EASIS

Electronic Architecture and System Engineering
for Integrated Safety Systems

Dependability Services in the EASIS Software Platform

Martin Hiller
Volvo Technology Corporation
martin.hiller@volvo.com

Workshop on Architecting Dependable Systems
June 27, 2006
Philadelphia, USA

VOLVO

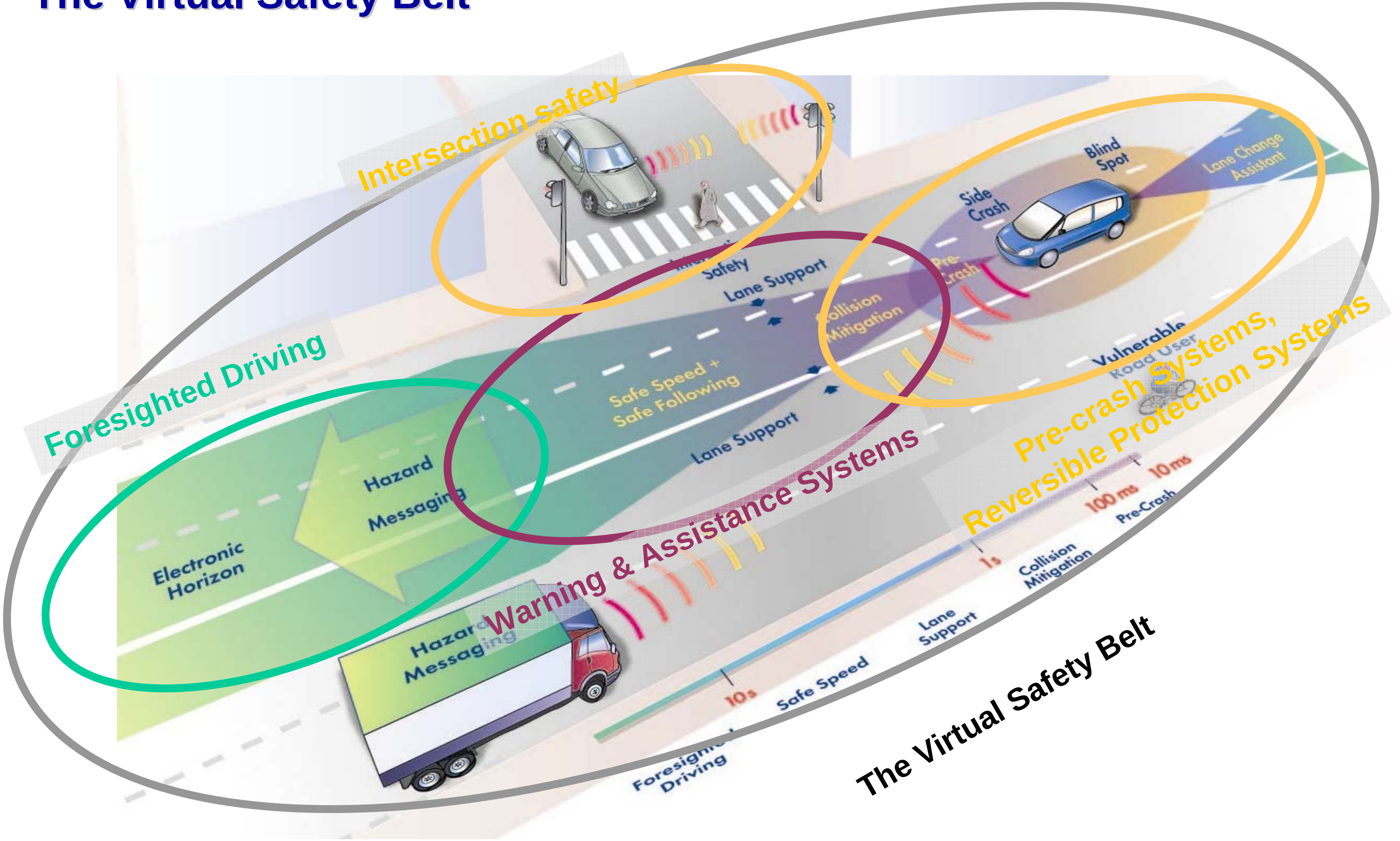
Information Society Technologies

Safety

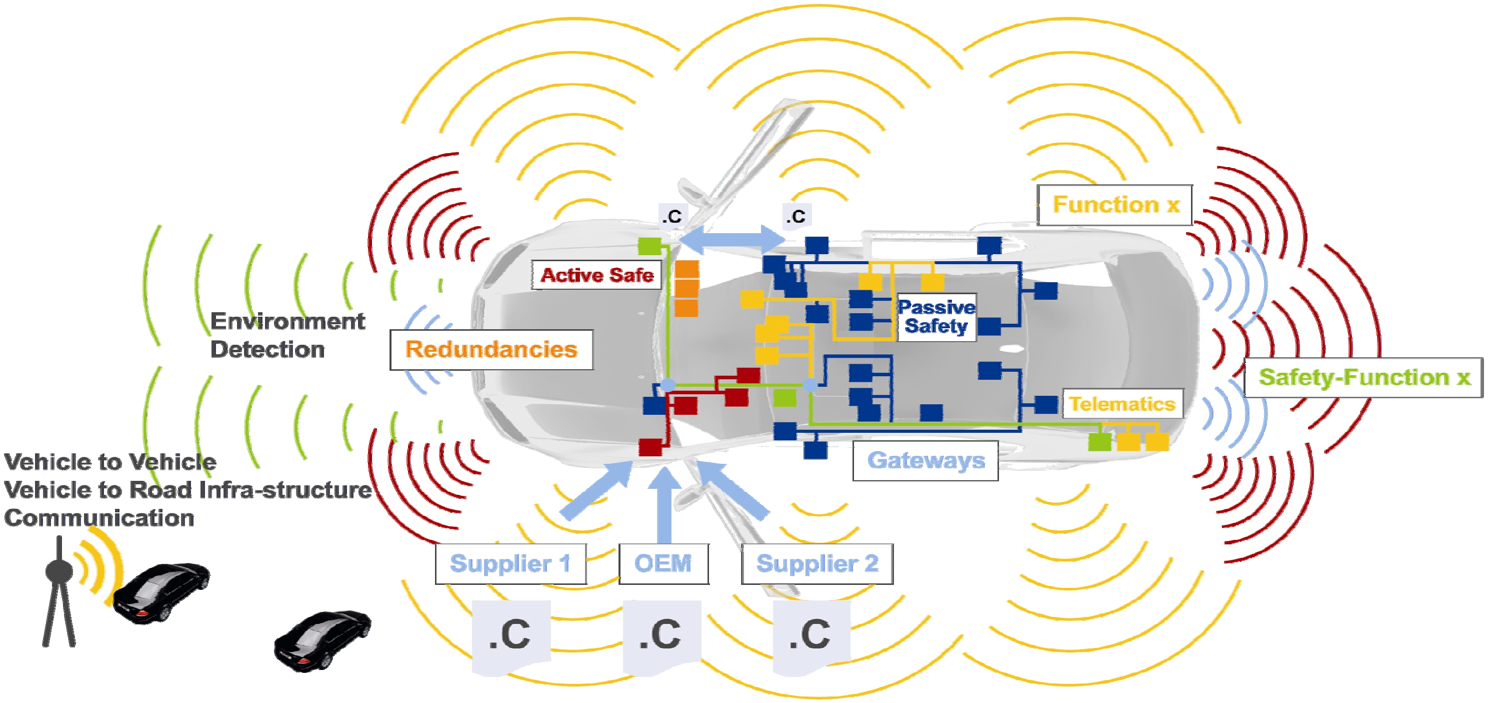
Outline

- Background
  - > "The Virtual Safety Belt"
  - > Project data
  - > Related projects
  - > Results overview
- Software platform
  - > Layered architecture
  - > Fault management framework
  - > Dependability support
  - > Security support

The Virtual Safety Belt



Issues for integrated safety systems



Challenges:

- > Integration of domain (cabin, chassis, powertrain, …) overlapping safety functions with high dependability
- > Handling of high system complexity
- > Integration and multi-usage of environment sensing
- > Integration of telematics services for safety systems

