Automotive Software Development Challenges
Virtualisation and Embedded Security
# Automotive Software Development Challenges

## Virtualisation and Embedded Security

### Agenda

<table>
<thead>
<tr>
<th></th>
<th>What Makes Automotive Software Uniquely Challenging?</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Virtualization: Helping to manage growing complexity when product life-cycles are getting shorter</td>
</tr>
<tr>
<td>3</td>
<td>Security: Managing new challenges that are only just beginning to be recognized within the automotive industry</td>
</tr>
</tbody>
</table>
## Agenda

<table>
<thead>
<tr>
<th></th>
<th>What Makes Automotive Software Uniquely Challenging?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Virtualization: Helping to manage growing complexity when product life-cycles are getting shorter</td>
</tr>
<tr>
<td>2</td>
<td>Security: Managing new challenges that are only just beginning to be recognized within the automotive industry</td>
</tr>
</tbody>
</table>
A Big Number…
…and a bigger one

80,000,000
Annual worldwide car production

1,000,000,000
Shipments of smartphones in 2013
Automotive Development
Who does what?

Car Makers (the OEMs)
Requirements provider
System integrator
Sometimes the ECU integrator

ECU Suppliers (Tier1s)
Design & implementation
Usually ECU integrator

Significant Interaction
OEM builds the “plant”
Tier1 builds the “controller”
Distributed functions
Sub-contracting to Tier2 suppliers and engineering service providers

... and many more
The Modern Car
A box of electronics on wheels

Complex mechatronic system
Hard real-time constraints
Designed by OEM and several Tier1 suppliers

1-3 Busses
30-40 ECUs

5-7 Busses
100+ ECUs
Vehicle Domains: Powertrain
(Or what does all that stuff do?)

**Engine Management**
- Injection/Spark timing
- Emissions control
- Noise control

**Transmission Control**
- Gear selection
- Terrain Adjustment
Braking
Anti-Lock Braking (ABS) since 1978

Traction Control
Electronic Stability (ESP) since 1995

Source: Robert Bosch GmbH
Vehicle Domains: Body
(Or what does all that stuff do?)

Wiper control / rain sensing

Wing mirrors

Vehicle access

Window lift/anti-trap/pinch

Electronic seats

Heating/ventilation

Airbags

Mood Lighting

Stop/start

Image: ATZ Online
Vehicle Domains: In-Vehicle Infotainment (IVI)
(Or what does all that stuff do?)

**Head Unit**
Radio/CD/MP3 integration
Navigation/Mapping
TV
Internet
Telephony

Basically a “PC in a car”

This area accounts for an increasing part of the “user experience”
Vehicle Domains: Advanced Driver Assistance (ADAS)
(Or what does all that stuff do?)

Adaptive Cruise Control

Park pilot

Lane departure warning

Blind spot warning

Collision mitigation

Active steering

Pedestrian protection

Images: Robert Bosch GmbH
The same as every other industry ...

Constantly **changing** application requirements

Quicker **time to market** demands

Increased **complexity** and functionality

Limited engineering **resources**
... but with some additional and unique challenges

Tight **performance** constraints

Must fit within very limited **resources**
enable **minimal** production **costs**

**High reliability** demands

In **massive** production **volumes**
In places where “patching the software” is **difficult**
Challenge: Manufacturing Cost Pressure
Resource constrained devices

Memory
8MB ROM/512kB RAM is “huge”
256kB ROM/32kB RAM is “typical”

Speed
280MHz is “fast”
40MHz is “typical”

Harsh environment

Trademark property of respective owners
Challenge: Software Development Cost Pressure

Lots of code

≈ 100,000 SLOC

≈ 6,500,000 SLOC

≈ 20,000,000 SLOC

= 500 copies of “The Complete Works of Shakespeare”

Sources
Charette, “This car runs on code”, IEEE Spectrum, Feb 2009
Challenge: Variation
Many models. Many configurations.

1974 2014

3000 Compile time options
35000 Calibration parameters
(for tuning performance)

Image sources: wikipedia.com / Daimler AG
Challenge: Exceptional Reliability Demands
Expensive to fix when it breaks.

22 million Vehicles recalled in US in 2013
17 million sold

$1,000,000,000 Excluding cost of repair
1-6% of company revenue

$1200 Estimated cost per SLOC for Toyota unintended acceleration problem

Sources: New York Times, Klokwork, Autocar AU, Daily Telegraph, EDNbvg
# Agenda

<table>
<thead>
<tr>
<th></th>
<th>Agenda Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>What Makes Automotive Software Uniquely Challenging?</td>
</tr>
<tr>
<td>2</td>
<td>Virtualization: Helping to manage growing complexity when product life-cycles are getting shorter</td>
</tr>
<tr>
<td>3</td>
<td>Security: Managing new challenges that are only just beginning to be recognized within the automotive industry</td>
</tr>
</tbody>
</table>
Feedback Loops in the standard V-Model ...

