AUTOSAR proofs to be THE automotive software platform for intelligent mobility

Dr.-Ing. Thomas Scharnhorst
AUTOSAR Spokesperson

Simon Fürst, BMW AG
Stefan Rathgeber, Continental Corporation
Lorenz Slansky, Daimler AG
Frank Kirschke-Biller, Ford Motor Company
Rick Flores, General Motors

Tony Jaux, PSA Peugeot Citroën
Thomas Rüping, Robert Bosch GmbH
Kenji Nishikawa, Toyota Motor Company
Dr. Carsten Krömke, Volkswagen AG

ELIV, Bonn Germany
October 18th -19th 2017
Overview

Introduction
- Overview and achievements

➢ New challenges and use cases

➢ The Adaptive Platform
  - Overview
  - 1st release of the Adaptive Platform 17-03
  - Roadmap

➢ New cooperation model
  - Agile development
  - Collaboration with other standardization bodies

➢ Summary
E/E innovations in vehicle development are increasing

- Mechanics
- Electronic Support
- Infotainment
- Linked Networks
- 90% of all innovations
- All major innovations are driven by E/E
- Vehicles are connected to the back-end

AUTOSAR – Core Partners and Partners (June 2017)

9 Core Partners
- BMW Group
- BOSCH
- Continental
- DAIMLER
- TOYOTA
- VOLKSWAGEN AG

44 Premium Partners
- Volvo Trucks
- Autoliv
- HELLA
- DENSO
- JTEKT
- Mentor Graphics
- Infineon
- Renesas
- NXP
- TATA Motors

29 Development Partners

91 Associate Partners
19 Attendees

General
- OEM
- Generic Tier 1

Standard Software
- Mentor Graphics
- Xeoma
- VECTOR
- esol

Tools and Services
- ETRI
- Deloitte
- SCSK
- TATA ELXSI Limited
- ARC
- Core

Semi-conductors
- Infineon
- Renesas
- NXP
- TATA ELXSI Limited
- ARC
- Core

Wissensforum
AUTOSAR vision

AUTOSAR aims to improve complexity management of integrated E/E architectures through increased reuse and exchangeability of SW modules between OEMs and suppliers.
Aims and benefits of using AUTOSAR

AUTOSAR aims to standardize the software architecture of Electronic Control Units (ECUs). AUTOSAR paves the way for innovative electronic systems that further improve performance, safety and environmental friendliness.

- Hardware and software will be widely independent of each other.
- Development can be de-coupled by horizontal layers, reducing development time and costs.
- The reuse of software increases at OEM as well as at suppliers. This enhances quality and efficiency during development.
Overview

➢ Introduction
   ■ Overview and achievements

New challenges and use cases

➢ The Adaptive Platform
   ■ Overview
   ■ 1st release of the Adaptive Platform 17-03
   ■ Roadmap

➢ New cooperation model
   ■ Agile development
   ■ Collaboration with other standardization bodies

➢ Summary
**Motivation**

*Main drivers to develop the Adaptive Platform*

- **Highly automated driving**
- **Open and secure access to vehicle**
- **Car-2-X applications**
- **Increased Connectivity**
Selected main drivers for new automotive software systems (1/4)

Highly automated driving will be on the road.

- Driver temporarily/partially passes responsibility for driving task to vehicle
- Support of communication with traffic lights
- Support of high-performance micro-controllers and computing
- Support of high quality map data
**Selected main drivers for new automotive software systems (2/4)**

Car-2-X applications will require the interaction of vehicles and off-board systems.

- Secure on-board communication
- Support of cross domain computing platforms
- Smartphone integration
- Integration of non-AUTOSAR systems

Use cases
Selected main drivers for new automotive software systems (3/4)

Open and secure access will require dedicated means for security

Use cases
- Support secure cloud interaction
- Support of emergency vehicle preemption
- Remote diagnostics and In-field flashing
- Support of distributed services e.g. repair and exchange handling
Selected main drivers for new automotive software systems (4/4)

Upcoming use cases will lead to a stronger interaction of automotive software systems.

