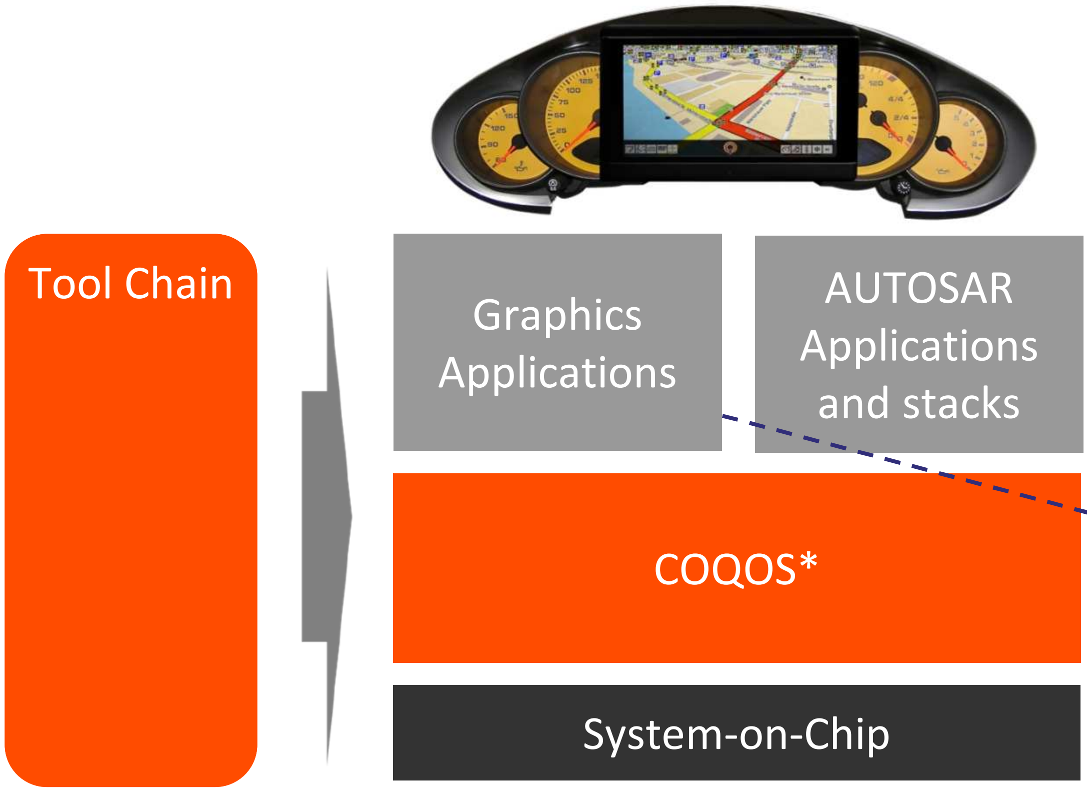
COQOS enables:

- **Mobile Device Compatibility**: through integration of industry-standard communication stacks
- **Critical applications** such as E-Call can run directly on COQOS
- **Firewall:** separation of the different communication stacks is the basis for improved security

OPENSYNERGY

COQOS for Instrument Clusters

COQOS makes it possible to run high-end displays and AUTOSAR on a single System-on-chip:



The product COQOS contains all orange modules

* Based on PikeOS Micro-Kernel from SYSGO AG

COQOS enables:

- **AUTOSAR:** Full reuse of AUTOSAR applications and basic-software (communication stacks)
- **Graphics:** Access to the graphics capabilities of the SOC