Project data

Coordinator:	DaimlerChrysler (Dr. Vera Lauer)
Starting Date:	01.01.2004
Ending Date:	28.02.2007
Budget Total/Funding:	9,4 M€ / 5 M€
Web site:	www.easis.org

22 partners

OEM's	CENTRO RICERCHE FIAT DAF DAIMLERCHRYSLER OPEL PSA PEUGEOT CITROËN RENAULT VOLVO
Automotive suppliers	BOSCH Continental TEVES freescale semiconductor (Launched by Motorola) LEAR CORPORATION TRW Valeo ZF
Tool suppliers	DECOMSYS (Dependable Computer Systems · Hardware und Software Entwicklung GmbH) dSPACE ETAS vector
Research institutes	MIRA OFFIS UNIVERSITÄT DUISBURG ESSEN

Related projects (Integrated Safety Programme defined by EUCAR*)

Ongoing Projects

Systems for accident prevention

“PReVENT”

DaimlerChrysler

Common, agreed adaptive HMI interface

“AIDE”

AB Volvo

Systems for passenger protection

“APROSYS”

TNO

Common E/E architecture for vehicles

“EASIS”

DaimlerChrysler

Systems for post accident rescue

“GST”

ERTICO

Recently Started

Connecting intelligent vehicle and infrastructure for enhanced SAFETY

“SAFESPOT”

CRF

Connecting intelligent vehicle and infrastructure for enhanced EFFICIENCY

“CVIS”

ERTICO

Methodology for accident causation analysis

“TRACE”

Renault

Vehicle-to-Vulnerable road user cooperative technologies to improve safety
“Watch-Over”

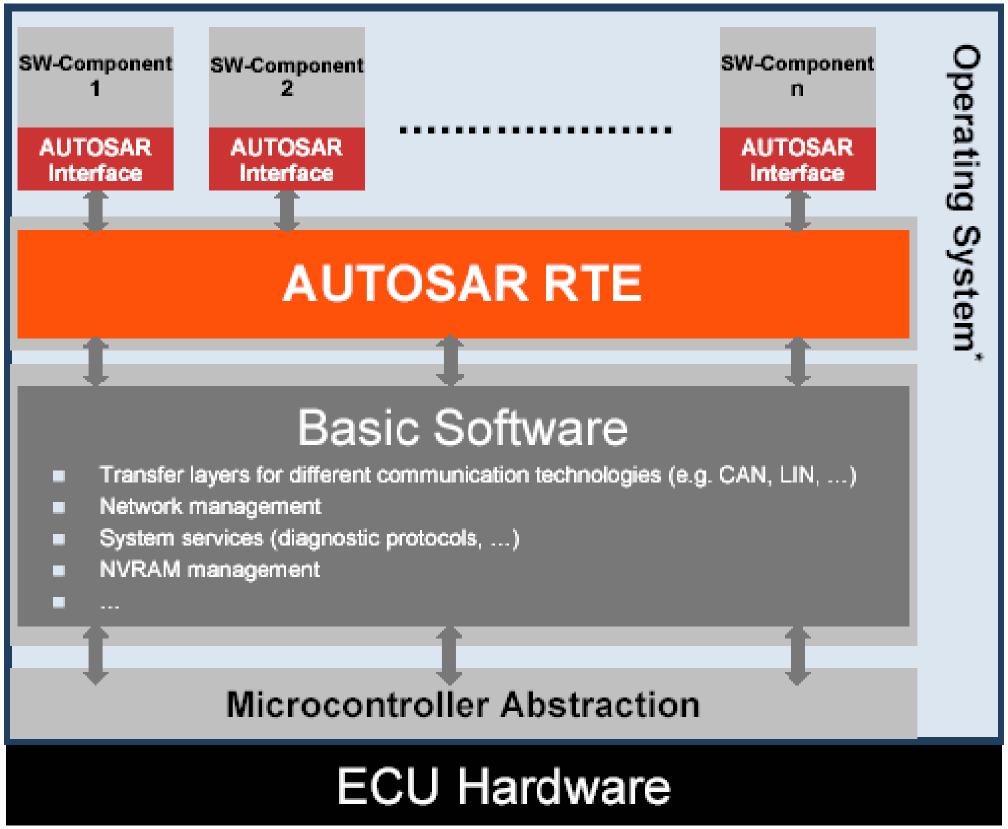
CRF

Software for Traffic Efficiency and Safety
“ATESST”

AB Volvo

***European Council on Automotive R&D**

AUTOSAR (AUTomotive Open System ARchitecture)

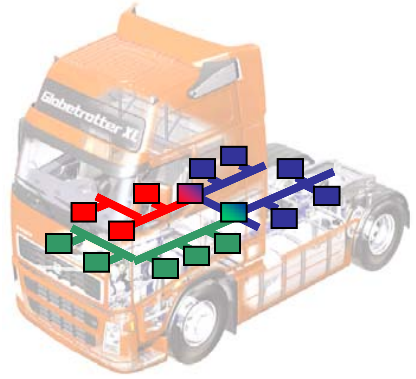


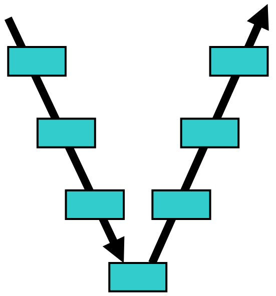
- AUTOSAR
  - > Standardized, openly disclosed interfaces
  - > HW independent SW layer
  - > Transferability of functions
- AUTOSAR RTE
  - > By specifying interfaces and their communication mechanisms, the applications are decoupled from the underlying HW and basic SW, enabling the realization of Standard Library Functions
- Consortium of over 100 members (and growing)
  - > Partners from Europe, US, Japan
  - > No public funding

Check www.autosar.org for more information.

Expected final results

EASIS will provide enabling technologies for the introduction of future integrated safety systems





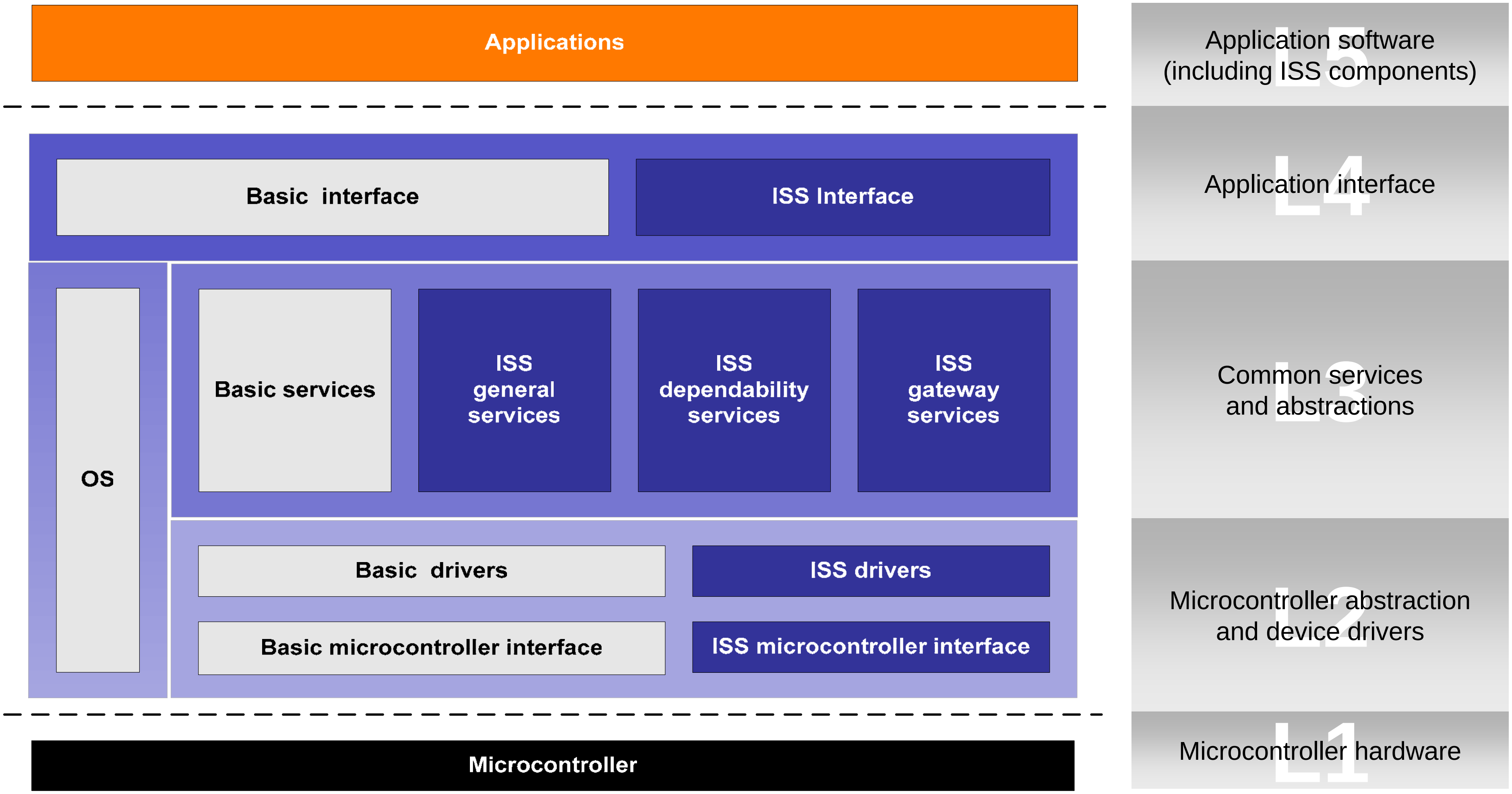
> **Software platform** providing common services for cooperation between safety systems

