Architectural thoughts about management technologies NETCONF and YANG for AUTOSAR-based Software-defined Vehicles

AUTOSAR	AUTomotive Open System ARchitecture www.autosar.org
IETF	Internet Engineering Task Force
NETCONF	a) Network Configuration Protocol b) IETF Working Group "Network Configuration"
YANG	Yet-Another-Next-Generation = name of a management data modeling language

Storyline

Walkthrough

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	AUTomotive Open System ARchitecture (www.autosar.org)
DevOps	Lifecycle model "Development / Operations"
MBSE	Model-based Systems Engineering
NETCONF	a) Network Configuration Protocol b) IETF Working Group "Network Configuration"
SDV	Software-defined Vehicle
YANG	Yet-Another-Next-Generation = name of a management data modeling language

AUTOSAR SDV-under-DevOps

in Operations and Management

AUTOSAR	AUTomotive Open System ARchitecture
DevOps	Lifecycle model "Development / Operations"
SDV	Software-defined Vehicle

DevOps lifecycle

Software-defined AUTOSAR Vehicle computing system

(3) DevOps-lifecycle "8" (here ETAS)

(4) unfolded DevOps-lifecycle phases

(5) main actors

(2) complementary management system

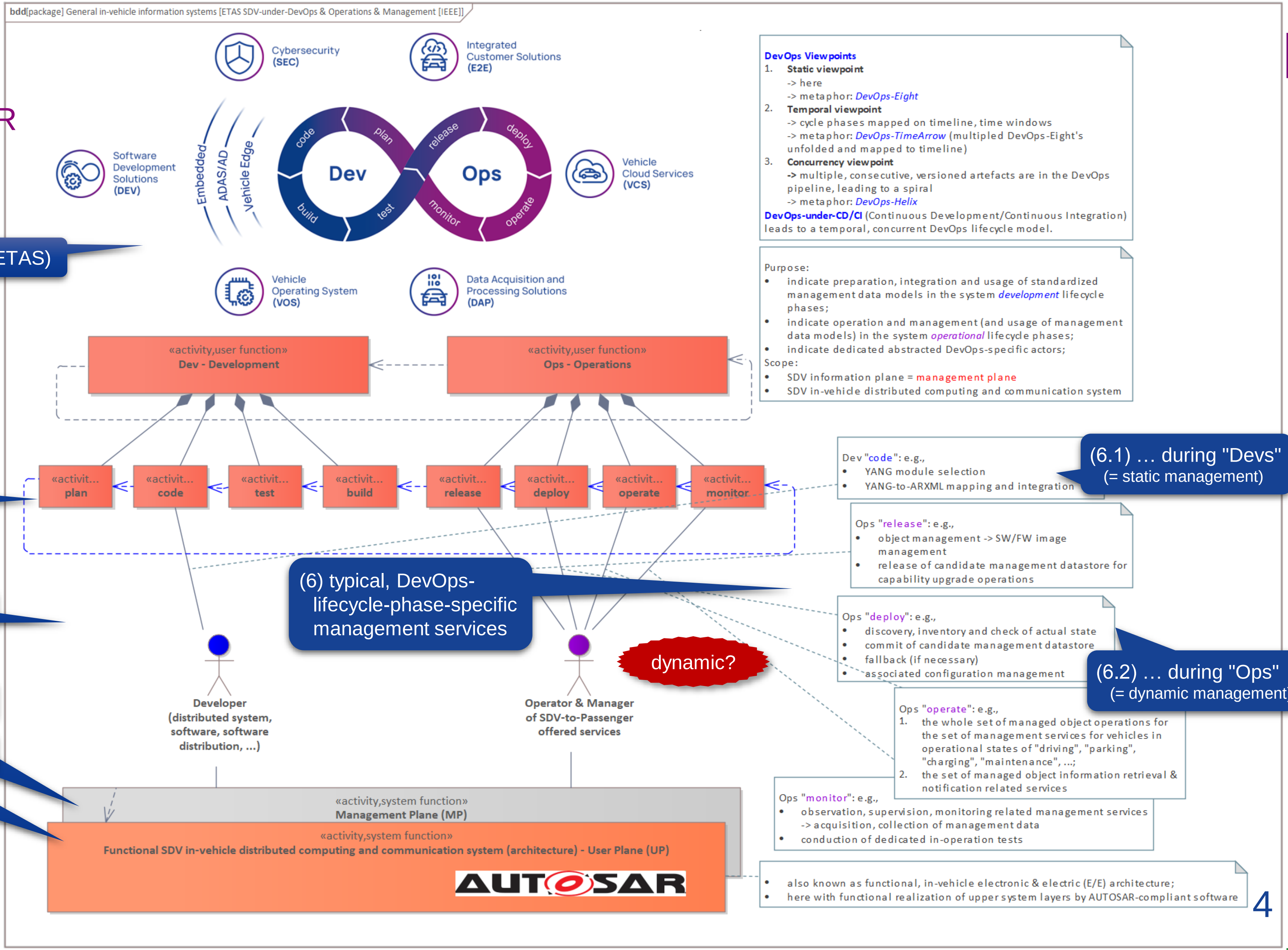
(1) SDV in-vehicle information system

(6) typical, DevOps-lifecycle-phase-specific management services

(6.1) … during "Devs" (= static management)