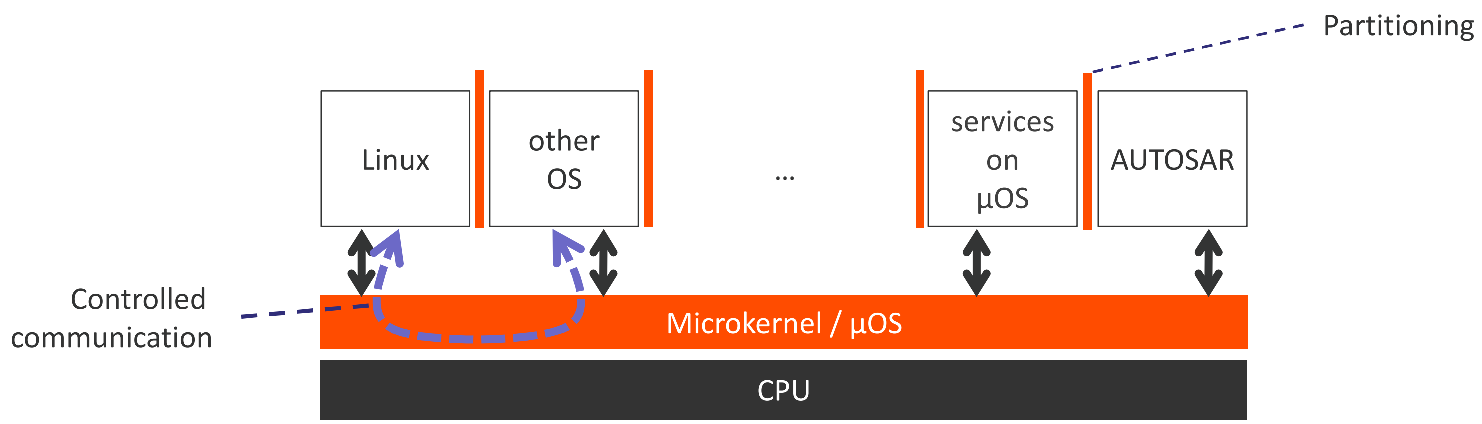
OPENSYNERGY

Microkernel Technology Primer

„In computer science, a microkernel is the near-minimum amount of software that can provide the mechanisms needed to implement an operating system (OS)" (Wikipedia)

- ~ 10K Lines of Code
- Mechanisms include:
  - Time partitioning (Scheduling) and
  - Space partitioning (Access to memory)
  - Inter process communication

- Microkernel technology successfully used in Aerospace
  - E.g., PikeOS Microkernel by SYSGO for A380
  - Intgrated Modular Avionics



OPENSYNERGY

Requirements Summary

ID_001: It shall be possible to integrate several functions over one piece of HW.

ID_002: The Linux operating system shall be supported for Infotainment applications.

ID_003: Some functions shall be developed using AUTOSAR methodology and architecture

ID_004: The ECU shall startup selected components from cold-start below 150ms

ID_005: The ECU shall be safe …

ID_006: The ECU shall be secure …

- ☑ *Integration*
- ☑ *Linux*
- ☑ *AUTOSAR*
- ☑ *Early Apps*
- ☑ *Real-time*
- ☑ *Safety*
- ☑ *Security*

OPENSYNERGY

Safety and Security defined …

Functional safety is „the state in which a vehicle function does not cause any intolerable endangering states" (ISO 26262)

"Security is concerned with the protection of assets" (Common Criteria)

against malicious attackers.

OPENSYNERGY

Future Head Unit

?

How do I know my system is safe and secure?

- ☑ *Integration*
- ☑ *Linux*
- ☑ *AUTOSAR*
- ☑ *Early Apps*
- ☑ *Real-time*
- ☑ *Safety*
- ☑ *Security*

OPENSYNERGY

Refining Safety and Security Requirements

Standards (relevant for Automotive)

- ISO 26262 – Road Vehicles – Functional Safety
- Common Criteria for Information Technology Security Evaluation (Common Criteria)

Standards help assessing whether my system is safe and secure.

Assumptions

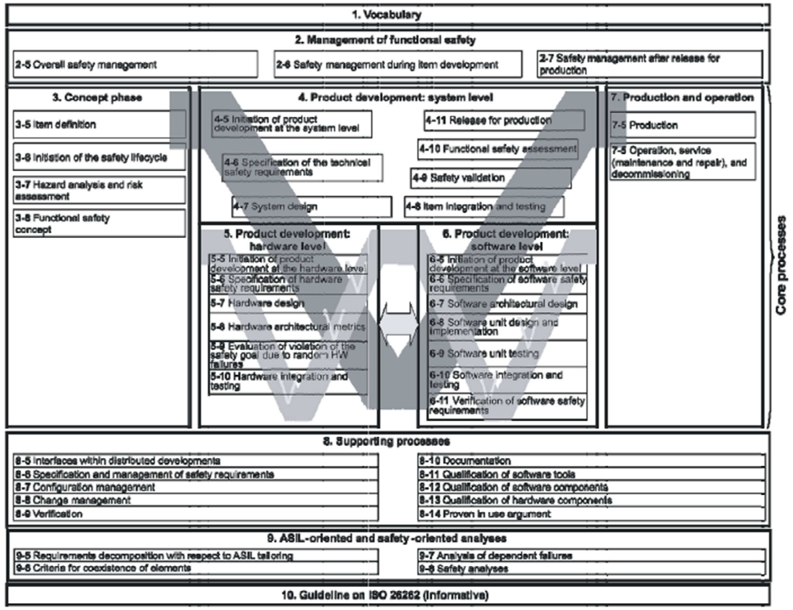
- Focus on key system component of integrated ECU, the Microkernel
- Can define common „functional requirements“
- Standards define a „methods and processes framework“ to implement and verify requirements and define HOW SAFE/SECURE the system is