> **Dependable electronics hardware infrastructure**, which supports the requirements of these systems in a cost effective manner

> **Methods and techniques** for handling critical dependability-related parts of the development lifecycle

> **Engineering process and tool chain** supporting the development of cooperating safety systems

Overall software topology



Applications

Basic interface

ISS Interface

OS

Basic services

ISS general services

ISS dependability services

ISS gateway services

Basic drivers

ISS drivers

Basic microcontroller interface

ISS microcontroller interface

Microcontroller

L5 Application software (including ISS components)

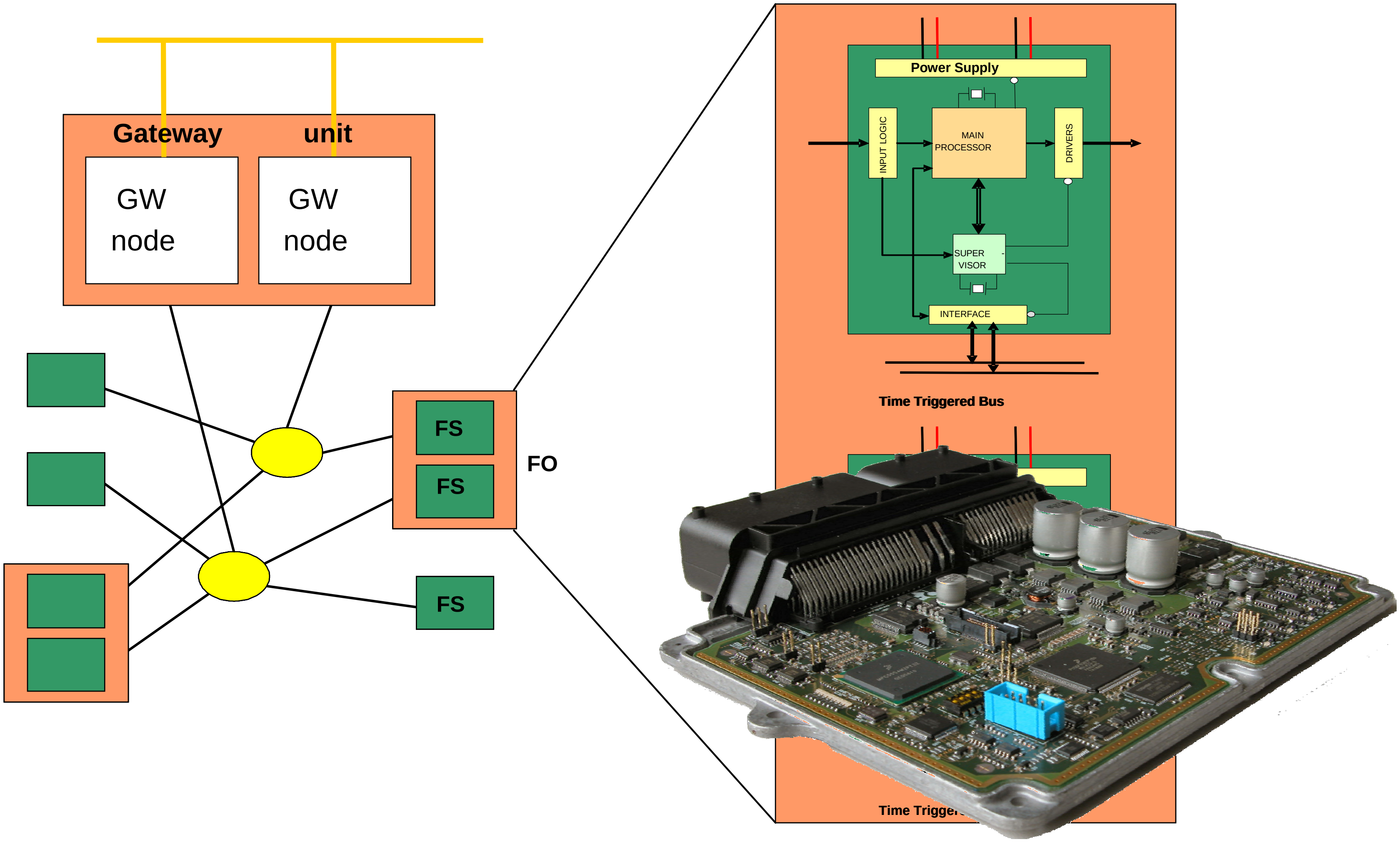
L4 Application interface

L3 Common services and abstractions

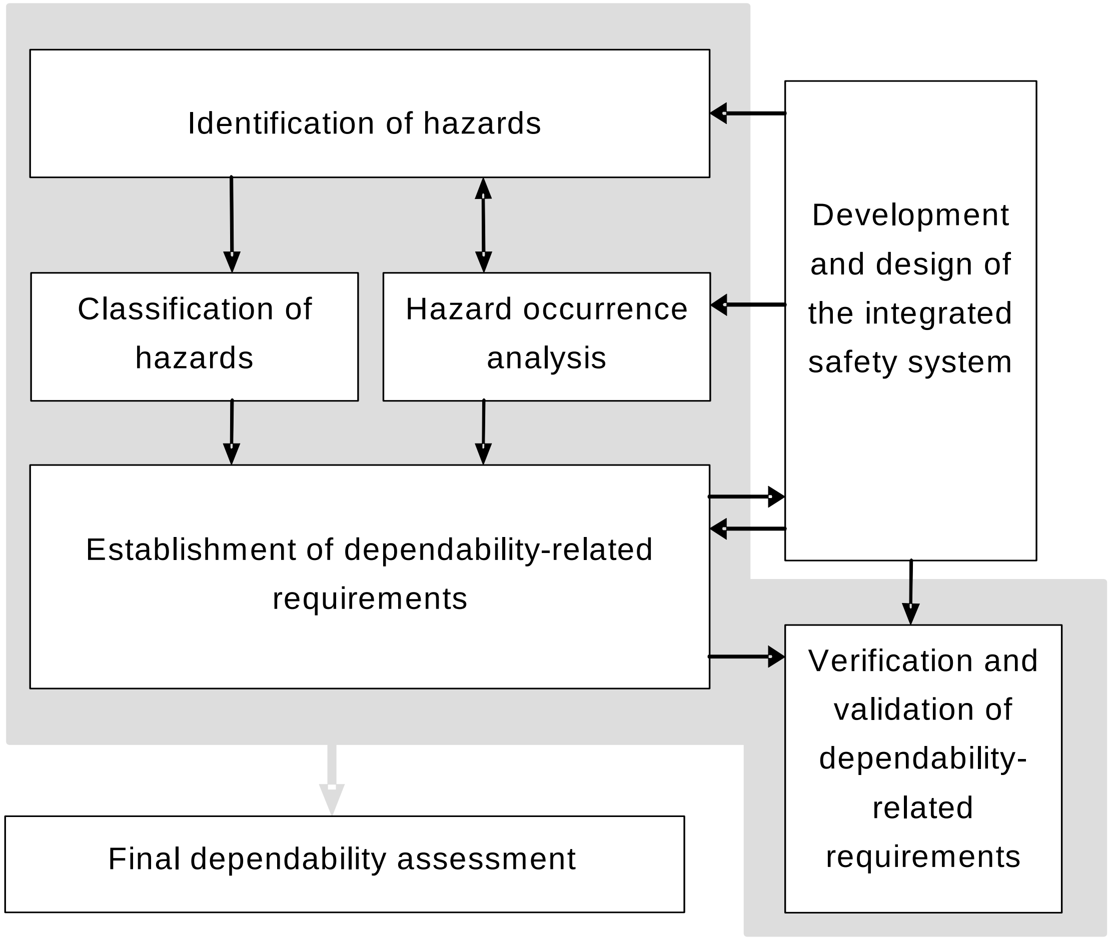
L2 Microcontroller abstraction and device drivers

