Architectural thoughts about management technologies NETCONF and YANG for AUTOSAR-based Software-defined Vehicles
1. SDVs-under-DevOps
   – management of SDVs in operation
   – dynamic management services?

2. Model-based in-vehicle computing AUTOSAR software systems
   – modeling framework for Model-based Systems Engineering (MBSE)
   – AUTOSAR higher level management system

3. Requirements viewpoint
   – system context, actors, goals
   – management architecture pattern "manager-agent"
   – management technologies (NETCONF, YANG)

4. Operational viewpoint
   – management use cases

5. Functional viewpoint
   – service layering, information planes
   – extended data model for AUTOSAR

6. Logical viewpoint
   – vision of a logical AUTOSAR management architecture

7. Technical viewpoint
   – out of scope (no discussion of technical AUTOSAR system and software architectural aspects, would be subject of a follow-up presentation)

8. Summary
   What's next?
AUTOSAR SDV-under-DevOps in Operations and Management
DevOps lifecycle

Software-defined AUTOSAR
Vehicle computing system

1. SDV in-vehicle information system
2. Complementary management system
3. DevOps-life cycle "8" (here ETAS)
4. Unfolded DevOps-life cycle phases
5. Main actors
6. Typical DevOps-life cycle-phase-specific management services
Dynamic management? For SDVs?

Time-dependent frequency of management activities over vehicle lifecycle phases

<table>
<thead>
<tr>
<th>Management activities $f_M(t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Categories:</td>
</tr>
<tr>
<td>I) Managed object lifecycle operations $f_C(t)$</td>
</tr>
<tr>
<td>II) Managed object information retrieval &amp; notifications $f_S(t)$</td>
</tr>
<tr>
<td>Purpose:</td>
</tr>
<tr>
<td>1. create object</td>
</tr>
<tr>
<td>2. delete object</td>
</tr>
<tr>
<td>3. modify or update object</td>
</tr>
<tr>
<td>4. subscribe to notifications</td>
</tr>
<tr>
<td>by manager</td>
</tr>
<tr>
<td>Management data:</td>
</tr>
<tr>
<td>☑ configuration data</td>
</tr>
<tr>
<td>☐ state data</td>
</tr>
</tbody>
</table>

$f_M(t)$ Frequency of overall management activities

$f_C(t)$ Frequency of lifecycle-related management operations

$f_S(t)$ Frequency of management information retrieval & notifications

Temporality of dynamic management services (management service rate). Also dependent whether managed object = 1. unconstrained object or 2. constrained object (like safety constrained).

SDV "x-axis": temporality with respect to
1. addition of new features
2. deletion of existing features
3. hardening of continued features (in order to increase maturity level) ("just think about smart IoT devices")

NOTE: sum $f_M$, $f_C(t)$, $f_S(t)$ not shown.
Model-based Systems Engineering (MBSE) of Model-based in-Vehicle Computing AUTOSAR Software systems

Matrix of Modeling Framework

Figure 10.1: Abstract System Description refactoring to a System Description
Overview – Matrix "Viewpoints vs Abstraction levels"

Location of AUTOSAR system engineering artefacts

Engineering framework:
- Engineering methodology: MBSE (Model-based Systems Engineering)
- MBSE modeling framework: SPES (Software Platform Embedded Systems)
- System modeling languages: SysML, UML
- Data modeling languages: YANG, ARXML

Abstracted as "E/E Reference Architecture"

the usual AUTOSAR "program code abstraction level"

DevOps anchoring

Out of scope.

Good news! That system architecture (levels) are (almost) independent of AUTOSAR! Thus, applicable for all kind of distributed automotive computing systems ☺
Requirements Viewpoint (I)
for an SDV-under-DevOps capable AUTOSAR management architecture
Basic ICT pattern

System context (standard management)

Management architecture pattern – take-aways:
1. Information planes: explicit separation between user and management plane!
2. Representation of management information: explicit management data model!
3. Management actors and roles: management agents and management manager!
4. Results in management plane interfaces: MPIs!
5. Manager-agent(s) communication: management protocol!

Legacy AUTOSAR system (as in-vehicle, distributed computing and communication system)

Addition:
- Manager (vehicle locally or/and remotely operated)
- Management Programming Interface (MPI)

Addition:
- Agent

= resources, objects, entities
⇒ technical components (HW, SW)
⇒ resources (physical, virtual; logical)
Vehicle local, heterogeneous, distributed computing system with management plane

Requirements viewpoint

Takeaway: digital twin = embedded, integral element of management architecture pattern!
Developers, operators, managers, … as humans or/and machines

Level of automation | Vehicle driving control | Vehicle management
--- | --- | ---
L5 | fully automated | 
L4 | 
L3 | 
L2 | 
L1 | 
L0 | fully manual |

Long-term goal "maximal automation" of vehicle management already to be considered in system architecture from day #1!