OPENSYNERGY

ISO 26262 – „Functional Safety for Road Vehicles“

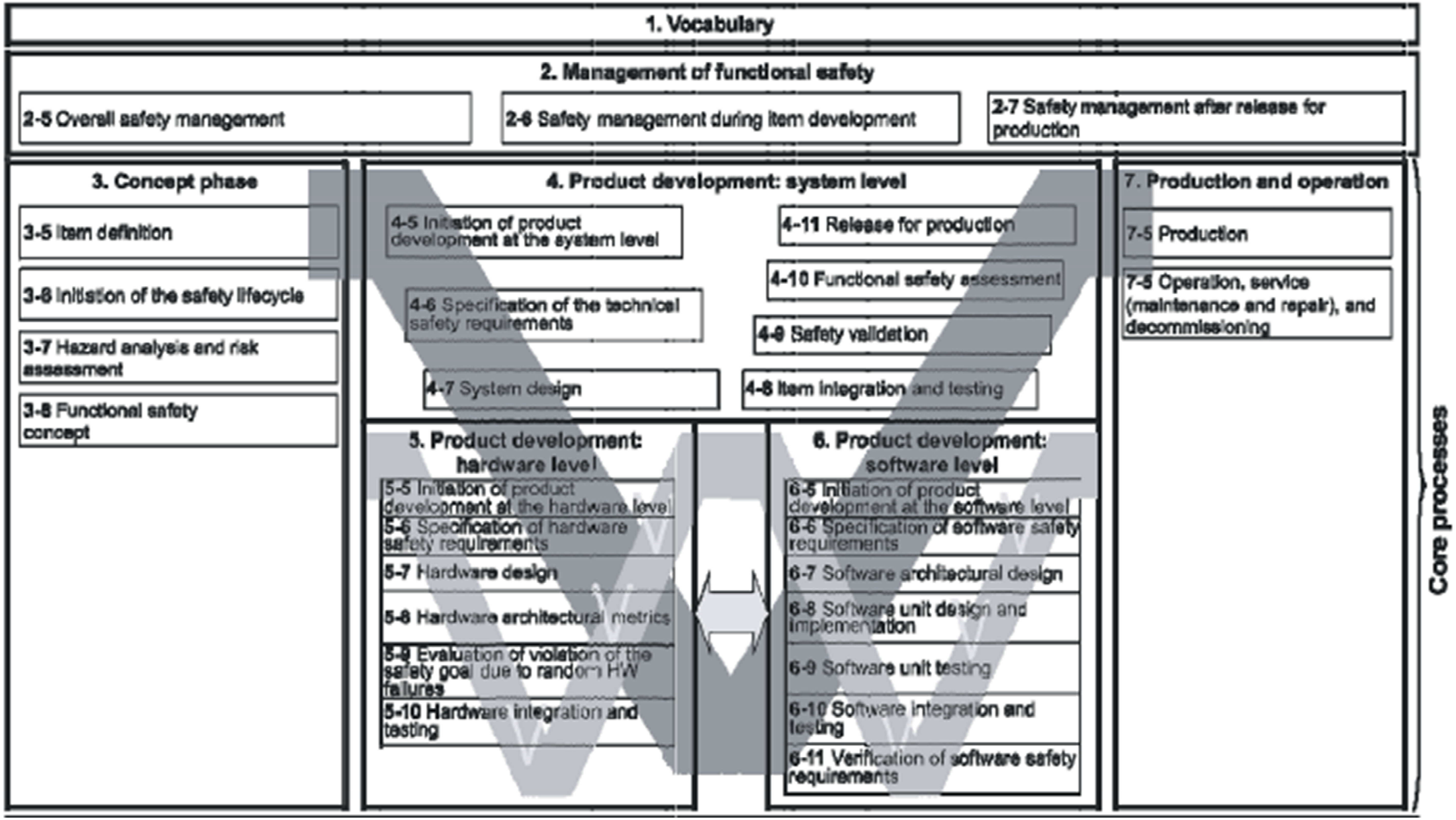
- Based on IEC 61508, with automotive specific adaptations
- Draft International Standard, 2009
- Published Norm expected in 2011