L1 Microcontroller hardware

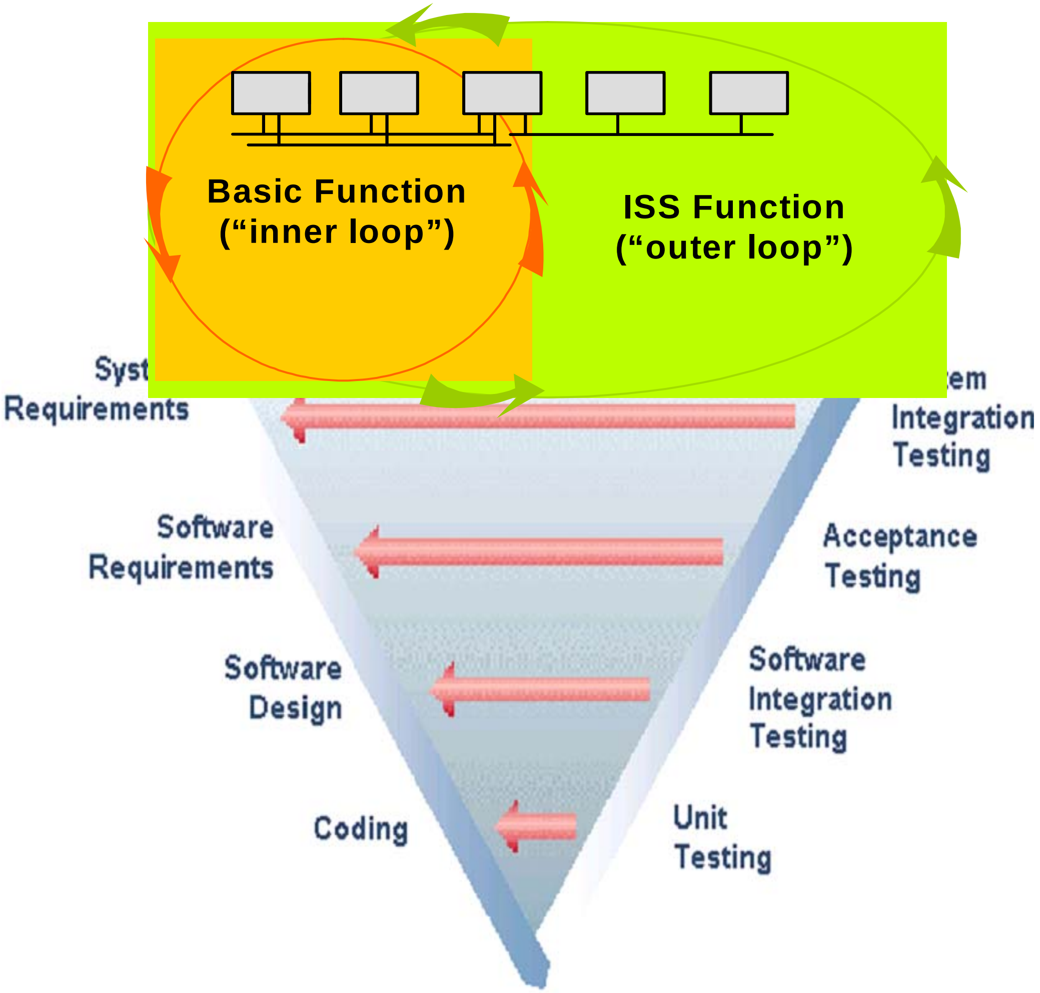
Scalable EASIS hardware architecture



EASIS framework for dependability

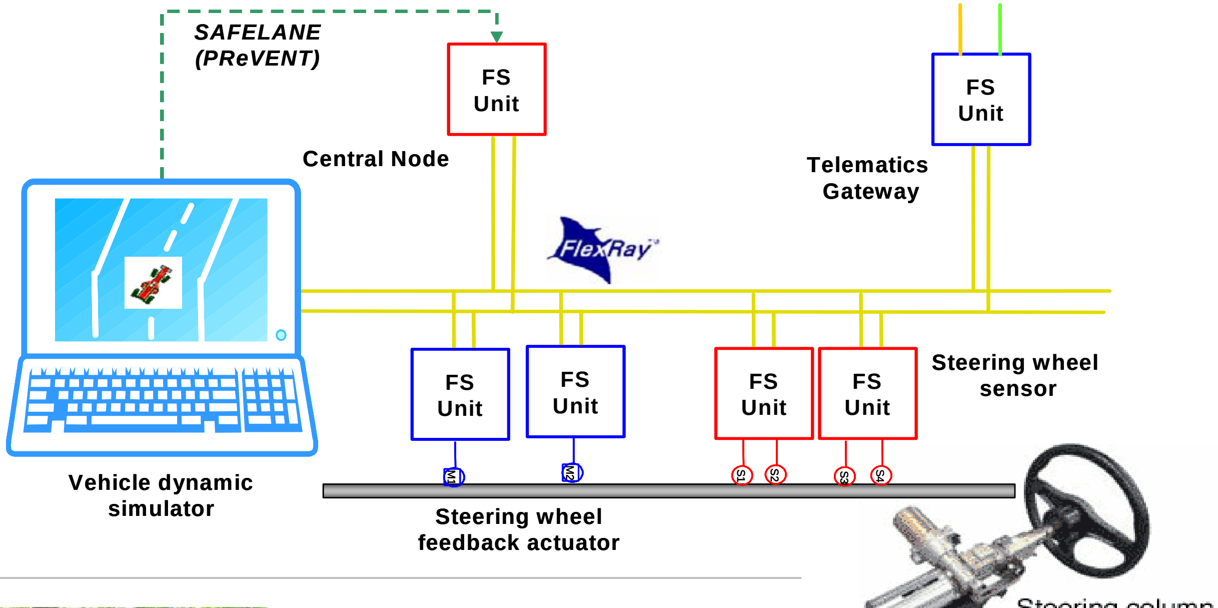

Identification of hazards
Classification of hazards
Hazard occurrence analysis
Establishment of dependability-related requirements
Development and design of the integrated safety system
Verification and validation of dependability-related requirements
Final dependability assessment