NOTE 1 – The degree of automation in the vehicle control and management domains are basically orthogonal, are mutually independent.

NOTE 2 – References are e.g., TM Forum Autonomous Networks Reference Architecture (2021), ETSI Zero-touch network and Service Management (2019).

NOTE 3 – Terminology: Assistance translates in Automation support. And Automation should not be confused with Autonomy (which would imply that humans grant machine actors intelligent independence beyond the human assigned constraints; autonomy relates to a concept of freedom).
Top goals & non-goals

Paradigm- and technology-driven architectural goals (non-exhaustive)

Management architecture:
Layered architectures: Functional Service Layering across Technical AUTOSAR Software Layering
Non service-layered model to be related to AUTOSAR system/software layering.
Motivations: destruction and decommissioning of legacy, non-contextual and non-contextual management architecture, existing technical system layers to be reverse engineered in functional service layer.

Management architecture: extension for centralized and distributed management
Introduction of - fully centralized manager(s) in management operation to legacy - fully distributed AUTOSAR management services.
Motivations: distributed systems benefit from centralized intelligence, which simplifies also easy management procedures.

Management architecture: extension for dynamic and static management
Introduction of - in-service, dynamic management services - services logics - out-of-service, planned management services.
Motivations: fully integrated management architecture, applicable for all DoD lifecycle phases and all management use cases.

Evaluation of AUTOSAR (concept) extensions with demand for dynamic management support
Check of current and latest AUTOSAR proposed extensions.
Motivations: current AUTOSAR extension proposal is the basis for future demand concerning in-operation management in demands.

Management data models
Plane separated architectures: explicit division in information planes "user" and "management"
Explicit information plane separation in - UP: user plane - MP: management plane
No resulting in UP- and MP-specific data models.
Motivation: precise information architecture separation, clear division of management overlay architecture.

Management protocols
Vehicle remote (or local) management of invehicle AUTOSAR systems using standardized ICT management protocols as NETCONF, RESTCONF, CORECONF

The above YANG goal may be generalized to other non-ARXML data (like COVESA VSS application data model).

⇒ AUTOSAR process extended by frontend non-ARXML-to-ARXML information converter.

Management protocols:
1. NETCONF is an established, mature, native protocol for professional management of distributed systems.
2. RESTCONF = NETCONF variant for web-based management.
3. CORECONF = NETCONF variant for constrained devices.
⇒ all automotive use cases could be fully covered!
Goals & non-goals

SDV-driven, in-service, dynamic management goals (examples, non-exhaustive)

(I.5.2) Requirements viewpoint

SDVs-under-DevOps-under-Management
Scope on dynamic manageable SDV
(excluding SDV modules, etc.)

SDV-in-Stop-phases
development artefact, parking vehicle, etc.

Stop

SDV-in-Motion-phases
vehicle driving

Legacy AUTOSAR quadrant: pre-operation (static) management

Management goals (some examples) for in-operation (dynamic) management

DevOps Lifecycle I
SDV-under-Development-under-Static-Management
static

DevOps Lifecycle II
SDV-in-Operation-under-Dynamic-Management
dynamic

SDV-in-Stop-phases
development artefact, parking vehicle, etc.

Communication matrix definition of connection topology

For all:
- permanent connections.
Not for:
- semi-permanent connections.
- signalized connections.
Status see AUTOSAR System Template, clause 6 Communications.

DevOps Lifecycle II
SDV-in-Operation-under-Dynamic-Management
dynamic

Operational phase “Charging”: high-bits related management services
given the satellite high-speed V2X (vehicle-to-everything communication) connectivity and communication interface capacity.

Legacy AUTOSAR quadrant: pre-operation (static) management

Management goals (some examples) for in-operation (dynamic) management

SDV-driven, in-service, dynamic management goals (examples, non-exhaustive)
Operational Viewpoint (II) for an SDV-under-DevOps capable AUTOSAR management architecture
example use cases (non-exhaustive)

(II.2) Operational viewpoint

Take-aways:
- there are SDV-in-Mobility-Service (= in-operation management use cases)
- in an order beyond legacy dynamic management

Reminder: constraint-dependent configuration management!
E.g., distribution between:
1. unstructured objects
2. constrained objects
   a) quality-constrained objects (like QoS, safety, security constructed) and/or
   b) resource-constrained objects (from management perspective).
=> constraint category dependent management use cases!
Functional viewpoint (III)

⇒ Functional service layering versus technical AUTOSAR software layering
⇒ Functional management architecture for vehicle local and/or remote management
Service layering

over the distributed vehicle computing & communication system, incl. management plane

(III.1) Functional viewpoint

1. Introduction and support of standard ICT communication technologies (Internet Protocol (IETF) and Ethernet (IEEE)) in AUTOSAR, in relation to their native ICT management information representation by (IETF, IEEE) YANG data models.