(6.2) … during "Ops" (= dynamic management)

dynamic?


bdd[package] General in-vehicle information systems [ETAS SDV-under-DevOps & Operations & Management [IEEE]]
Cybersecurity (SEC)
Integrated Customer Solutions (E2E)
Software Development Solutions (DEV)
Embedded
ADAS/AD
Vehicle Edge
Dev
Ops
code
plan
build
test
release
deploy
monitor
operate
Vehicle Cloud Services (VCS)
Vehicle Operating System (VOS)
Data Acquisition and Processing Solutions (DAP)
«activity,user function» Dev - Development
«activity,user function» Ops - Operations
«activit... plan
«activit... code
«activit... test
«activit... build
«activit... release
«activit... deploy
«activit... operate
«activit... monitor
Developer (distributed system, software, software distribution, ...)
Operator & Manager of SDV-to-Passenger offered services
«activity,system function» Management Plane (MP)
«activity,system function» Functional SDV in-vehicle distributed computing and communication system (architecture) - User Plane (UP)
AUTOSAR

DevOps Viewpoints

1. **Static viewpoint**
   -> here
   -> metaphor: *DevOps-Eight*
2. **Temporal viewpoint**
   -> cycle phases mapped on timeline, time windows
   -> metaphor: *DevOps-TimeArrow* (multipled DevOps-Eight's unfolded and mapped to timeline)
3. **Concurrency viewpoint**
   -> multiple, consecutive, versioned artefacts are in the DevOps pipeline, leading to a spiral
   -> metaphor: *DevOps-Helix*

DevOps-under-CD/CI (Continuous Development/Continuous Integration) leads to a temporal, concurrent DevOps lifecycle model.

Purpose:
- indicate preparation, integration and usage of standardized management data models in the system *development* lifecycle phases;
- indicate operation and management (and usage of management data models) in the system *operational* lifecycle phases;
- indicate dedicated abstracted DevOps-specific actors;

Scope:
- SDV information plane = management plane
- SDV in-vehicle distributed computing and communication system

Dev "code": e.g.,
- YANG module selection
- YANG-to-ARXML mapping and integration

Ops "release": e.g.,
- object management -> SW/FW image management
- release of candidate management datastore for capability upgrade operations

Ops "deploy": e.g.,
- discovery, inventory and check of actual state
- commit of candidate management datastore
- fallback (if necessary)
- associated configuration management

Ops "operate": e.g.,
1. the whole set of managed object operations for the set of management services for vehicles in operational states of "driving", "parking", "charging", "maintenance", ...;
2. the set of managed object information retrieval & notification related services

Ops "monitor": e.g.,
- observation, supervision, monitoring related management services -> acquisition, collection of management data
- conduction of dedicated in-operation tests

- also known as functional, in-vehicle electronic & electric (E/E) architecture;
- here with functional realization of upper system layers by AUTOSAR-compliant software

AUTOSAR	AUTomotive Open System ARchitecture
DevOps	Lifecycle model "Development / Operations"
FW / SW	Firmware, Software
SDV	Software-defined Vehicle

Dynamic management? For SDVs?

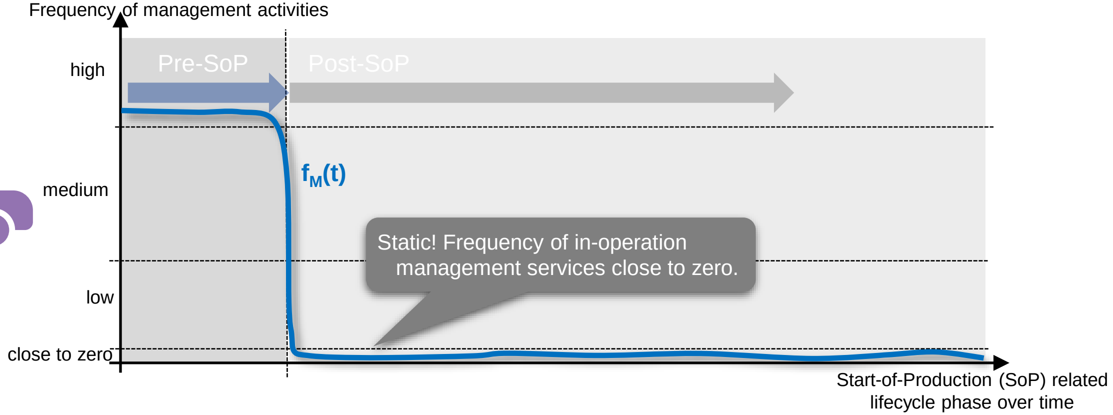
Time-dependent frequency of management activities over vehicle lifecycle phases

	Management activities $f_M(t)$	
Categories:	I) Managed object life-cycle operations $f_C(t)$	II) Managed object information retrieval & notifications $f_S(t)$
Purpose:	1. create object 2. delete object 3. modify or update object 4. subscribe to notifications by manager	1. retrieve (or read) information from agent by manager 2. notify information to manager by agent besides basic 3. discovery and inventory of actual capabilities and features
Manage-ment data:	☑ **configuration data** ☐ state data	☑ configuration data ☑ **state data**

