AUTOSAR Introduction

Part 2 – Current Features in a Nutshell
Part 1
► The AUTOSAR Partnership
► The AUTOSAR Standardization

Part 2
► Architecture and Features
  • AUTOSAR in Vehicle Network
  • AUTOSAR Foundation
  • AUTOSAR Classic Platform
  • AUTOSAR Adaptive Platform
► Smart Solutions Based on AUTOSAR
► Processes and Quality
AUTOSAR in a Vehicle Network

Common Bus Interface Specification

e.g. SOME / IP

BUS
AUTOSAR Foundation
Common Features

The Foundation **assures compatibility** of the different AUTOSAR standards and therefore **contains** all **common artifacts** such as ...
The Methodology, derived out of the Meta Model, …

… provides means to describe the AUTOSAR architecture with all its interfaces.

… defines exchange formats and description templates (e.g. manifest) to enable

• a seamless integration of the complete vehicle E/E architecture,
• the automatized configuration of the µC- and µP-software stacks and
• the seamless integration of application software.

… supports means to ensure safety and security of the system.

… provides templates to document the standard.
The layered architecture of the classic platform basically supports:

- Hardware abstraction
- Scheduling of runnables and tasks (OS)
- Communication between applications on the same hardware and over the network
- Diagnosis and diagnostic services
- Safety- and
- Security Services
AUTOSAR Classic Platform
Layered Software Architecture (2/2)
AUTOSAR Adaptive Platform
Architecture - Logical view

User Applications

<table>
<thead>
<tr>
<th>Adaptive Application</th>
<th>Adaptive Application</th>
<th>Adaptive Application</th>
<th>ASW::XYZ Non-PF Service</th>
<th>ASW::ABC Non-PF Service</th>
</tr>
</thead>
</table>

**AUTOSAR Runtime for Adaptive Applications (ARA)**

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication Management</td>
<td>ara::com</td>
<td></td>
</tr>
<tr>
<td>Persistence</td>
<td>ara::per</td>
<td></td>
</tr>
<tr>
<td>Execution Management</td>
<td>ara::exec</td>
<td></td>
</tr>
<tr>
<td>Identity and Access Management</td>
<td>ara::iam</td>
<td></td>
</tr>
<tr>
<td>Cryptography</td>
<td>ara::crypto</td>
<td></td>
</tr>
<tr>
<td>Time Synchronization</td>
<td>ara::tsync</td>
<td></td>
</tr>
<tr>
<td>Diagnostics</td>
<td>ara::diag</td>
<td></td>
</tr>
<tr>
<td>Log and Trace</td>
<td>ara::log</td>
<td></td>
</tr>
<tr>
<td>Platform Health Management</td>
<td>ara::phm</td>
<td></td>
</tr>
<tr>
<td>State Management</td>
<td>ara::sm service</td>
<td></td>
</tr>
<tr>
<td>Update and Config Management</td>
<td>ara::ucm service</td>
<td></td>
</tr>
<tr>
<td>Network Management</td>
<td>ara::nm service</td>
<td></td>
</tr>
</tbody>
</table>

Key

- **SERVICE Non-PF Service**
- **SERVICE Platform Service FCs**
- **API Platform Foundation FCs**

POSIX PSE51 / C++ STL
Operating System Interface

(Virtual) Machine / Container / Hardware
AUTOSAR Standards
Roadmap

- Release R19-11: Nov 2019
- Release R20-11: Nov 2020
- Release R21-11: Nov 2021
- Release R22-11: Nov 2022

Q4 2019 - Q4 2023

- Q4 2019
- Q2 2020
- Q4 2020
- Q2 2021
- Q4 2021
- Q2 2022
- Q4 2022
- Q4 2023
Agenda