2. Automotive TSN and the relation between IEEE TSN YANG data models and AUTOSAR ARXML data models.

3. Existing in-vehicle fully distributed management architectures (AUTOSAR) and the possible direction in centralizing specific management services, particular in IEEE 802.1 TSN and IEEE 1722 AVB communication context.
(III.2) Functional viewpoint

YANG-to-ARXML integration

Legacy AUTOSAR
1. system development process = unchanged!
2. mature tool chain = unchanged!
⇒ no revolution of process and development tools!
⇒ no revolution of AUTOSAR system architecture!

AUTOSAR extended:
⇒ standardized, open YANG data models

similar for application data models, e.g., COVESA VSS → AUTOSAR extended:
⇒ AUTOSAR concept Vehicle API
⇒ mapping VSS-to-ARXML

NOTE - The notion of AR-XML and YANG-XML intends to underline the common syntactical language baseline.
Logical viewpoint (IV)

Functional management architecture for vehicle local and/or remote management
Hybrid centralized/distributed management managers

**Managers:**
- Might be realized technically in a centralized or distributed manner (dependent on your logical-to-technical deployment model).
- Each manager is responsible for his service layer related objects.
- Manager "APP" and "COM & RES" are termed as "User" and "Network" managers in IEEE (e.g., 802.1, 60802 and CUC/CNC for centralized deployment variant).
- Long-term goal: merged, single and centralized manager (in small area networks like in-vehicle distributed computing systems).

**Manager-Agent interfaces (MPI):**
- Management data transfer...
- … over a management protocol.

**Agents:**
- The two logical agents may be technically realized by a single one (even if there are multiple managers).
- Example shows two for the discussion of the combination of a centralized manager COM & RES and a distributed manager APP.
  - E.g., as a first evolution path step from a fully distributed manager (like in legacy AUTOSAR), e.g., IEEE 60802 follows a similar evolution strategy (for manager and agents).

**Take-aways:**
- A single, common logical management architecture may serve many technical deployment variants.
- Logical user and management plane architecture allows to discuss e.g., dynamic establishment/release of end-to-end communication services (like TSN streams inclusive entire QoS and resource management).

**Reminder!**
Exemplification indicates only a few example, important model elements (in order not to overload the illustration).
Summary
What’s next?
Model-based operation and management of SDVs = Model-based Systems Engineering!

Conclusions:
- Overlay management architecture engineered
- Candidate entry point(s) for YANG data models identified
- Candidate management protocol NETCONF integration via manager-agent pattern

Next steps:
- … follow the MBSE development path …

Model-based engineering:
- Model-based AUTOSAR software engineering implies
- Model-based system engineering first (due to system-embedded software)

Summary
Frankly speaking, you might replace "AUTOSAR" by any general in-vehicle computing systems, whether for their operation or even the development.

Model-based operation and management of SDVs

<table>
<thead>
<tr>
<th>Requirements Viewpoint</th>
<th>Operational Viewpoint</th>
<th>Functional Viewpoint</th>
<th>Logical Viewpoint</th>
<th>Technical Viewpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstraction level I &quot;DevOps stakeholders&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System Context of Generic Management System (ICT Patterns)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>There is no case at that level where this applies</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AUTOSAR Software &amp; Operations Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>There are no logical components at that level</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>There is no logical component at that level due to the lack of a logical system architecture, thus no logical to technical mapping</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Abstraction level II &quot;System Engineering&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System Context with Management Plan Overlay</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DevOps Manager for All DevOps Processes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AUTOSAR Architecture Evolution Goals</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Particular System &amp; Software Modeling Languages</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integration of YANG in AUTOSAR &amp; ARXML data models</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logical AUTOSAR Management Architecture</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOTE: An abstract technical system architecture and a candidate management architecture could be developed, but out of scope</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data models vs. atomic AUTOSAR logical concepts</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AUTOSAR &quot;Functional Viewpoint&quot; in Functional (vertical) Systems Description (not part of functional - logical space)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AUTOSAR &quot;Technical Viewpoint&quot; in Technical (horizontal) Systems Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Modeling languages:
- System & Software: SysML, UML
- Information & Data: YANG, ARXML, …

External | ETAS DAP | 2023-07-28
Thank you!

Dr. Wolfgang Hauck (ETAS SDV.DAP, Reference Architecture)
Dr. Albrecht Schwarz (ETAS SDV.DAP, Reference Architecture)

2023 IEEE SA Ethernet & IP @ Automotive Technology Day (E&IP@ATD)