- Consideration of non-AUTOSAR systems within methodology
- Dynamic deployment of software components
- Interaction with non-AUTOSAR and off-board systems

Use cases

- Statically deployed application
- Dynamically deployed application
- Off Board application

Diagram:

- SW-C1
- SW-C2
- SW-C3
- SW-C4
- SW-C5
- SW-C6
- ECU1
- ECU2
- ECU3
- ECU4
- Server
Technology Drivers

Ethernet
- High bandwidth
- Communication system is no longer a limiting aspect
- Switched network
- Efficient point-to-point communication
- Efficient transfer of long messages

Processors
- Switch from microcontroller to processors with external memory (and maybe filesystems)
- Many core processors
- Parallel computing
- „Cheap“ availability of computing power

Heterogeneous architectures
- Special purpose processors
Overview

➢ Introduction
   ■ Overview and achievements

➢ New challenges and use cases

The Adaptive Platform

■ Overview
■ 1\textsuperscript{st} release of the Adaptive Platform 17-03
■ Roadmap

➢ New cooperation model
   ■ Agile development
   ■ Collaboration with other standardization bodies

➢ Summary
AUTOSAR Adaptive Platform – emerging from deeply embedded systems

Application framework
- Support for run-time configuration
- Service-oriented communication
- Partial update

Formats for design data
- Planning of dynamic behavior (e.g. constraints for scheduling and communication)
- Consider automotive specific cooperation scenarios
- Support integration with existing systems (Classic Platform)

Reference architecture
- Reuse existing (non-automotive) standards
- Ease software development
- Support automotive use-cases and protocols
- Reference Implementation

… and many more
**Architectural Overview**

**Functional Clusters**

### AUTOSAR Runtime for Adaptive Application (ARA)

#### Adaptive Platform Services
- Service: Diagnostics
- Service: Communications
- Service: Hardware Acceleration
- Service: Logging and Tracing
- Service: Platform Health Management
- Service: Persistency
- Service: Security Management
- Service: Software Configuration Management

#### Adaptive Platform Foundation
- API: Operating system (*

#### (Virtual) Machine / Hardware
- Bootloader
- Time Management
- Execution Management
- Platform Health Management
- Persistency

**AUTOSAR Runtime for Adaptive Applications = \(\Sigma\) of all Functional Cluster APIs / Services**

#### Programming language specific API for a Functional Cluster as specified in SWS

#### Service Interface of a Functional Cluster. API is generated according to ara::com specification

#### Behavioral specification of Functional Cluster

*(*) POSIX OS, not separately standardized
Address space virtualization

Each application lives in its own address space

Access to platform functionality via libraries

Communication via impl. spec. IPC

Adaptive AUTOSAR Services

Adaptive AUTOSAR Foundation

(Virtual) Machine / Hardware

(*) POSIX OS, not separately standardized
Service-oriented communication (1/2)

Dynamic establishing of communication path

➢ Service Discovery finds all local and remote Service Instances in the System.
➢ Available Service Instances are represented by Proxies (P1 … P3) to the Application.
➢ Application can choose which Service Instance(s) to use.
Service-oriented Ethernet communication (2/2)

Adaptive AUTOSAR Foundation

(Virtual) Machine / Hardware

Application

API(libc)

Communication API

Operating system (*)

Bootloader

Adaptive AUTOSAR Foundation

Service Consumer

Proxy

Skeleton

Service Implementation

Provided Interface
Events
Methods
Fields

SOME/IP-Serialization
E2E Protection

SOME/IP Service Discovery

SOME/IP

Communication Management

(*) POSIX OS, not separately standardized
Execution model

Classic Platform vs. Adaptive Platform

Classic Platform:
- ECU
  - SWC
    - Runnables
      - specifies
      - executes
  - RTE

Adaptive Platform:
- ECU
  - Application
    - Manifest
      - Calls Backs
      - Threads
        - void main()
      - creates
      - executes
  - Communication Management
  - Execution Management
    - executes
    - configures
The Adaptive Platform is developed iteratively – the functionality grows with each release.
**AUTOSAR Adaptive Platform Feature Roadmap**