- Specific for series production cars
- Represents „state of the art“ for safe produ
- Covers all aspects of product lifecycle (Syste
- Specifies concrete measures



OPENSYNERGY

ISO 26262: V-Model Approach

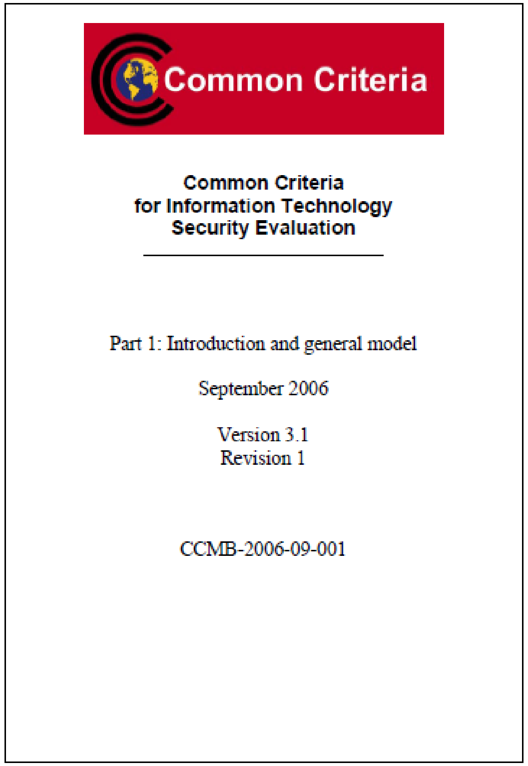


Common Criteria (for Information Technology Security Evaluation)

- Published as ISO/IEC 15408:2005
- Common, international standard for secure informa
- Dates back to activities in the 1990ies

- Parts:
  - 1: General Model
  - 2: Security Functional Requirements
  - 3: Security Assurance Requirements

- Protection Profile for Mikrokernel exists


Common Criteria

Common Criteria
for Information Technology
Security Evaluation

Part 1: Introduction and general model

September 2006

Version 3.1
Revision 1

CCMB-2006-09-001

OPENSYNERGY

Approach in Common Criteria

Item definition

- System and Environment
- Thread analsiys

Requirements

- Security Objectives
- Security Functional Requirments

Implementation

- Security Assurance Requirements

Validation

- Security Assurance Requirements
- Traceability from Requirements to Implementation

OPENSYNERGY

Common Criteria: Protection Profile

Protection Profile

- Intended to describe a TOE (Target of Evaluation) type
- Abstracts from concrete implementation of TOE

Example: Seperation Kernel Protection Profile (SKPP)

- Profile for Seperation Microkernels
- Used for existing Mikrokernels, such as PikeOS by SYSGO