$f_M(t)$ Frequency of overall management activities
$f_C(t)$ Frequency of lifecycle-related management operations
$f_S(t)$ Frequency of management information retrieval & notification

HWV

Hard-wired Vehicle

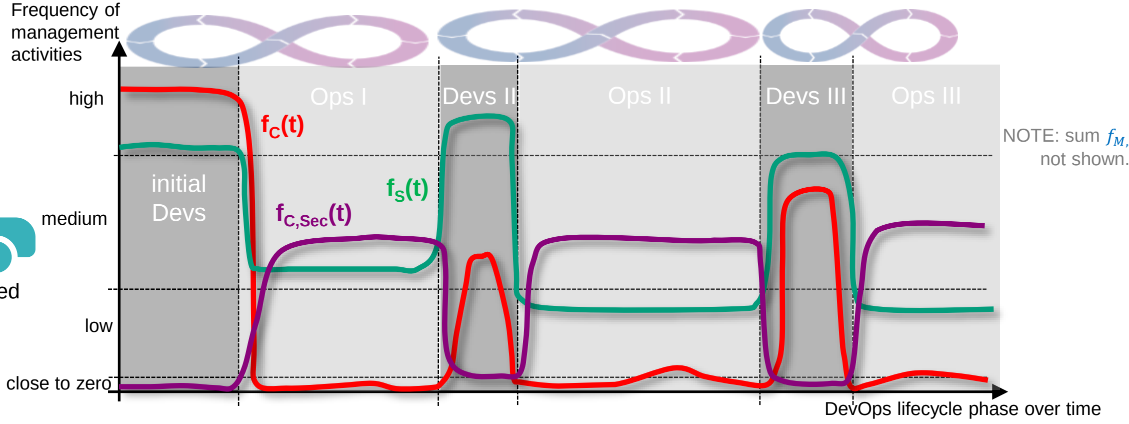


Static! Frequency of in-operation management services close to zero.

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

SDV

Software-defined Vehicle



NOTE: sum $f_{M,SDV}(t)$ not shown.

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")

Example management activity:
$f_{C,Sec}(t)$ Frequency of security management operations
(e.g., policy management → security policy management → update of security policy rules)

HWV	Hard-wired Vehicle	IoT	Internet of Things
MO	Managed Object	SDV	Software-defined Vehicle

Model-based Systems Engineering (MBSE) of Model-based in-Vehicle Computing AUTOSAR Software systems

Matrix of Modeling Framework

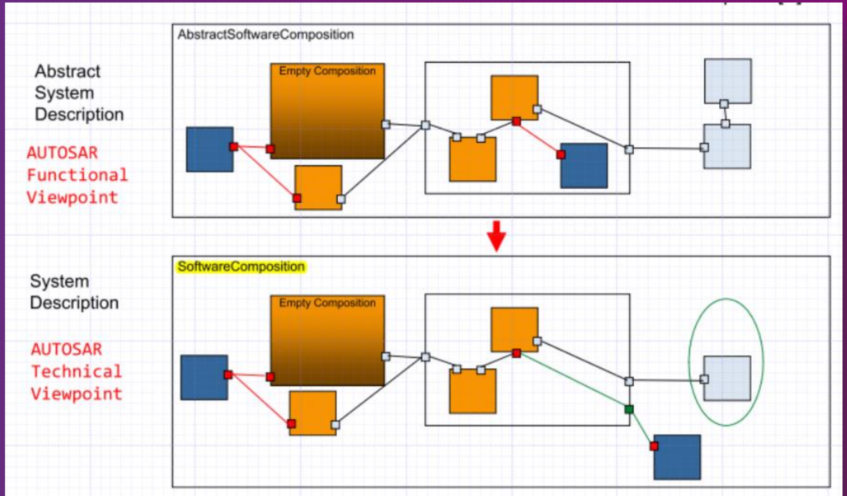

AbstractSoftwareComposition
Abstract System Description
AUTOSAR Functional Viewpoint
Empty Composition
SoftwareComposition
System Description
AUTOSAR Technical Viewpoint
Empty Composition

Figure 10.1: Abstract System Description refactoring to a System Description

Overview – Matrix "Viewpoints vs Abstraction levels"