<table>
<thead>
<tr>
<th>AP R17-03</th>
<th>AP R17-10</th>
<th>AP R18-03</th>
<th>AP R18-10</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Safety</strong></td>
<td><strong>Security</strong></td>
<td><strong>Logging / Tracing</strong></td>
<td><strong>Persistency</strong></td>
</tr>
<tr>
<td>• Platform Health Management</td>
<td>• Crypto API</td>
<td>• Logging and Tracing</td>
<td>• Data Storage</td>
</tr>
<tr>
<td>• C++/C++14 Coding Guideline</td>
<td>• Authentication &amp; Certificates</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• E2E Communication Integrity</td>
<td>• Key Management</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Methodology</strong></td>
<td><strong>Safety</strong></td>
<td><strong>Logging / Tracing</strong></td>
<td><strong>Methodology</strong></td>
</tr>
<tr>
<td>• Diagnostic Extract</td>
<td>• Platform Health Management</td>
<td>• Logging and Tracing</td>
<td>• Methodology Extensions for Adaptive</td>
</tr>
<tr>
<td>• Manifests</td>
<td>• C++/C++14 Coding Guideline</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Methodology Extensions for Adaptive</td>
<td><strong>Persistency</strong></td>
<td><strong>Methodology</strong></td>
<td><strong>Persistency</strong></td>
</tr>
<tr>
<td><strong>Diagnostics</strong></td>
<td>• Data Storage</td>
<td>• Methodology Extensions for Adaptive</td>
<td>• Data Storage</td>
</tr>
<tr>
<td>• DTC Management</td>
<td>• Safe Data Storage</td>
<td>• Service to signal modeling</td>
<td>• Safe Data Storage</td>
</tr>
<tr>
<td>• ISO 14229 / ISO 13400</td>
<td>• Persistent Data Encryption</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Communication</strong></td>
<td><strong>Diagnostics</strong></td>
<td><strong>Communication</strong></td>
<td><strong>Execution Management</strong></td>
</tr>
<tr>
<td>• Service-Discovery and Service-based Communication</td>
<td>• ISO 14300 / ISO 14229 completion</td>
<td>• Time Synchronization</td>
<td>• Resource Management</td>
</tr>
<tr>
<td>• Events, Fields and Methods</td>
<td>• Handling of SW Clusters</td>
<td>• Support of RESTful</td>
<td>• Parallel Processing by HWA</td>
</tr>
<tr>
<td>• Language-Binding: C++</td>
<td></td>
<td></td>
<td>• Machine State handling</td>
</tr>
<tr>
<td>• Inter-Process-Communication</td>
<td></td>
<td></td>
<td><strong>Execution Management</strong></td>
</tr>
<tr>
<td>• Bus-Binding: SOME/IP</td>
<td></td>
<td></td>
<td>• Resource Management</td>
</tr>
<tr>
<td><strong>Execution Management</strong></td>
<td><strong>Communication</strong></td>
<td><strong>Execution Management</strong></td>
<td><strong>Maintainance and Improvements</strong></td>
</tr>
<tr>
<td>• Integration of Applications onto Platform</td>
<td>• Signal-based Communication</td>
<td>• Resource Management</td>
<td>• Maintenance and Improvements</td>
</tr>
<tr>
<td>• Start and Stop of Applications</td>
<td>• Support of RESTful</td>
<td>• Parallel Processing by HWA</td>
<td>• Machine State handling</td>
</tr>
<tr>
<td><strong>POSIX Operating System</strong></td>
<td><strong>Diagnostics</strong></td>
<td><strong>Security</strong></td>
<td><strong>SW-Configuration Mgmt</strong></td>
</tr>
<tr>
<td>• OS Application Interface (PSE51 &amp; C++STL)</td>
<td>• Maintenance and Improvements</td>
<td>• Crypto API</td>
<td>• Container Support</td>
</tr>
<tr>
<td><strong>Methodology</strong></td>
<td>• Common System Model</td>
<td>• Authentication &amp; Certificates</td>
<td>• Package Management</td>
</tr>
<tr>
<td>• Maintenance and Improvements</td>
<td></td>
<td>• Key Management</td>
<td>• Installation routine</td>
</tr>
<tr>
<td><strong>Logging / Tracing</strong></td>
<td>• Service to signal modeling</td>
<td>• Secure Communication</td>
<td>• Diagnostic Client</td>
</tr>
<tr>
<td>• Logging and Tracing</td>
<td></td>
<td>• Support of trusted platform</td>
<td>• Methodology Extensions for Adaptive</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Service to signal modeling</td>
</tr>
</tbody>
</table>