Part 1
► The AUTOSAR Partnership
► The AUTOSAR Standardization

Part 2
► Architecture and Features
► Smart Solutions Based on AUTOSAR
  • Software Architecture – AUTOSAR Defined Interfaces
  • Distribution ECUs
  • AUTOSAR Platform Application
► Processes and Quality
Software Architecture – AUTOSAR Defined Interfaces
Use Case ‘Front Light Management’: Exchange Type of Front Light
Software Architecture – AUTOSAR Defined Interfaces

Use Case ‘Front Light Management’: Exchange Type of Front Light

<table>
<thead>
<tr>
<th>Integrator</th>
<th>Supplier B</th>
<th>OEM</th>
<th>Supplier A</th>
</tr>
</thead>
<tbody>
<tr>
<td>SwitchEvent</td>
<td>LightRequest</td>
<td>Front-Light Manager</td>
<td>Xenonlight</td>
</tr>
<tr>
<td>check_switch()</td>
<td>switch event (event)</td>
<td>request_light (type, mode)</td>
<td>set_light(type, mode)</td>
</tr>
<tr>
<td>Switch_event (event)</td>
<td>request_light (type, mode)</td>
<td>get_keyposition()</td>
<td>set_current(...)</td>
</tr>
<tr>
<td>AUTOSAR Interface</td>
<td>AUTOSAR Interface</td>
<td>AUTOSAR Interface</td>
<td>AUTOSAR Interface</td>
</tr>
</tbody>
</table>

- **Integrator**: SwitchEvent
- **Supplier B**: LightRequest
- **OEM**: Front-Light Manager
  - request_light (type, mode)
  - get_keyposition()
  - set_light (type, mode)
  - set_dboard(type, mode)
- **Supplier A**: Xenonlight
  - set_light(type, mode)
  - set_current(...)
Distribution on ECUs – ‘Front-Light Management’

- SwitchEvent
  - check_switch()
  - switch_event(event)

- LightRequest
  - switch_event(event)
  - request_light(type, mode)

- Front-Light Manager
  - request_light(type, mode)
  - get_keyposition()
  - set_light(type, mode)

- Xenonlight
  - set_light(type, mode)
  - set_current(…)

AUTOSAR Interface

CAN Bus
AUTOSAR Platform Application
Continuous improvement cycle for ADAS systems

**On Board**
- Scene Understanding
- Sensor Fusion
- Perception
- ara:adi
- ara:com
- Plan Execution, Motion Control

**Off Board**
- Labeling of trained data
- Evolution by training
- Simulation and sign-off

Secure data exchange on top of DDS SOME/IP or REST
Autosar Platform Application

User Applications

- ara::com Communication Mgmt.
- ara::rest RESTful
- ara::time Time Synchronization
- ara::per Persistence
- ara::plm Platform Health Mgmt.
- ara::idm Identity Access Mgmt.
- ara::log Logging & Tracing
- ara::crypto Cryptography

Run Execution Manager:

- `ara::exec start { Vehicle Manager }`
- `ara::exec start { HMI }`
- `ara::exec start { Distance Radar }`

The operation completed successfully.

**User Applications**

- Vehicle Manager
- HMI
- Distance Radar

(Virtual) Machine / Container / Hardware

>> Run Execution Manager
...
...
...
ara:: exec > start { Vehicle Manager } The operation completed successfully
ara:: exec > start { HMI } The operation completed successfully
ara:: exec > start { Distance Radar } The operation completed successfully
Adaptive Cruise Control (ACC) installation:

- Run Persistency
- Deploy persistent data to Key-Value Database:
  - Vehicle Manager
  - HMI
  - Distance Radar
  - ACC

User Applications:

- ara::com Communication Mgmt.
- ara::rest RESTful
- ara::time Time Synchronization
- ara::per Persistence
- ara::exec Execution Mgmt.
- ara::iam Identity Access Mgmt.
- ara::log Logging & Tracing
- ara::s2s service Signal to Service Mapping
- ara::diag service Diagnostics
- ara::net service Network Management
- ara::adi service Automated Driving Interfaces
- ara::ucm service Update and Configuration Management