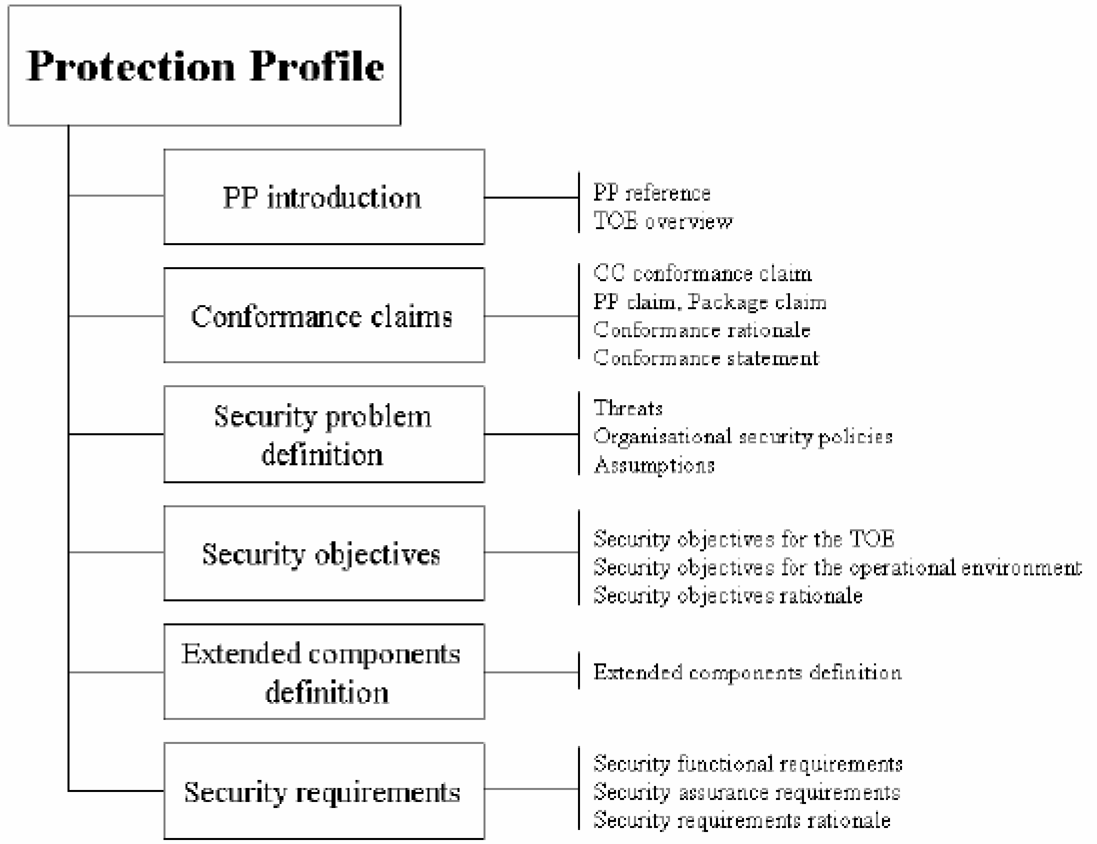
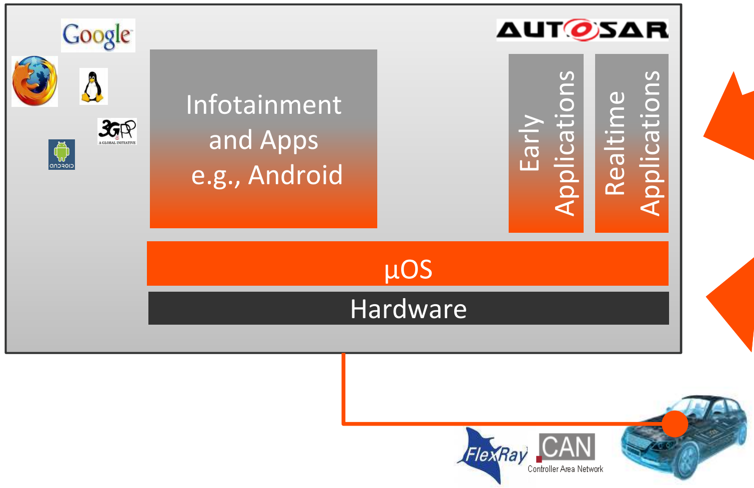


Figure 9 - Protection Profile contents

[Source: Common Criteria Part 1]

Safety – Refined



Correct scheduling to ensure real-time properties

Partitioning / Isolation to prevent a failure of one partition to propagate

Safe communication between components

These requirements can also be found in ISO 26262!