---

**Security**
- Crypto API
- Authentication & Certificates
- Key Management
- Secure Communication
- Support of trusted platform

**Safety**
- Platform Health Management
- Safety concept for AP
- E2E Communication Integrity

**Logging / Tracing**
- Maintenance and Improvements

**Persistency**
- Data Storage

**Diagnostics**
- ISO 14300 / ISO 13400

**Communication**
- Time Synchronization
- Support of RESTful

**Execution Management**
- Resource Management
- Parallel Processing by HWA
- Recovery action framework

**Methodology**
- Methodology Extensions for Adaptive
- Service to signal modeling

**Persistency**
- Maintenance and Improvements

**Diagnostics**
- Maintenance and Improvements

**Communication**
- Signal-based Communication
- Support of RESTful
- Network Management Ethernet

**Execution Management**
- Resource Management
- Parallel Processing by HWA
- Recovery action framework

**Persistency**
- Data Storage
- Safe Data Storage
- Persistent Data Encryption

**Diagnostics**
- ISO 14300 / ISO 14229 completion
- Handling of SW Clusters

**Methodology**
- Methodology Extensions for Adaptive
- Service to signal modeling

**Persistency**
- Maintenance and Improvements

**Diagnostics**
- Maintenance and Improvements

**Communication**
- Signal-based Communication
- Support of RESTful
- Network Management Ethernet

**Execution Management**
- Resource Management
- Parallel Processing by HWA
- Recovery action framework

**Persistency**
- Data Storage
- Safe Data Storage
- Persistent Data Encryption

**Diagnostics**
- ISO 14300 / ISO 14229
- DTC Management
- Handling of SW Clusters

**Methodology**
- Methodology Extensions for Adaptive
- Service to signal modeling

**Persistency**
- Maintenance and Improvements

**Diagnostics**
- Maintenance and Improvements

**Communication**
- Signal-based Communication
- Support of RESTful
- Network Management Ethernet

**Execution Management**
- Resource Management
- Parallel Processing by HWA
- Recovery action framework

**Persistency**
- Data Storage
- Safe Data Storage
- Persistent Data Encryption

**Diagnostics**
- ISO 14300 / ISO 14229
- DTC Management
- Handling of SW Clusters

**Methodology**
- Methodology Extensions for Adaptive
- Service to signal modeling

**Persistency**
- Maintenance and Improvements

**Diagnostics**
- Maintenance and Improvements

**Communication**
- Signal-based Communication
- Support of RESTful
- Network Management Ethernet

**Execution Management**
- Resource Management
- Parallel Processing by HWA
- Recovery action framework

**Persistency**
- Data Storage
- Safe Data Storage
- Persistent Data Encryption

**Diagnostics**
- ISO 14300 / ISO 14229
- DTC Management
- Handling of SW Clusters

**Methodology**
- Methodology Extensions for Adaptive
- Service to signal modeling

**Persistency**
- Maintenance and Improvements

**Diagnostics**
- Maintenance and Improvements

**Communication**
- Signal-based Communication
- Support of RESTful
- Network Management Ethernet

**Execution Management**
- Resource Management
- Parallel Processing by HWA
- Recovery action framework

**Persistency**
- Data Storage
- Safe Data Storage
- Persistent Data Encryption

**Diagnostics**
- ISO 14300 / ISO 14229
- DTC Management
- Handling of SW Clusters

**Methodology**
- Methodology Extensions for Adaptive
- Service to signal modeling

**Persistency**
- Maintenance and Improvements

**Diagnostics**
- Maintenance and Improvements

**Communication**
- Signal-based Communication
- Support of RESTful
- Network Management Ethernet

**Execution Management**
- Resource Management
- Parallel Processing by HWA
- Recovery action framework

---

**Stay tuned for updates.**

---

**22 19-October-2017**

**ELIV VDI Congress Bonn 2017**
The Challenge: Integration of Different Platforms

Software Abstraction

Common Bus Interface Specification
Overview

➢ Introduction
  ■ Overview and achievements

➢ New challenges and use cases

➢ The Adaptive Platform
  ■ Overview
  ■ 1st release of the Adaptive Platform 17-03
  ■ Roadmap

