AUTOSAR Adaptive Platform
Key concepts and development process

Markus Bechter
AUTOSAR Open Conference
2016-09-28, Gothenburg
<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>
</tbody>
</table>
Service Interface of a Functional Cluster. API is generated according to ara::com specification
Programming language specific API for a Functional Cluster as specified in SWS
Behavioral specification of Functional Cluster
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

- Software Configuration Management
- Security Management
- Diagnostics

Adaptive AUTOSAR Foundation

(Virtual) Machine / Hardware

- Time Management
- Execution Management
- Persistency
- Platform Health Management
- Logging and Tracing
- Hardware Acceleration
- Communication Management

Application 1

Application 2

API(libc) API API

Bootloader

API

API

API

API

API

API
ara::com – Service-oriented communication

Adaptive AUTOSAR Foundation

Application

API(libc)
Communication API

Operating system

Bootloader

(Virtual) Machine / Hardware

SOME/IP

Communication Management

Service Consumer
Proxy

Service Implementation
Skeleton

SOME/IP-Serialization
E2E Protection

SOME/IP Service
Discovery

Service Interface
Definition

Provided Interface
Events
Methods
Fields
Service-oriented communication
Dynamic establishing of communication path

ServiceDiscovery finds all local and remote ServiceInstances in the System.
Available ServiceInstances are represented by Proxies (P1 … P3) to the Application.
Application can choose which ServiceInstance(s) to use.
Work mode and development process
Process to introduce features

Plan
- Development readiness
- Agree on features

Monitor
- Create visions
  - Decide and ramp up of new feature teams
  - Accept visions

Approve
- Support material received
- Technical readiness
- Create epics and user stories
  - Elaborate user stores

Release and Publication
- Post Release
- Publication readiness

Technical readiness
Scope for a release defined and understood by FBOs

PL-Team
- FBOs
- Feature Teams
Feature Teams

PL Subgroup Adaptive Platform (SG-AP)

FBO
Feature Team Communication Management

FBO
Feature Team Execution Management & OS Interface

FBO
Feature Team Diagnostics

FBO
Feature Team Methodology & Manifests

FBO
Feature Team Safety

FBO
Feature Team Persistency

FBO
Feature Team Demonstrator Integration

Call for experts coming soon
Work mode and how-to contribute

- Active and passive members
- Joint expert group meeting
- Run 3 – 4 months
- Continuous development according to Scrum
- Sprint planning meetings per feature team defined by FBO
- Scrum Meetings (2-5 per week)
- Sprint Review
- Retrospective

Sprint planning meetings per feature team defined by FBO

Continuous development according to Scrum

Feature Backlog refinement

Sprint planning

Scrum Meetings

Sprint Review

Retrospective

Active (0.5 FTE) members

Run 3 – 4 months
Infrastructure

- Ticketing system
- Backlog management
- Progress tracking
- Designed to support agile development processes
- ...

https://jira.autosar.org

- Git repository management
- Code reviews
- Activity feeds
- ...

https://code.autosar.org
Planned features

Security
- User management
- File Encryption
- Crypto Hardware

Safety
- Platform Health Mgmt.
- E2E Protection
- Persistent storage of data
- SW Lockstep

Communication
- Ethernet
- CAN support
- Signal based comm.
- FlexRay
- LIN

Base system
- Diagnostics
- DLT
- Execution Management
- Operating System
- Resource management
- Software installation
- HW Acceleration (OpenCL)
- Platform Health Mgmt.
- File Encryption
- Signal based comm.
- CAN support
- FlexRay
- LIN

R17-03
R17-10
R18-03
R18-10

Vehicle API
Container support
More information available online

Welcome to the AUTOSAR development partnership

AUTOSAR (AUTomotive Open System ARchitecture) is a worldwide development partnership of vehicle manufacturers, suppliers and other companies from the electronics, semiconductor and software industry.

- Paves the way for in-vehicle systems that are flexible enough to adapt to changes in functionality, safety and environmental friendliness.
- Is a strong global standard for the automotive industry, allowing manufacturers of vehicles to compete on implementation costs and performance.
- Is a key enabling technology for the next generation of vehicles. It aims to be the foundation for standard vehicle architecture to enable mass adoption of connected vehicles, making any compromise with reliability, performance and security quite impossible.
- Facilitates the exchange and update of software and hardware over the service life of the vehicle.

For information only (see disclaimer)

Published Releases

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

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