OPENSYNERGY

Security – Refined

5.6.2 Explicit: Predictable Resource Utilization by the TSF (FRU_PRU_EXP.1)

5.6.2.1 **Explicit: TSF Predictable Resource Utilization (FRU_PRU_EXP.1.1)**

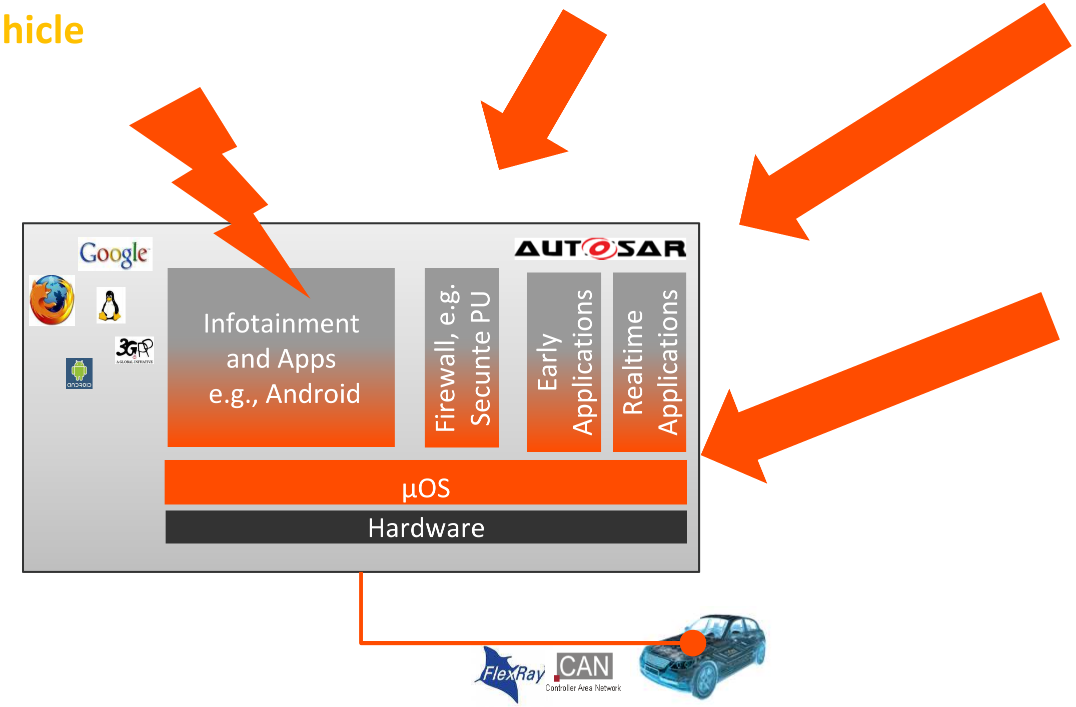
FRU_PRU_EXP.1.1 The TSF shall exhibit predictable and bounded execution behavior with respect to its usage of processor time and memory resources.

Application Note: The TOE developer is to document the expectations for memory and processor usage by the TSF in completing ADV_ARC_EXP.1.5C.

Control communication between Linux and other partitions (Authorizations on higher level)

Attacker may use external interfaces to manipulate vehicle

Correct scheduling to ensure real-time properties



Partitioning / Isolation

5.5.11 Domain Separation (FPT_SEP)

5.5.11.1 Complete Reference Monitor (FPT_SEP.3)

FPT_SEP.3.1 **Refinement:** The unisolated portion of the TSF shall **use hardware mechanisms to** maintain a security domain for its own execution that protects **the code and data of the unisolated portion of the TSF** from interference and tampering by untrusted subjects. 14

Application Note: Examples of hardware mechanisms that might be used to support a protected security domain for the execution of the TSF include: privilege bits; rings; hardware mechanisms that support controlled entry points to domains; and a variety of memory management features.

FPT_SEP.3.2 The TSF shall enforce separation between the security domains of subjects in the TSC.

FPT_SEP.3.3 **Refinement:** The TSF shall maintain the part of the TSF that enforces the information flow control SFPs in a security domain for its own execution that protects **that part of the TSF** from interference and tampering by the remainder of the TSF and by subjects untrusted with respect to the TSP. 15