EASIS engineering process



Validation / Proof of concept

Validator 1: Telematics gateway validator to prove EASIS SW & HW architecture

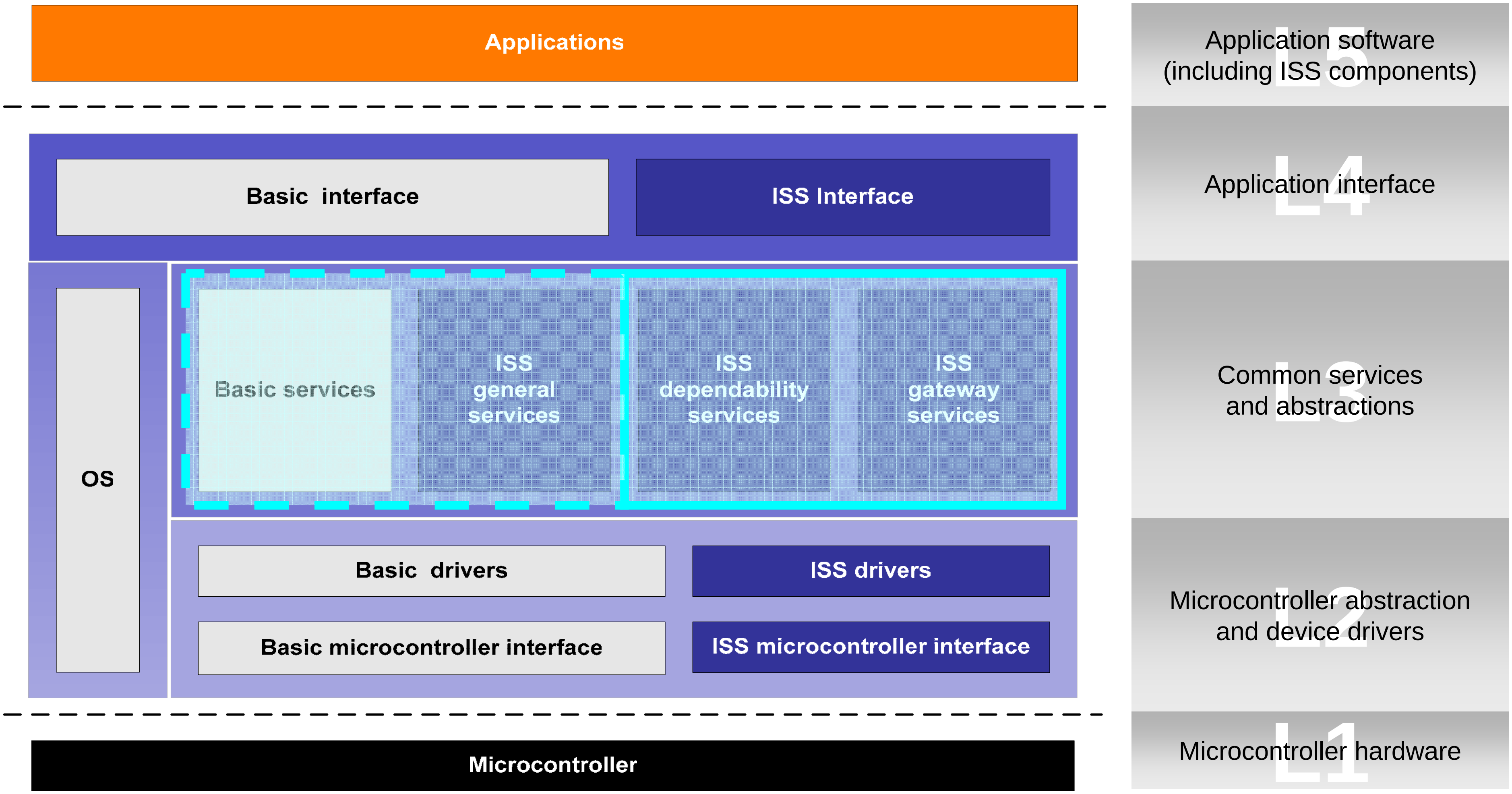


Validator 2: Commercial vehicle Hardware In the Loop testbench to prove EASIS dependability guidelines and development process

Outline

- Background
  - ”The Virtual Safety Belt”
  - Project data
  - Related projects
  - Results overview
- Software platform
  - Layered architecture
  - Fault management framework
  - Dependability support
  - Security support

Overall software topology


Applications
Basic interface
ISS Interface
OS
Basic services
ISS general services
ISS dependability services
ISS gateway services
Basic drivers
ISS drivers
Basic microcontroller interface
ISS microcontroller interface
Microcontroller
L5
Application software (including ISS components)
L4
Application interface
L3
Common services and abstractions
L2
Microcontroller abstraction and device drivers
L1
Microcontroller hardware

Basic software platform – assumptions

- **EASIS will not primarily focus on defining basic services**
  - **However, we will specify which services we require, and assume that these services are defined or being defined elsewhere**
- **Basic services**
  - **Communication managers (CAN, LIN, FlexRay, etc.)**
  - **High-level protocols**
    - Basic network specific transport protocols (e.g. ISO 15765, LIN TP)
    - Calibration protocols (e.g. XCP)
    - …
  - **Network management**
  - **Diagnostic interfaces**
    - E.g. ISO 14229 or ISO 14230
  - **NVRAM manager**
  - **Operating system**
  - **…**

Dependability services

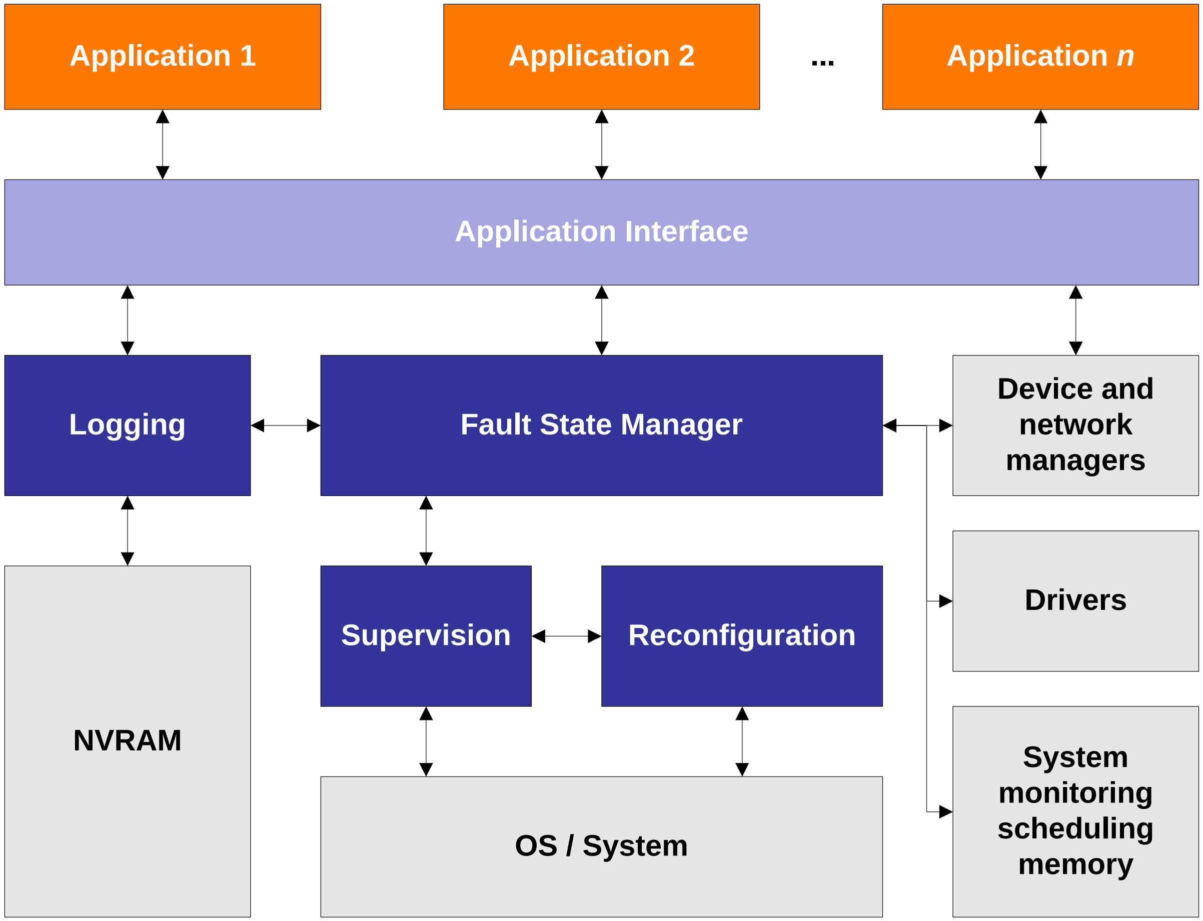
- A set of services and mechanisms concerning dependability has been defined. These are to support the following
  - 🟩 > **Fault management framework**
  - 🟨 > **Fault tolerant communication**
    - OSEK/VDX FTCom – same as used in AUTOSAR
  - 🟩 > **Voting/Agreement protocol**
  - 🟩 > **Watchdog management**
  - 🟥 > **Reconfiguration of applications**
  - 🟥 > **Replication of application components**
  - 🟩 > **Gateway**
  - 🟩 > **Firewall**