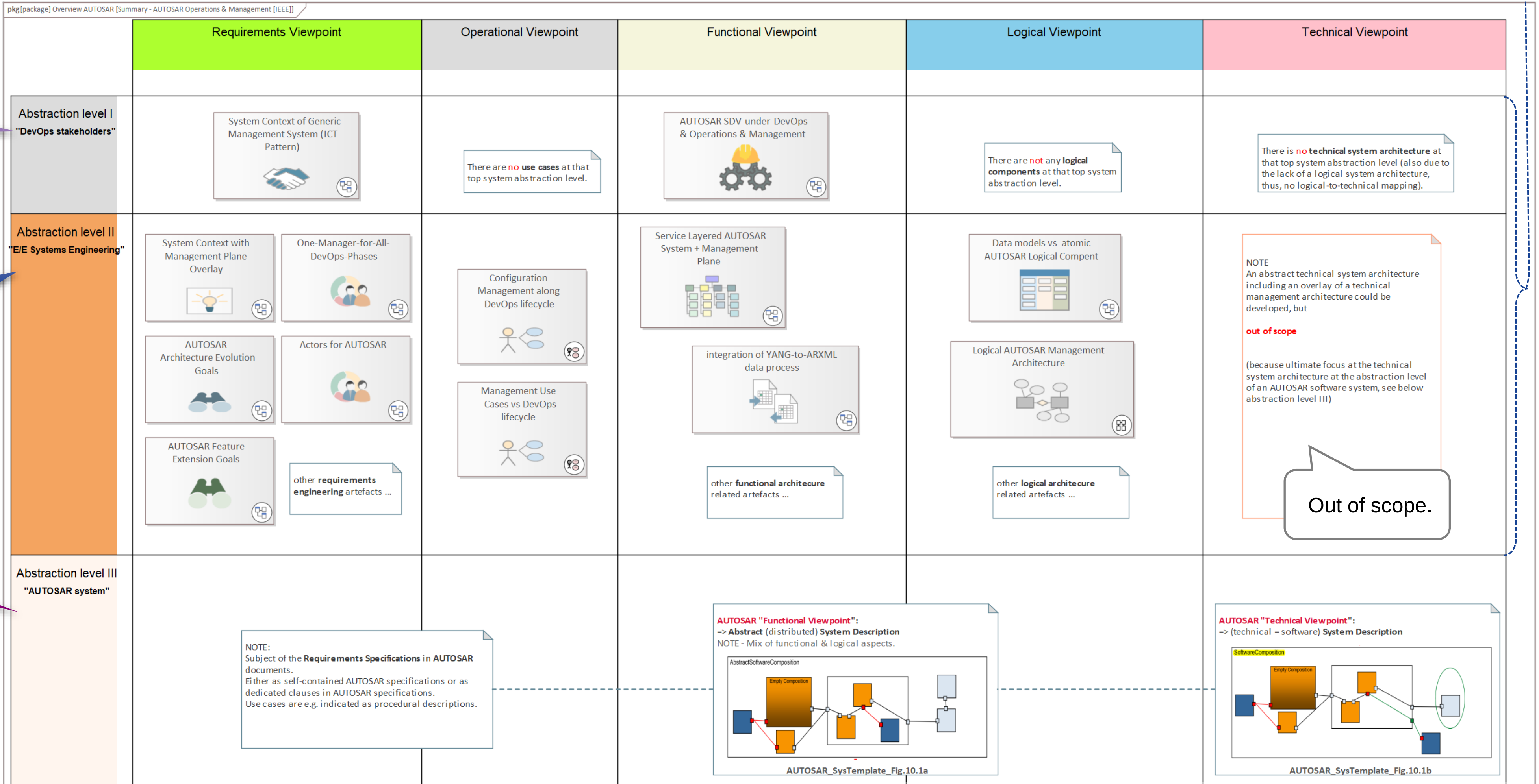
Location of AUTOSAR system engineering artefacts

Good news!
That system architecture (levels) are (almost) independent of AUTOSAR!
Thus, applicable for all kind of distributed automotive computing systems …☺

DevOps anchoring

Abstracted as "E/E Reference Architecture"

the usual AUTOSAR "program code abstraction level"



pkg [package] Overview AUTOSAR [Summary - AUTOSAR Operations & Management [IEEE]]

	Requirements Viewpoint	Operational Viewpoint	Functional Viewpoint	Logical Viewpoint	Technical Viewpoint
Abstraction level I **"DevOps stakeholders"**	System Context of Generic Management System (ICT Pattern)	There are no **use cases** at that top system abstraction level.	AUTOSAR SDV-under-DevOps & Operations & Management	There are not any **logical components** at that top system abstraction level.	There is no **technical system architecture** at that top system abstraction level (also due to the lack of a logical system architecture, thus, no logical-to-technical mapping).
Abstraction level II **"E/E Systems Engineering"**	System Context with Management Plane Overlay One-Manager-for-All-DevOps-Phases AUTOSAR Architecture Evolution Goals Actors for AUTOSAR AUTOSAR Feature Extension Goals other **requirements engineering** artefacts ...	Configuration Management along DevOps lifecycle Management Use Cases vs DevOps lifecycle	Service Layered AUTOSAR System + Management Plane integration of YANG-to-ARXML data process other **functional architecture** related artefacts ...	Data models vs atomic AUTOSAR Logical Compent Logical AUTOSAR Management Architecture other **logical architecture** related artefacts ...	NOTE An abstract technical system architecture including an overlay of a technical management architecture could be developed, but out of scope (because ultimate focus at the technical system architecture at the abstraction level of an AUTOSAR software system, see below abstraction level III) Out of scope.
Abstraction level III **"AUTOSAR system"**	NOTE: Subject of the **Requirements Specifications** in **AUTOSAR** documents. Either as self-contained AUTOSAR specifications or as dedicated clauses in AUTOSAR specifications. Use cases are e.g. indicated as procedural descriptions.		AUTOSAR "Functional Viewpoint": => **Abstract** (distributed) **System Description** NOTE - Mix of functional & logical aspects. AUTOSAR_SysTemplate_Fig.10.1a		AUTOSAR "Technical Viewpoint": => (technical = software) **System Description** AUTOSAR_SysTemplate_Fig.10.1b