- Specification, design & implementation
- Virtual integration, test & validation, calibration
- Integration, test and validation, calibration
- System & ECU Network
- Sub-System & ECU Network
- ECU
- Function / Composition
- Unit / Component

... require **hardware prototypes** for validation purposes
... result in **long reaction times** through late validation
Virtualisation in Automotive Software Development

Introduction & Motivation

60% of development time no prototypes are available

Only 10% of engineers get to test in a real car

Source: 7th MODPROD Workshop on Model-Based Product Development, Linköping University – February 5-6, 2013, Stefan-Alexander Schneider, BMW, Berlin
Shorter Feedback Loops ➔ Faster to Market

The competing pressures of reducing time to market, reduced engineering capacity and increasing complexity requires significant jumps in development efficiency.

AUTOSAR Enables Early Development & Validation in Virtual Environment

- The AUTOSAR Microcontroller abstraction layer encapsulates hardware dependencies.
- Identical source code can be executed in a virtual environment for early validation.
Validate early
- Test your ECU SW on the developer PC (just as your models)
- Virtually integrate your SW components prior to their release
- Independent of the availability of HW prototypes or test benches

Speed up the functional validation
- Faster cycles – no reprogramming/flashing after every change
- Easier debugging /reproducing of problems on the PC

Utilize the flexibility of a PC (full access to variables, HDD, …)
- Fault stimulation, single-step debugging, “unlimited” data logging
- Keep the same interfaces as the ECU (e.g. access via XCP)

Save even more costs
- Less ECU samples (especially of the costly ones in early phases)
- Reuse your test assets (from virtual to real ECU)
## EVE Control

<table>
<thead>
<tr>
<th>VECU</th>
<th>Target Type</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>VECU_EcuInstance</td>
<td>RTPC</td>
<td>rtpc (192.168.40.14)</td>
</tr>
</tbody>
</table>

**Note:**
- **ETAS Virtualisation Solutions**
- **Software Development and Validation: ISOLAR - EVE**
- **© ETAS GmbH 2014. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.**
# Agenda

<table>
<thead>
<tr>
<th></th>
<th>Agenda</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>What Makes Automotive Software Uniquely Challenging?</td>
</tr>
<tr>
<td>2</td>
<td>Virtualization: Helping to manage growing complexity when product life-cycles are getting shorter</td>
</tr>
<tr>
<td>3</td>
<td>Security: Managing new challenges that are only just beginning to be recognized within the automotive industry</td>
</tr>
</tbody>
</table>
Automotive Security Challenges

There are threats & attacks on **all parts** of the vehicle! There are threats & attacks during **all periods** of a vehicle lifecycle!

**Challenges**

- Security for the **whole vehicle**
- Security for the **whole life cycle**

**Challenges: Secure E/E Architecture (#1) over complete vehicle lifecycle (#2)**
Attacks on remote diagnosis & telematics

GSM modem attack to access car internals

Malware via manipulated Audio-CD

Malware/Hackers via Bluetooth vulnerability

Malware/Hackers via cellphone vulnerability

Hacking remote key fobs

Cloning remote key fobs

Disable safety locks

Chip tuning, steal IP, manipulate mileage

Delete, manipulate, disable logging, privacy infringement

Install counterfeit parts

Manipulate brakes via fake/malicious CAN messages

Steal intellectual property

Execute unauthorized commands

Install counterfeit parts
Challenge #1:
Secure E/E Architecture

- Secure E/E Architecture
  - Secure E/E Platform
  - Secure On-board Network
  - Secure External Communication
Challenge #1a: Secure E/E Platform

ESCRYPT Solutions for Automotive Security

- Secure E/E Architecture
  - Secure E/E Platform
    - Secure Software
    - Secure Hardware
  - Secure On-board Network
  - Secure External Communication

info@escrypt.com
Challenge #1b: Secure On-board Network

- Secure E/E Architecture
  - Secure E/E Platform
  - Secure On-board Network
    - Firewall, IDS/IRS
    - Secure Channel
    - Access Control
  - Secure External Communication
Challenge #1c: Secure External Communication

Secure E/E Architecture

Secure E/E Platform
Secure On-board Network
Secure External Communication

Firewall, IDS/IRS
Secure Channel
Key/Identity Management
Challenge #2: Security over Complete Life Cycle

We offer necessary security assistance (e.g., development, development support, reviews, trainings) for all phases of the entire vehicle life cycle, for instance:

- **Analysis**
  - Security assets
  - Security threats
  - Potential attackers
  - Potential losses
  - Security risks

- **Design**
  - Security requirements
  - Security architecture
  - Security data and functions
  - Security processes
  - Security integration

- **Specification**
  - Low-level security hardware & software descriptions
  - Low-level security process & infrastructure descriptions
  - Security testing specifications

- **Implementation**
  - Software & hardware implementations
  - Infrastructure implementations
  - 3rd party component integration
Challenge #2: Security over Complete Life Cycle

We offer necessary security assistance (e.g., development, development support, reviews, trainings) for all phases of the entire vehicle life cycle, for instance:

**Testing & Evaluation**
- Functional testing
- **Penetration testing**
- Security evaluation (e.g., NIST, Common Criteria)

**Production**
- Security device initialization
- Security infrastructure initialization
- **Personalization (e.g., key injection)**
- Deployment monitoring & logging

**Operation**
- Security maintenance
- Credential management
- **Secure updates**
- Configuration management
- Monitoring / CERT

**Phase-out**
- Deregistration processes
- **Credential revocations**
- Proof of phase-out
Automotive Software Development Challenges

Thank-you for listening... please feel free to ask questions.