Fault management framework

- Dependability and diagnosis services are part of a larger framework for on-board diagnosis

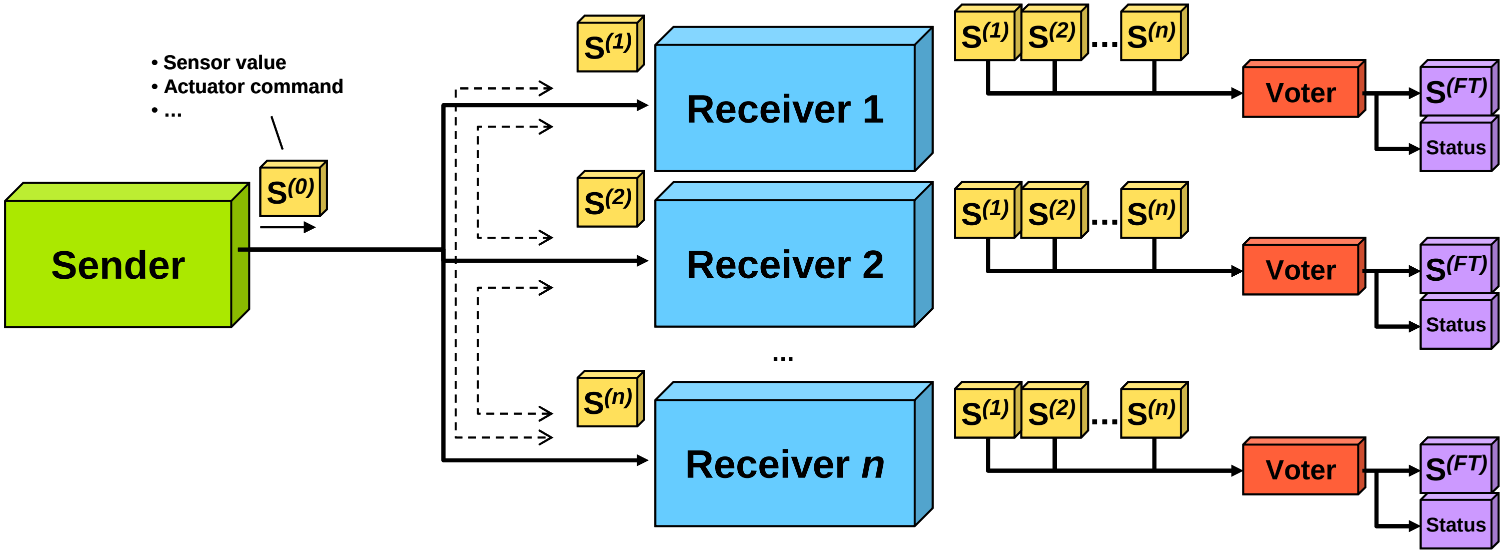
  → Fault management framework
- Main goals
  - > To give a global view of the fault management issue
  - > To ensure the consistency of the fault management strategies
  - > To define central software artifacts for in-vehicle fault management and dependability
- Focus of activities
  - > Act upon error detection notification
  - > Trace and identify faults
  - > Tolerate faults
  - > Other dependability activities
- The structure which “glues” those different elements together
  - > Note that not all parts of the framework are necessarily implemented software artifacts

Fault management framework – current modular view



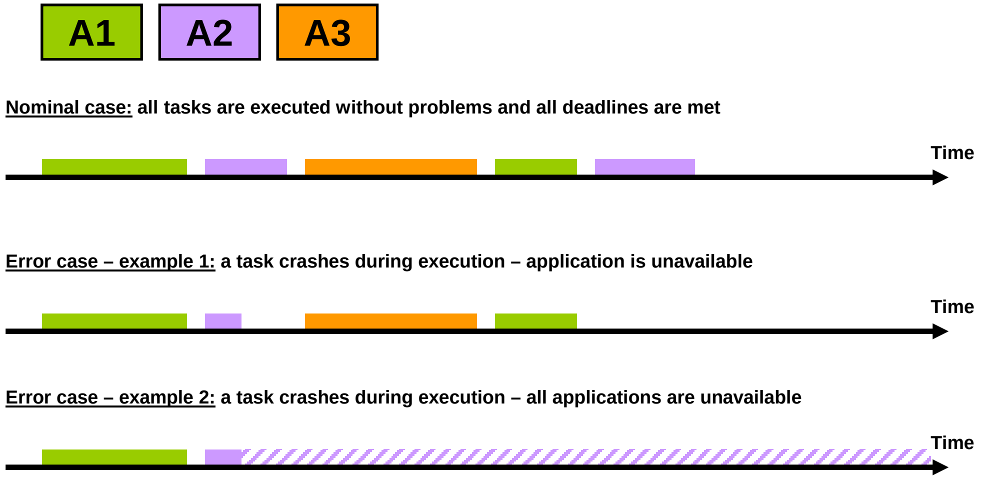
Agreement (and voting)

- Byzantine faults may occur
  - Sender malfunctioning
  - Communication medium faulty
  - …
- Based on *Signed Message Protocol*



Watchdog

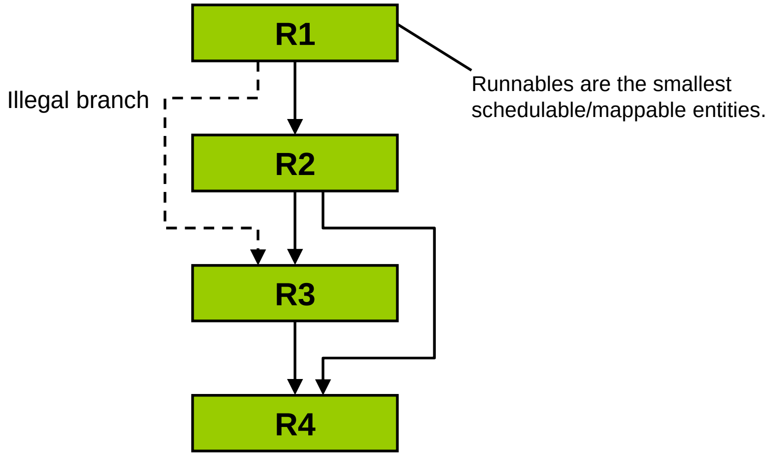
- **Error detection at two levels**
  - **Task level**
    - Crashing tasks
    - Hanging tasks
  - **Runnable level**
    - Faulty execution order



Nominal case: all tasks are executed without problems and all deadlines are met

Error case – example 1: a task crashes during execution – application is unavailable

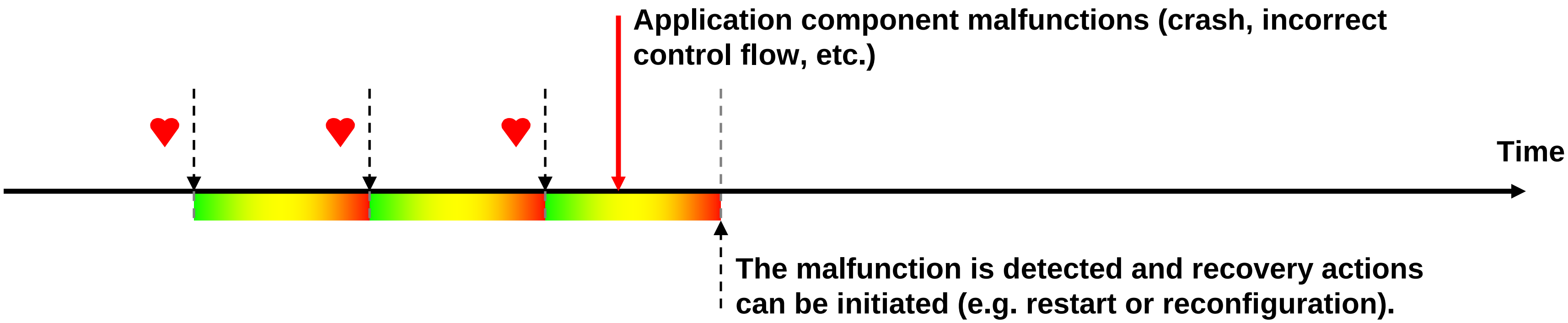
Error case – example 2: a task crashes during execution – all applications are unavailable



Runnables are the smallest schedulable/mappable entities.