Engineering framework:

- Engineering methodology:
  = MBSE (Model-based Systems Engineering)
- MBSE modeling framework:
  = SPES (Software Platform Embedded Systems)

  see *Model-Based Engineering of Embedded Systems: The SPES 2020 Methodology*, Springer, 2014.
- System modeling languages:
  = SysML, UML
- Data modeling languages:
  = YANG, ARXML

ARXML	AUTOSAR XML
XML	Extensible Markup Language
YANG	Data modeling language "Yet-Another-Next-Generation"
E/E	Electrical & Electronic (in-vehicle architecture)

Requirements Viewpoint (I)

for an SDV-under-DevOps capable AUTOSAR management architecture

AUTOSAR	AUTomotive Open System ARchitecture
DevOps	Lifecycle model "Development / Operations"
SDV	Software-defined Vehicle

System context (standard management)

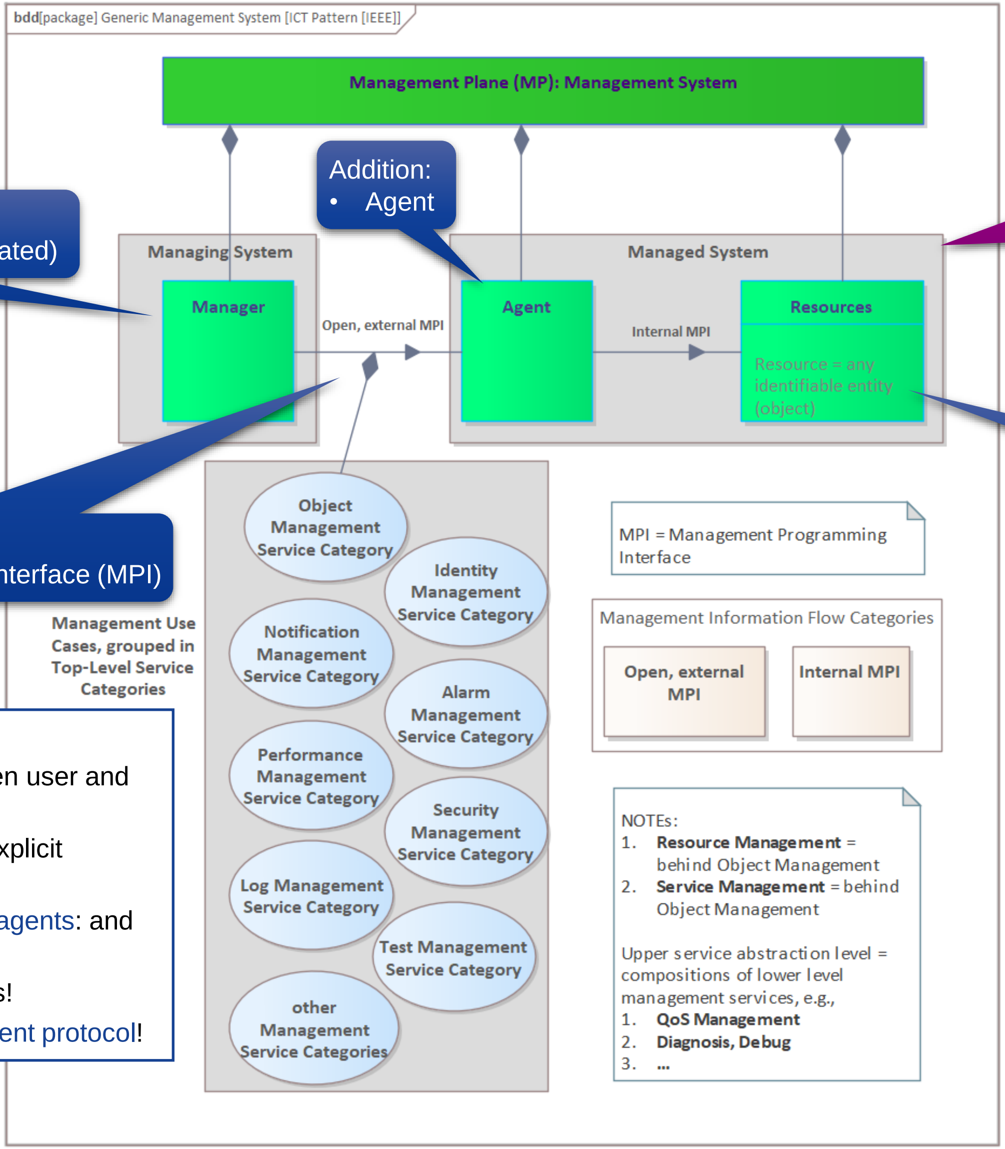
Basic ICT pattern

ICT Information & Communication Technologies

Addition:
- Manager (vehicle locally or/and remotely operated)