AUTOSAR Runtime for Adaptive Applications (ARA)

(Virtual) Machine / Container / Hardware

>> Run Persistency
...
The operation completed successfully

```
ara::exec > start { Vehicle Manager }
The operation completed successfully
ara::exec > start { HMI }
The operation completed successfully
ara::exec > start { Distance Radar }
The operation completed successfully
ara::exec > start { ACC }
The operation completed successfully
```
AUTOSAR Introduction - Part 2

Vehicle Manager
HMI
Distance Radar
ACC

 ara::com Communication Mgmt.
 ara::rest RESTful
 ara::time Time Synchronization
 ara::per Persistency
 ara::ptm Platform Health Mgmt.
 ara::exec Execution Mgmt.
 ara::iam Identity Access Mgmt.
 ara::log Logging & Tracing
 ara::s2s service
 Signal to Service Mapping
 ara::sdi service
 State Management
 ara::diag service
 Diagnostics
 ara::adi service
 Automated Driving Interfaces
 ara::crypto Cryptography
 ara::cm service
 Network Management
 ara::cm service
 Update and Configuration Management

POSIX PSE51 / C++ STL Operating System

(Virtual) Machine / Container / Hardware
## AUTOSAR Introduction - Part 2

### AUTOSAR Runtime for Adaptive Applications (ARA)

<table>
<thead>
<tr>
<th>Communication Mgmt.</th>
<th>Time Synchronization</th>
<th>Platform Health Mgmt.</th>
<th>Identity Access Mgmt.</th>
<th>Logging &amp; Tracing</th>
<th>Service State Management</th>
<th>Diagnostics</th>
<th>Automated Driving Interfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>ara::com</td>
<td>ara::rest</td>
<td>ara::ptm</td>
<td>ara::core</td>
<td>ara::log</td>
<td>ara::ss2s</td>
<td>ara::cmm</td>
<td>ara::adi</td>
</tr>
<tr>
<td>Persistency</td>
<td>RESTful</td>
<td>Platform</td>
<td>Core Types</td>
<td>Encryption</td>
<td>Service Mapping</td>
<td>Service</td>
<td>Interfaces</td>
</tr>
<tr>
<td>Execution Mgmt.</td>
<td>Time</td>
<td>Health Mgmt.</td>
<td>Logging</td>
<td>Network Management</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>POSIX PSE51 / C++ STL Operating System</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(Virtual) Machine / Container / Hardware

- Vehicle Manager
- HMI
- Distance Radar
- ACC
Agenda

Part 1
► The AUTOSAR Partnership
► The AUTOSAR Standardization

Part 2
► Architecture and Features
► Smart Solutions Based on AUTOSAR
► Processes and Quality
  • AUTOSAR Adaptive Platform Development Approach
AUTOSAR Adaptive Platform Development Approach

**Specification**

**Identify needs & use-cases:**
1) Concepts
2) Features
3) Requirements

**Gain speed:**
1) Spec validation
2) Reduce room for spec interpretation
3) Training / dissemination of AP

**Quality:**
- TF-ARC approval
- Cross team review
- Lifecycle: preliminary → draft → valid

**Implementation**

**Attracting environment for coders:**
- Appealing technology (C++, Yocto, Git, …)
- Modern use case (ADAS EBA)
- Handy documentation (Wiki)
- Peer programming sessions

**Demonstration**

**Gain trust:**
1) Advertisers the progress
2) Highlights some specific features

**Show AUTOSAR interoperability**
- of classic and adaptive platforms
- but also with others

---

Best tradeoff between commercial cooperation & compatibility between different vendors
Thank you for your attention

If you’d like to become a partner, contact us at:

+49 89 23 88 57 410
admin@autosar.org
http://autosar.org

Bremer Str. 11
80807 Munich
Germany