Software watchdog – The solution (part I)

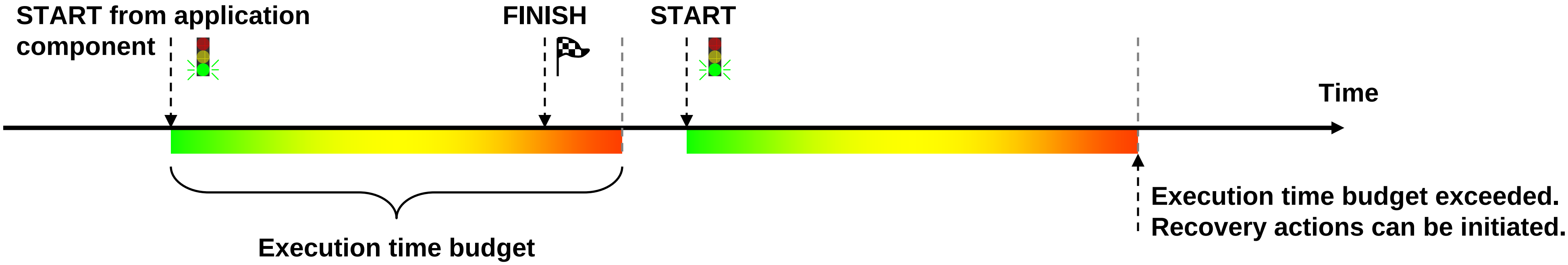
ALIVE-signals/heartbeat from application components



Application component malfunctions (crash, incorrect control flow, etc.)

The malfunction is detected and recovery actions can be initiated (e.g. restart or reconfiguration).

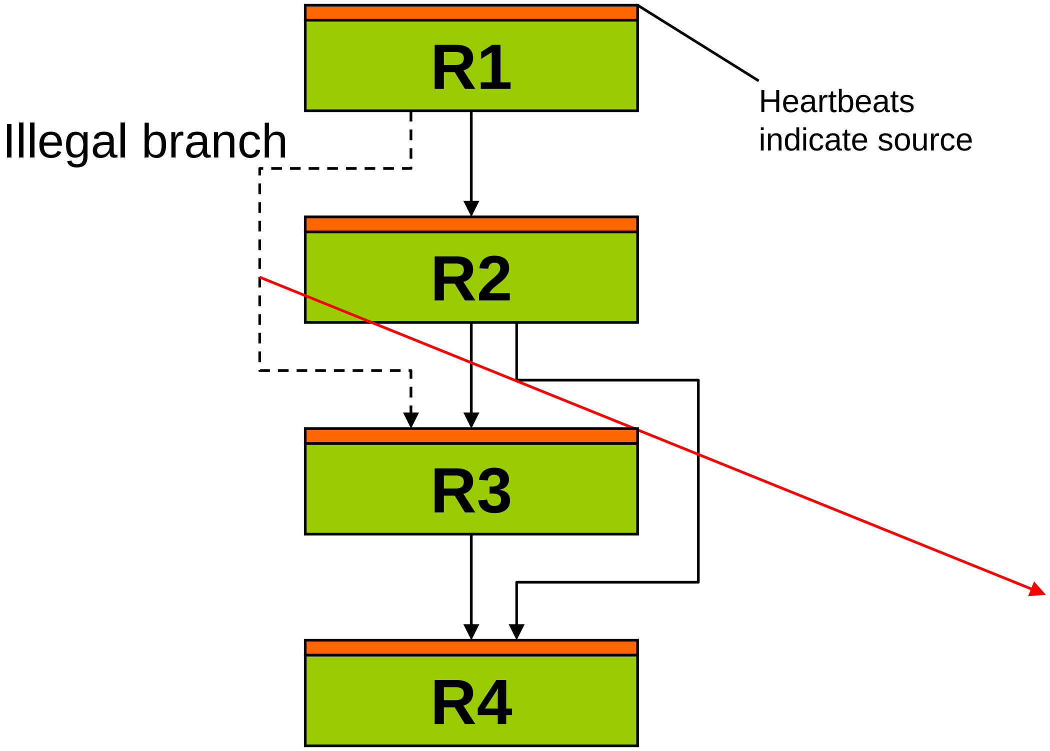
Execution time monitoring of application components



Execution time budget

Execution time budget exceeded. Recovery actions can be initiated.

Software watchdog – The solution (part II)



Runnable	Successors
R1	**R2**
R2	**R3, R4**
R3	**R4**
R4	**…**
…	**…**

$R3 \notin SUCC(R1) = \{R2\}$

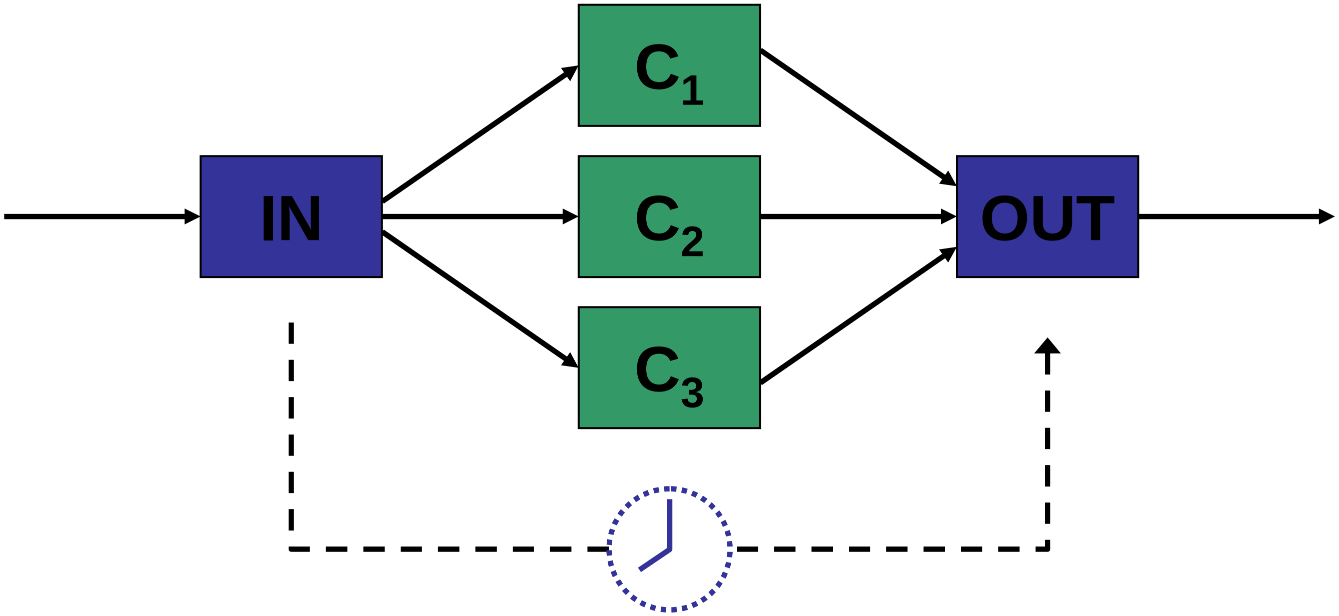
Error detected!