Addition:
- Agent

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



bdd[package] Generic Management System [ICT Pattern [IEEE]]
Management Plane (MP): Management System
Managing System
Managed System
Manager
Open, external MPI
Agent
Internal MPI
Resources
Resource = any identifiable entity (object)
Object Management Service Category
Identity Management Service Category
Notification Management Service Category
Alarm Management Service Category
Performance Management Service Category
Security Management Service Category
Log Management Service Category
Test Management Service Category
other Management Service Categories
Management Use Cases, grouped in Top-Level Service Categories
Management Information Flow Categories
Open, external MPI
Internal MPI

MPI = Management Programming Interface

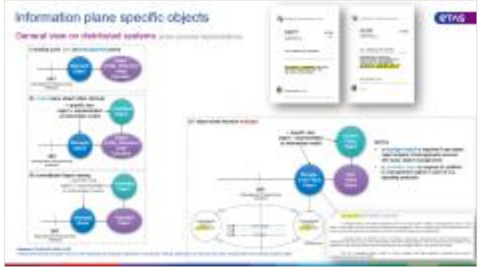
NOTEs:
1. **Resource Management** = behind Object Management
2. **Service Management** = behind Object Management

Upper service abstraction level = compositions of lower level management services, e.g.,
1. **QoS Management**
2. **Diagnosis, Debug**
3. **…**

Addition:
- Management Programming Interface (MPI)

= resources, objects, entities
⇒ technical components {HW, SW}
⇒ resources {physical, virtual; logical}

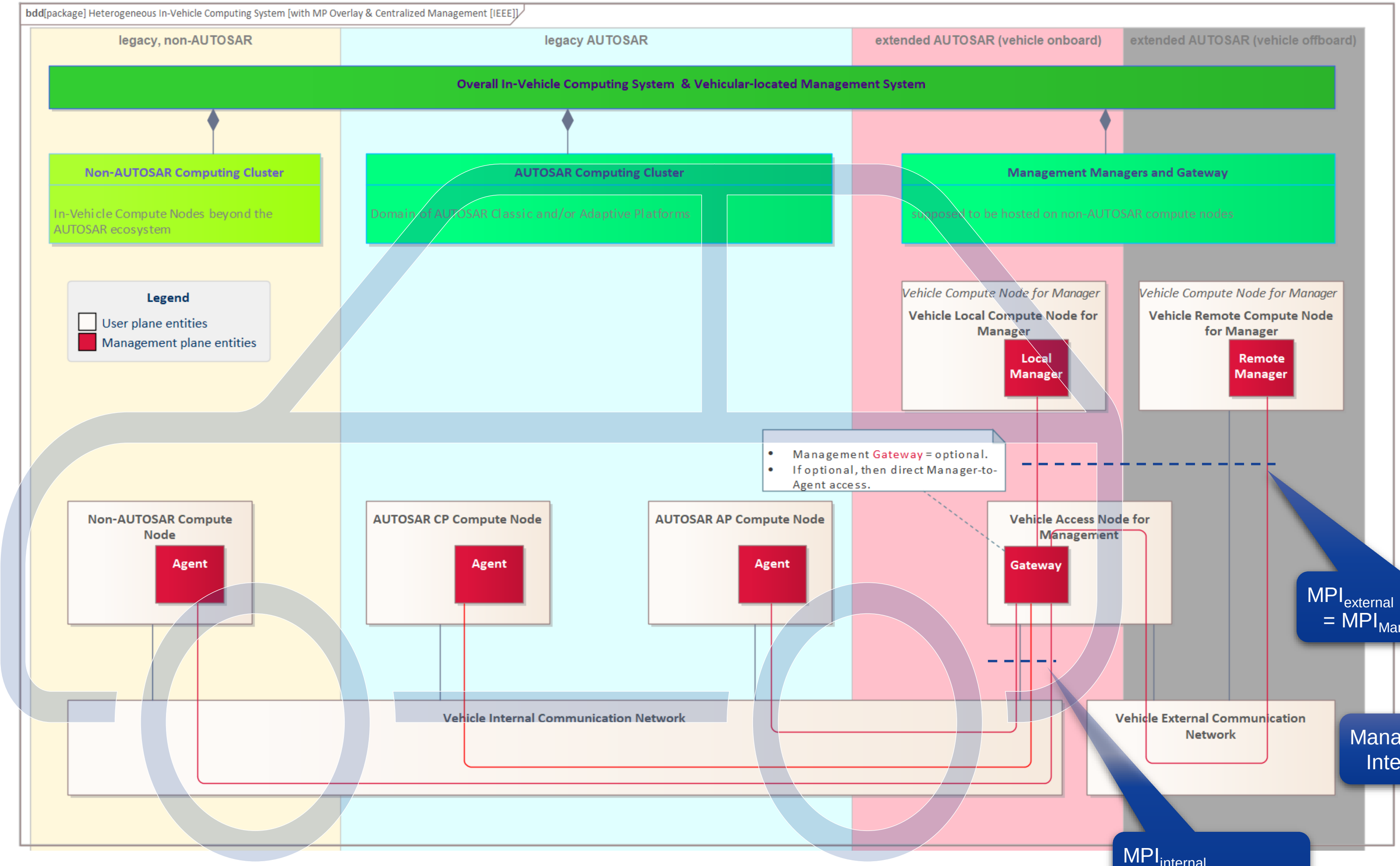
Management architecture pattern – take-aways:
1. information planes: explicit separation between user and management plane!
2. representation of management information: explicit management data model!
3. management actors and roles: management agents: and management manager!
4. results in management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



