Dealing with the Challenges for Future Software Systems in the Automotive Industry with the AUTOSAR Standards
Dealing with the challenges for future software systems with the AUTOSAR standards

Goals of the presentation

1. Summarize the challenges in software development for future automotive systems

2. Show ways how standardization within AUTOSAR can support to handle these challenges

3. Emphasize the benefits for series projects using the AUTOSAR standards
Challenges for future automotive systems
The future is there

Future of AUTOSAR aims to specify fully dynamic service discovery mechanism.

Now we are in the „Future of AUTOSAR“ – Adaptive Platform becomes reality!

- Adaptive Platform extends the scope beyond dynamic communication via Ethernet
- Dynamic systems are the enabler for update over the air, V2X and highly automated driving use cases
Challenges for future automotive systems

Why dynamic systems?

Customers expect functionality that is known from consumer IT domain

- Definition of abstraction from HW and lower SW layers
- POSIX is one enabler for e.g. reusability and updateability in the consumer IT domain

Abstraction layers

Dynamically changing availability of external infrastructure used by vehicle functions

- New platforms can deliver benefits for the integration of the vehicle with external infrastructure
- Service oriented communication leads to easier definition of interfaces

Dynamic communication paradigms
Challenges for future automotive systems

Why dynamic systems?

Highly automated driving needs control over the whole vehicle

- Highly automated driving necessarily needs the intelligence to control all sensors / actuators and therefore leads to a centralized architecture.

Centralized vehicle and software architectures

- Centralizing more functions on one ECU increases bandwidth, CPU power and RAM needs.

Increased resource requirements
To be able to use a product in the automotive industry, it should …

- … be standardized
- … have at least an open specification
- … be open for changes
## Standardization within AUTOSAR

### AUTOSAR Adaptive Platform compared to AGL

<table>
<thead>
<tr>
<th>AUTOSAR Adaptive Platform</th>
<th>Automotive Grade Linux</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harmonization with Classic Platform on technical level</td>
<td>OEM/Tier1 requirements have to be fulfilled separately by each ECU project</td>
</tr>
<tr>
<td>Non-functional automotive specific requirements realized by the standard (e.g. Security, E2E protection)</td>
<td>Code based standard</td>
</tr>
<tr>
<td>Harmonized exchange formats, tools and development processes</td>
<td>Current versions focus on infotainment domain (e.g. multimedia, telematics)</td>
</tr>
</tbody>
</table>

**Compatibility in architecture ensures flexibility**

**Can be a basis for the OS of an Adaptive Platform implementation**

**AUTOSAR Adaptive Platform is the consistent advancement of AUTOSAR Classic Platform and is therefore subject to our further investigation**
Standardization within AUTOSAR
Challenges with interaction between AUTOSAR Classic (CP) and Adaptive Platform (AP)

- 1s to some minutes: Strategies, long time planning, emotions
- 10ms to 5000ms: unconscious cognitive processes & movements / core vehicle functionality
- 1ms to 50ms: Reflexes / fast control loops

Cognitive ECUs
Activity ECUs
Sensor / Actuator ECUs
Standardization within AUTOSAR
What to consider when implementing “neural layers“

- Starting with ASIL C, strictly cyclic execution is required
- Hard to realize with AUTOSAR Adaptive Platform
Standardization within AUTOSAR
What to consider when implementing “neural layers“

- Starting with ASIL C, strictly cyclic execution is required
- Hard to realize with Adaptive Platform

Safety

- In case of changes: recompile the whole function
- Smaller segments of functions can lead to cleaner software architecture

Monolithic functions
Standardization within AUTOSAR
What to consider when implementing “neural layers”

- Starting with ASIL C, strictly cyclic execution is required
- Hard to realize with Adaptive Platform

Safety

- In case of changes: recompile the whole function
- Smaller segments of functions can lead to cleaner software architecture

Monolithic functions

- Customer is used to partial updates in a fast manner
- Customer is used to installing apps

Software updates & installation
Standardization within AUTOSAR
What to consider when implementing “neural layers“

- Starting with ASIL C, strictly cyclic execution is required
- Hard to realize with Adaptive Platform
- In case of changes: recompile the whole function
- Smaller segments of functions can lead to cleaner software architecture
- Customer is used to partial updates in a fast manner
- Customer is used to installing apps
- Implementation of the Model-View-Controller pattern important for
  - the ability to update e.g. only HMI
  - test use cases
- Separation of HMI and algorithms
- Safety
- Software updates & installation
- Monolithic functions
Standardization within AUTOSAR
What to consider when implementing “neural layers“