New cooperation model
  ■ Agile development
  ■ Collaboration with other standardization bodies

➢ Summary
The first release 17-03 already provides specifications and software

Development in AUTOSAR Feature Teams

validate

AUTOSAR Specifications

improve

AUTOSAR Software Implementations

Licensing for exploitation

 Exploitation of released Specifications and exemplary Software Implementation by AUTOSAR partners
Organization of development of the Adaptive Platform
Work mode and contributions

Joint expert group meeting

Run
3 – 4 months

Sprint planning meetings per feature team defined by FBO

Release
approx. twice a year

Sprint
2-5 Days
4 Weeks

Continuous development according to Scrum

Feature Backlog
refinement

Sprint
Sprint planning

Backlog

Scrum
Scrum Meetings

Meetings

Deliverable
Increment

sprint review

sprint retrospective
Releases and revisions of AUTOSAR

<table>
<thead>
<tr>
<th>Year</th>
<th>Quarter</th>
<th>Classic Platform</th>
<th>Foundation</th>
<th>Adaptive Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>2016</td>
<td>Q1</td>
<td>R4.3.0</td>
<td>R 1.0.0</td>
<td>Release R 17-03</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td></td>
<td>R 1.1.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td></td>
<td>R 1.2.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Q4</td>
<td></td>
<td>R 1.X.Y</td>
<td></td>
</tr>
<tr>
<td>2017</td>
<td>Q1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Q4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2018</td>
<td>Q1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Q4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- R4.3.0
- R4.3.1
- R4.4.0
- R 1.0.0
- R 1.1.0
- R 1.2.0
- R 1.X.Y
Overview

➢ Introduction
  ■ Overview and achievements

➢ New challenges and use cases

➢ The Adaptive Platform
  ■ Overview
  ■ 1st release of the Adaptive Platform 17-03
  ■ Roadmap

➢ New cooperation model
  ■ Agile development
  ■ Collaboration with other standardization bodies

Summary
## Architectural Overview

### Classic Platform vs. Adaptive Platform

<table>
<thead>
<tr>
<th>Classic Platform</th>
<th>Adaptive Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Based on OSEK</td>
<td>Based on POSIX (PSE51)</td>
</tr>
<tr>
<td>Execution of code directly from ROM</td>
<td>Application is loaded from persistent memory into RAM</td>
</tr>
<tr>
<td>Same address space for all applications (MPU support for safety)</td>
<td>Each application has its own (virtual) address space (MMU support)</td>
</tr>
<tr>
<td>Optimized for signal-based communication (CAN, FlexRay)</td>
<td>Service-oriented communication</td>
</tr>
<tr>
<td>Fixed task configuration</td>
<td>Support of multiple (dynamic) scheduling strategies</td>
</tr>
<tr>
<td>Specification</td>
<td>Standard is defined by specification Code as reference implementation</td>
</tr>
</tbody>
</table>
Summary

Achievements

➢ Established a worldwide software standard focusing on automotive applications
➢ Classic Platform is massively used in series production

AUTOSAR Standards

➢ Already launched: AUTOSAR Classic Platform, AUTOSAR Foundation and AUTOSAR Acceptance Tests
➢ First release of AUTOSAR Adaptive Platform 17-03 launched
➢ Reference Implementation available for partners

Future of AUTOSAR

➢ Improvement and stabilization of existing standard
➢ Anticipate the future by providing the next generation of platform software
➢ Creation of new eco-systems by new collaboration models

AUTOSAR will continue to be THE creator of automotive software standards.
More information available online

More information about AUTOSAR:
http://www.autosar.org

Become a partner and get exploitation rights for the AUTOSAR standard
admin@autosar.org

For information only (see disclaimer)

Published Releases