API	Programming Interface for user Applications
MPI	Programming Interface for Management applications
QoS	Quality of Service

System context (operations)

Vehicle local, heterogeneous, distributed computing system with management plane

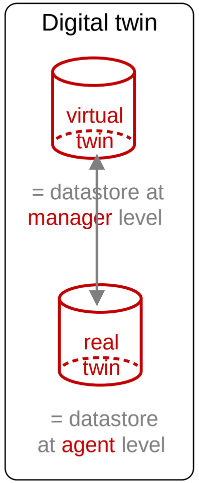

bdd[package] Heterogeneous In-Vehicle Computing System [with MP Overlay & Centralized Management [IEEE]]
legacy, non-AUTOSAR
legacy AUTOSAR
extended AUTOSAR (vehicle onboard)
extended AUTOSAR (vehicle offboard)
Overall In-Vehicle Computing System & Vehicular-located Management System
Non-AUTOSAR Computing Cluster
In-Vehicle Compute Nodes beyond the AUTOSAR ecosystem
AUTOSAR Computing Cluster
Domain of AUTOSAR Classic and/or Adaptive Platforms
Management Managers and Gateway
supposed to be hosted on non-AUTOSAR compute nodes
Legend
□ User plane entities
🟥 Management plane entities
Vehicle Compute Node for Manager
Vehicle Local Compute Node for Manager
Local Manager
Vehicle Compute Node for Manager
Vehicle Remote Compute Node for Manager
Remote Manager
Management Gateway = optional.
If optional, then direct Manager-to-Agent access.
Non-AUTOSAR Compute Node
Agent
AUTOSAR CP Compute Node
Agent
AUTOSAR AP Compute Node
Agent
Vehicle Access Node for Management
Gateway
Vehicle Internal Communication Network
Vehicle External Communication Network

$\text{MPI}_{\text{external}} = \text{MPI}_{\text{Manager-Gateway|Agent}}$

Management Programming Interfaces (MPI)

$\text{MPI}_{\text{internal}} = \text{MPI}_{\text{Gateway-Agent}}$

Takeaway: digital twin
= embedded, integral element of management architecture pattern!