- Starting with ASIL C, strictly cyclic execution is required
- Hard to realize with Adaptive Platform

- Customer is used to partial updates in a fast manner
- Customer is used to installing apps

- In case of changes: recompile the whole function
- Smaller segments of functions can lead to cleaner software architecture

- Restrict access to memory and I/O resources for specific software parts
- In general: freedom from interference is often necessary

- Implementation of the Model-View-Controller pattern important for
  - the ability to update e.g. only HMI
  - test use cases

Safety
Software updates & installation
Security
Monolithic functions
Separation of HMI and algorithms

CARMED

AUTOSAR Open Conference 2017 - Carmeq, final, 1.0, public
Standardization within AUTOSAR
What to consider when implementing “neural layers”

- Starting with ASIL C, strictly cyclic execution is required
- Hard to realize with Adaptive Platform

- Customer is used to partial updates in a fast manner
- Customer is used to installing apps

- Restrict access to memory and I/O resources for specific software parts
- In general: freedom from interference is often necessary

- In case of changes: recompile the whole function
- Smaller segments of functions can lead to cleaner software architecture

- Implementation of the Model-View-Controller pattern important for
  - the ability to update e.g. only HMI
  - test use cases
Impact of AUTOSAR Adaptive Platform on series ECUs
Effects on software and system architecture

<table>
<thead>
<tr>
<th>Current challenges</th>
<th>Benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td>◆ No scheduling through OS, but application has to ensure timing behavior</td>
<td>◆ Service orientation simplifies usage of interfaces for application developers</td>
</tr>
<tr>
<td>◆ Application needs to set up runtime environments</td>
<td>◆ Higher flexibility due to less strict binding to OS</td>
</tr>
<tr>
<td>◆ Bridge between signal based and service oriented communication needs to be established</td>
<td>◆ Exchange formats are using e.g. the same data types for AUTOSAR Classic and Adaptive Platform</td>
</tr>
<tr>
<td>◆ Leads to latencies and additional safety relevance</td>
<td>◆ Leads to lower effort for migration</td>
</tr>
<tr>
<td></td>
<td>◆ … and all the arguments motivating the usage of AUTOSAR Adaptive Platform as mentioned on previous slides</td>
</tr>
</tbody>
</table>
## Outlook

<table>
<thead>
<tr>
<th>Next Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Further drive harmonization of AUTOSAR Classic and Adaptive Platform until joint Release CP 4.4.0 / AP 2018-10 and beyond</td>
</tr>
<tr>
<td>- Continue elaboration of harmonized security concepts</td>
</tr>
<tr>
<td>- Working concepts for partial updates/upgrades of software</td>
</tr>
<tr>
<td>- Including AUTOSAR Classic Platform ECUs</td>
</tr>
<tr>
<td>- Synchronization between AUTOSAR Adaptive Platform specification and the respective code</td>
</tr>
<tr>
<td>- Further drive the implementation of quality measures for the standard while integrating new features</td>
</tr>
</tbody>
</table>
### Outlook

#### Next Steps

- Further drive harmonization of AUTOSAR Classic and Adaptive Platform until joint Release CP 4.4.0 / AP 2018-10 and beyond
- Continue elaboration of harmonized security concepts
- Working concepts for partial updates/upgrades of software
- Including AUTOSAR Classic Platform ECU
- Synchronization between AUTOSAR Adaptive Platform specification and the respective code
- Further drive the implementation of quality measures for the standard while integrating new features

AUTOSAR is “just” the basis for all the beautiful and great products you can build with it!
Outlook

Next Steps

- Further drive harmonization of AUTOSAR Classic and Adaptive Platform until joint Release CP 4.4.0 / AP 2018-10 and beyond
- Continue elaboration of harmonized security concepts
- Working concepts for partial updates/upgrades of software
- Including AUTOSAR Classic Platform ECUs
- Synchronization between AUTOSAR Adaptive Platform specification and the respective code
- Further drive the implementation of quality measures for the standard while integrating new features

AUTOSAR is “just” the basis for all the beautiful and great products you can build with it!

Make use of the combination of AUTOSAR Classic Platform and Adaptive Platform … and get it alive!
Thank you for your attention!

Nadym Salem, Jan Hegewald
nadym.salem@carmeq.com
jan.hegewald@carmeq.com

Carmeq GmbH
Carnostraße 4
10587 Berlin