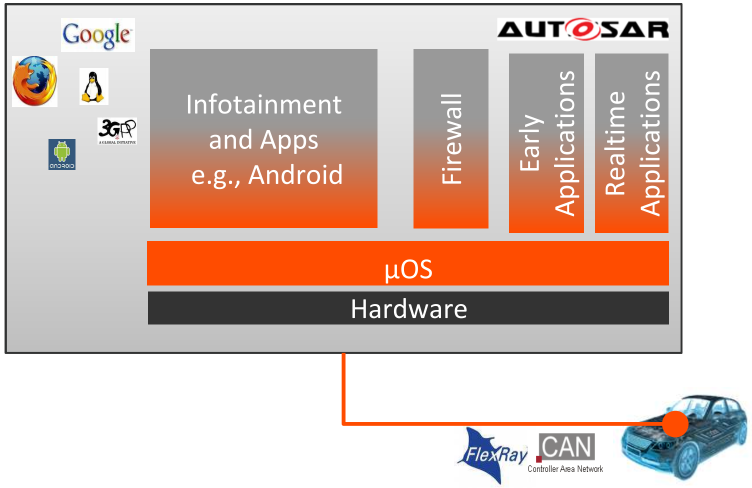
[Source: Seperation Kernel Protection Profile]

OPENSYNERGY

Common Requirements Summary for Safety and Security

Architecture

- „Future Head Unit“



Requirements

- System partitioning
- Safe communication
- Monitoring of components (and transition to safe/secure state)
- Timing and synchronization of components

OPENSYNERGY

Comparability

Development process similar

- Requirements
  → Architecture/Design
    → Implementation
      → Testing against list of Requirements

**Evaluation assurance Levels (EAL) (Common Criteria)
vs. Automotive Safety Integrity Level (ASIL) (ISO 26262)**

- EAL describes development rigour
- ASIL proportional to criticality of component (severity, exposure, controllability tuple) BUT implies development rigour (by means of subsequent recommondations in development process)

OPENSYNERGY

Example (Development, Design)

ISO 26262 Part 6

Methods and Measures		According to req.	ASIL A	ASIL B	ASIL C	ASIL D
1b	Semi-formal notations for software architectural design	6.4.1	+	++	++	++
1c	Formal notations for software architectural design	6.4.1	+	+	++	++
2	Computer-aided tools for software architectural design	6.4.1	+	+	++	++
3	Guidelines for the application of the selected methods and measures for software architectural design	6.4.1	+	++	++	++

NOTE: The software architectural design needs to be described completely and consistently by an appropriate combination of methods 1x).

Table 6.1 — Methods and measures for software architectural design

6.4.2 A software architectural design shall be developed in compliance with design guidelines that shall follow the design principles listed in table 6.2.

Methods and Measures		According to req.	ASIL A	ASIL B	ASIL C	ASIL D
1	Restricted size of software components	6.4.2	++	++	++	++
2	Restricted size of interfaces	6.4.2	+	+	+	++
3	High cohesion within software components	6.4.2	+	++	++	++
4	Limitation of coupling between software components	6.4.2	+	++	++	++
5	Restricted use of interrupts	6.4.2	+	+	+	++

NOTE 1: Method 4 addresses the limitation of the external coupling of software components.

NOTE 2: For these methods appropriate metrics are to be used.

Table 6.2 — Design principles for software architectural design

Common Criteral, EAL 5, semiformally designed and tested

Assurance Class	Assurance components
ADV: Development	ADV_ARC.1 Security architecture description
	ADV_FSP.5 Complete semi-formal functional specification with additional error information
	ADV_IMP.1 Implementation representation of the TSF
	ADV_INT.2 Well-structured internals
	ADV_TDS.4 Semiformal modular design

OPENSYNERGY

Example 2 (Testing)

Methods and Measures		According to req.	ASIL A	ASIL B	ASIL C	ASIL D
1	Statement coverage	8.4.3	++	++	+	+
2	Decision coverage	8.4.3	+	+	++	+
3	MC/DC (Modified Condition Decision Coverage), conditions affecting the decision	8.4.3	+	+	+	++
4	Model coverage	8.4.3	++	++	++	++

NOTE 1: Degrees of coverage demanded in item 1 have to be determined with appropriate tools on source code level. The objective is source code coverage of 100%. As this is not always possible in practice deviations are to be analysed and justified. Complementary analytical measures e.g. inspections, have to be executed for not covered source code.

NOTE 2: If in case of model based development software unit testing is substituted by tests on model level instead of the measures in items 1, 2, 3 and 4 analogous model coverage metrics have to be used.

NOTE 3: For structural tests measuring the degree of coverage usually instrumented code is used. There, it has to be shown that instrumentation of the source code or object code will not lead to functional changes. This can be done for example by repeating the tests with non-instrumented code.