Reconfiguration of applications

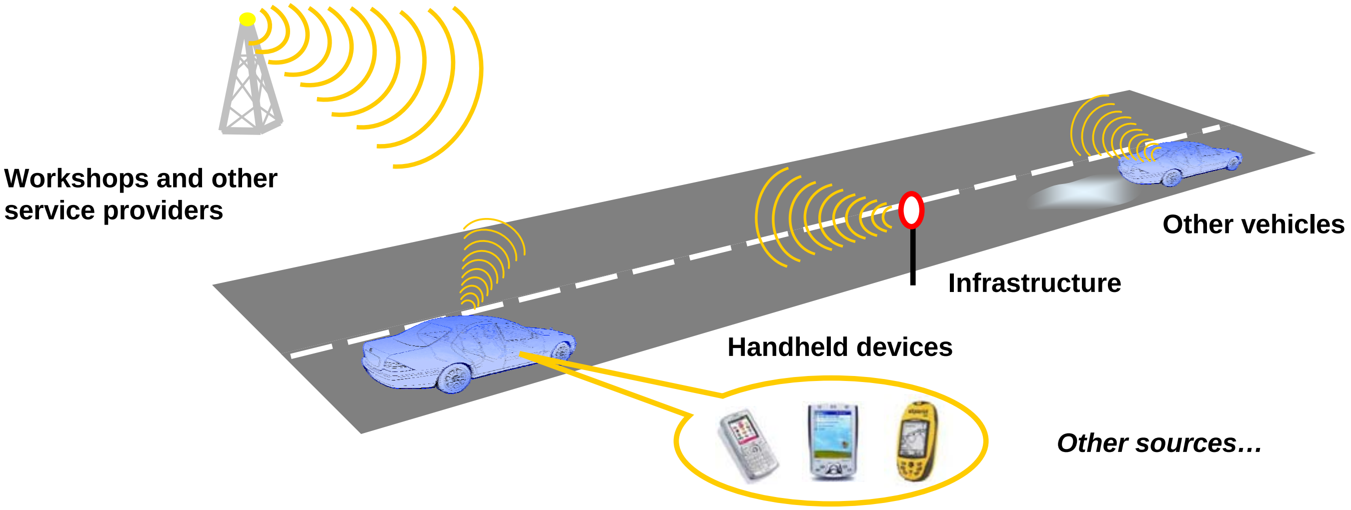
- Reconfiguration can be triggered by the FMF
- Three levels
  - Reconfiguration of active task set
    - Switch between predefined task sets
    - Puts some constraints on mapping of runnables to tasks
  - Functional inhibition
    - Passive w.r.t. platform → application receives info and has to act on this
  - ECU level reset
    - If all else fails → Ctrl-Alt-Del the ECU

Replication of application components at the task level

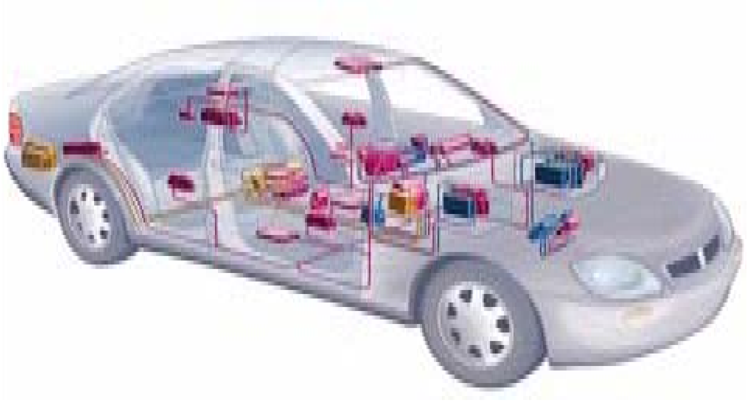
- Basic support required:
  - > Input provider service
  - > Output collector service
  - > Synchronization service



The Communicating Vehicle



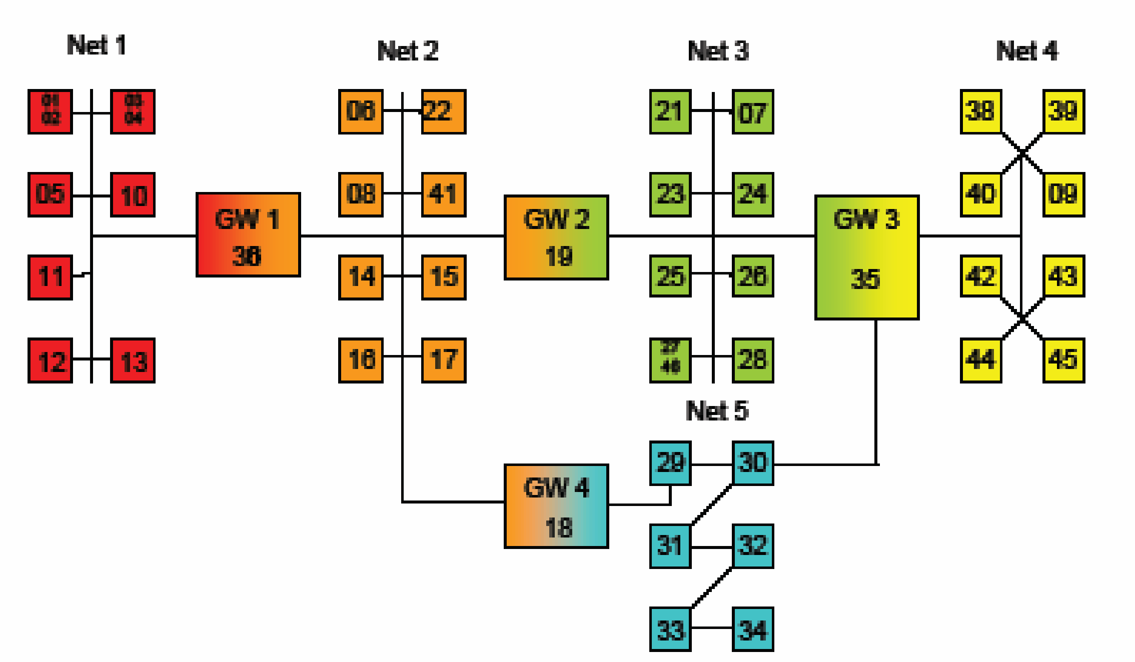
External communication



Internal communication

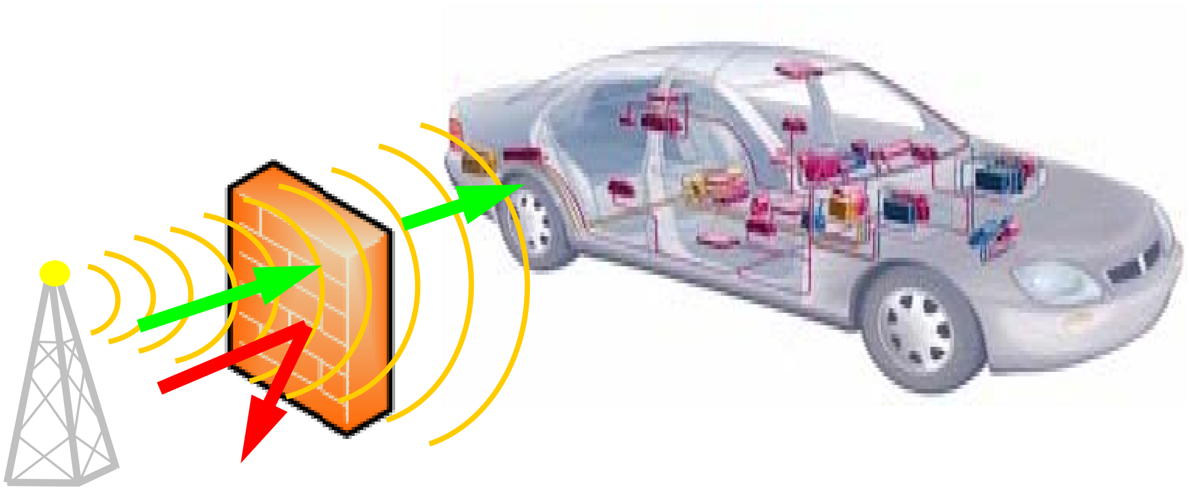
Gateway – relaying information from source to destination

- Vehicle wide transport protocol: EASIS Common Transport Protocol (CTP)
  - > Every ECU has a global alias
  - > Routing tables in the gateways ensure proper delivery



Firewall – Protection against malicious attackers

- Platform support contains firewall-based access control
- Application level firewalls have to be implemented by the application developers
  - Thus, the platform will not check contents of messages – this is the responsibility of the applications



Outline

- Background
  - ”The Virtual Safety Belt”
  - Project data
  - Related projects
  - Results overview
- Software platform
  - Layered architecture
  - Fault Management Framework
  - Dependability support
  - Security support

EASIS

Thank you for your attention!

Get more info on www.easis.org or get your copy of the project folder