NOTE 4: When programming in a language that implements short circuit operators, (e.g. in "C" language), "MC/DC" and "Condition" + "Decision" coverage are equivalent.

Table 8.3 — Methods and measures for structural software unit testing

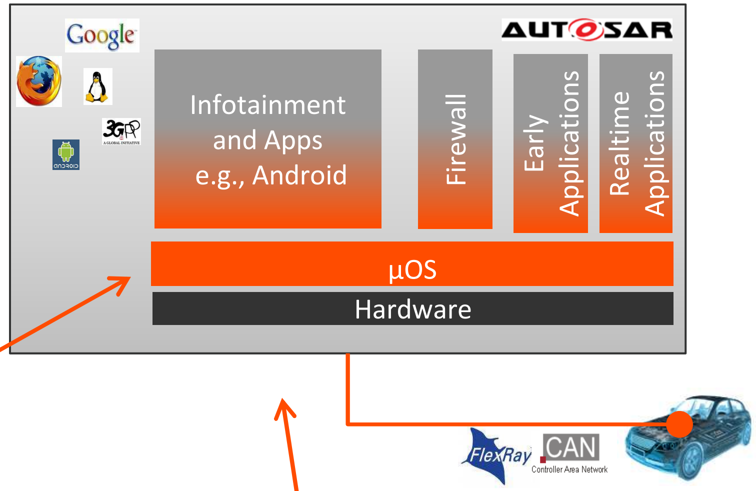
Common Criteral, EAL 5, semiformally designed and tested

Methods and Measures		According to req.	ASIL A	ASIL B	ASIL C	ASIL D
	Functional tests	8.4.3	See table 8.2			
	Structural tests	8.4.3	See table 8.3			
3	Resource usage test	8.4.3	+	+	+	++
4	Back-to-back test between simulation model and code	8.4.3	+	+	++	++

ATE: Tests	ATE_COV.2 Analysis of coverage
	ATE_DPT.3 Testing: modular design
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing - sample

OPENSYNERGY

Two Birds …



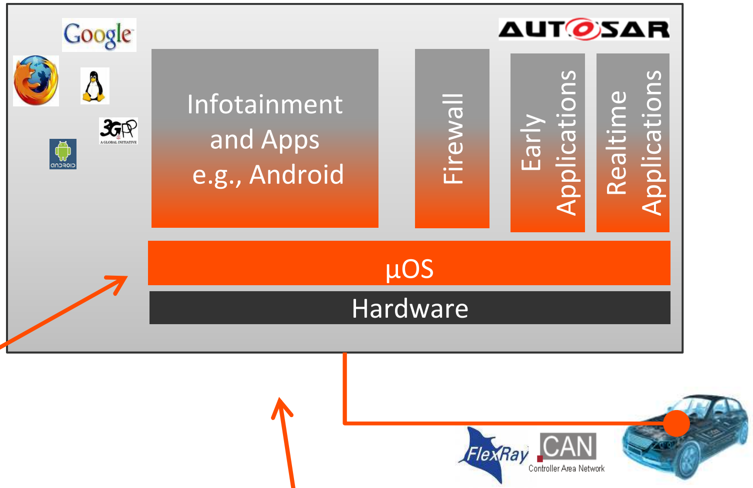
Microkernel certified / certifiable according to IEC 61508 / Common Criteria SKPP

AND

Make Use Core Microkernel Properties (e.g., separation) for System Design

OPENSYNERGY

... One Stone !!



Microkernel certified / certifiable according to IEC 61508 / Common Criteria SKPP

AND

Make Use Core Microkernel Properties (e.g., separation) for System Design

OPENSYNERGY

Conclusion

- Microkernel-based Systems address both safety and security issues
- Standards provide indication about level of security / safety
- Approach to develop safe/secure software is similar for both standards
- Common requirements for safety and security concerning the microkernel → „double insurance“

☑ *Integration*
☑ *Linux*
☑ *AUTOSAR*
☑ *Early Apps*
☑ *Real-time*
☑ *Safety*
☑ *Security*

COQOS
the core

OPENSYNERGY

OpenSynergy GmbH
Rotherstraße 20
D-10245 Berlin
Germany

tel +49 30 / 60 98 54 0 - 0
fax +49 30 / 60 98 54 0 - 99
mail info@opensynergy.com

OPENSYNERGY