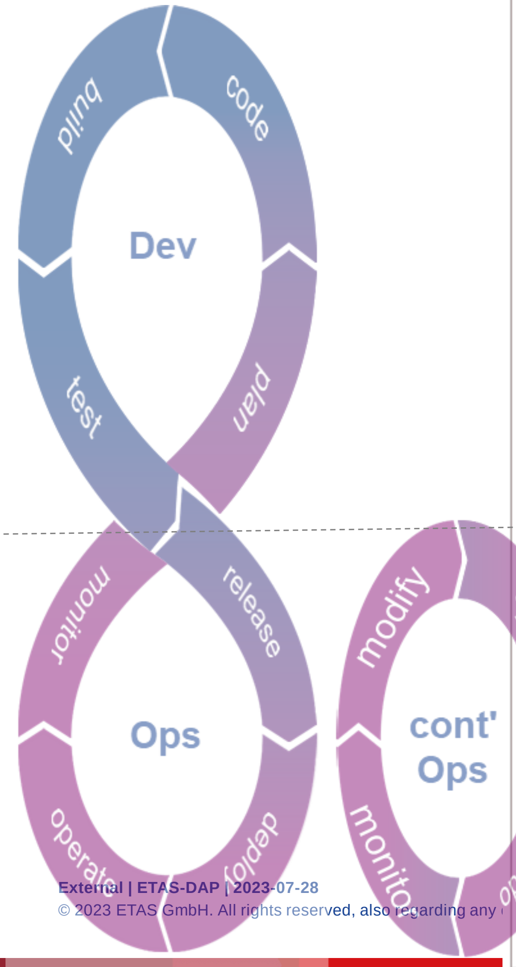


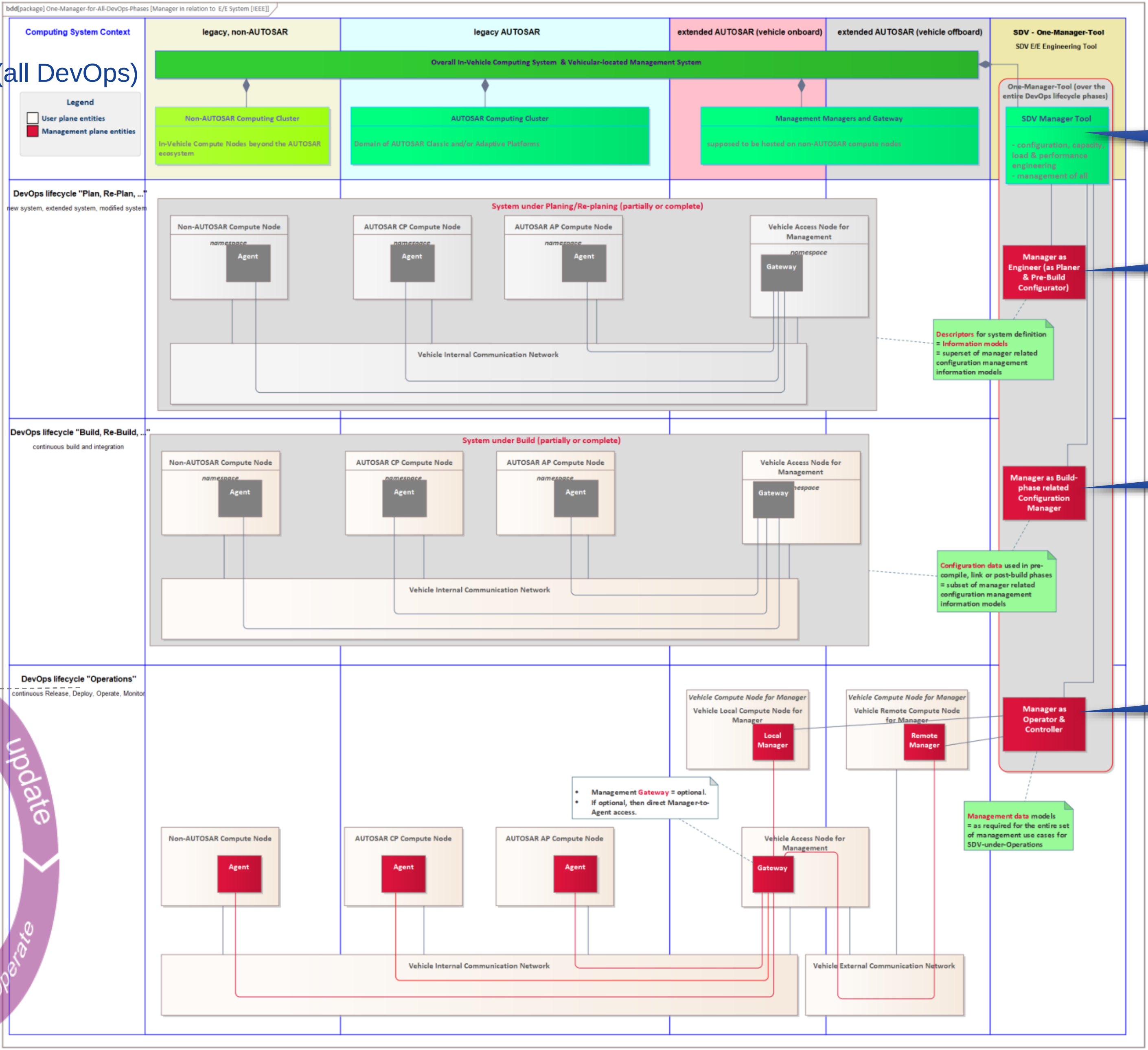
AP	(AUTOSAR) Adaptive Platform
CP	(AUTOSAR) Classical Platform
MPI	Management Programming Interface

System context (all DevOps)

One-Manager-Tool

AP	(AUTOSAR) Adaptive Platform
CP	(AUTOSAR) Classical Platform
MPI	Management Programming Interface





bdd[package] One-Manager-for-All-DevOps-Phases [Manager in relation to E/E System [IEEE]]
Computing System Context
legacy, non-AUTOSAR
legacy AUTOSAR
extended AUTOSAR (vehicle onboard)
extended AUTOSAR (vehicle offboard)
SDV - One-Manager-Tool
SDV E/E Engineering Tool
Legend
User plane entities
Management plane entities
Overall In-Vehicle Computing System & Vehicular-located Management System
Non-AUTOSAR Computing Cluster
In-Vehicle Compute Nodes beyond the AUTOSAR ecosystem
AUTOSAR Computing Cluster
Domain of AUTOSAR Classic and/or Adaptive Platforms
Management Managers and Gateway
supposed to be hosted on non-AUTOSAR compute nodes
One-Manager-Tool (over the entire DevOps lifecycle phases)
SDV Manager Tool
- configuration, capacity, load & performance engineering
- management of all
DevOps lifecycle "Plan, Re-Plan, ..."
new system, extended system, modified system
System under Planing/Re-planing (partially or complete)
Non-AUTOSAR Compute Node
AUTOSAR CP Compute Node
AUTOSAR AP Compute Node
Vehicle Access Node for Management
namespace
Agent
Gateway
Vehicle Internal Communication Network
Manager as Engineer (as Planer & Pre-Build Configurator)
DevOps lifecycle "Build, Re-Build, ..."
continuous build and integration
System under Build (partially or complete)
Manager as Build-phase related Configuration Manager
DevOps lifecycle "Operations"
continuous Release, Deploy, Operate, Monitor
Vehicle Compute Node for Manager
Vehicle Local Compute Node for Manager
Local Manager
Vehicle Remote Compute Node for Manager
Remote Manager
Manager as Operator & Controller
Vehicle External Communication Network

Descriptors for system definition
= Information models
= superset of manager related configuration management information models

Configuration data used in pre-compile, link or post-build phases
= subset of manager related configuration management information models

- Management Gateway = optional.
- If optional, then direct Manager-to-Agent access.

Management data models
= as required for the entire set of management use cases for SDV-under-Operations

One manager, used …

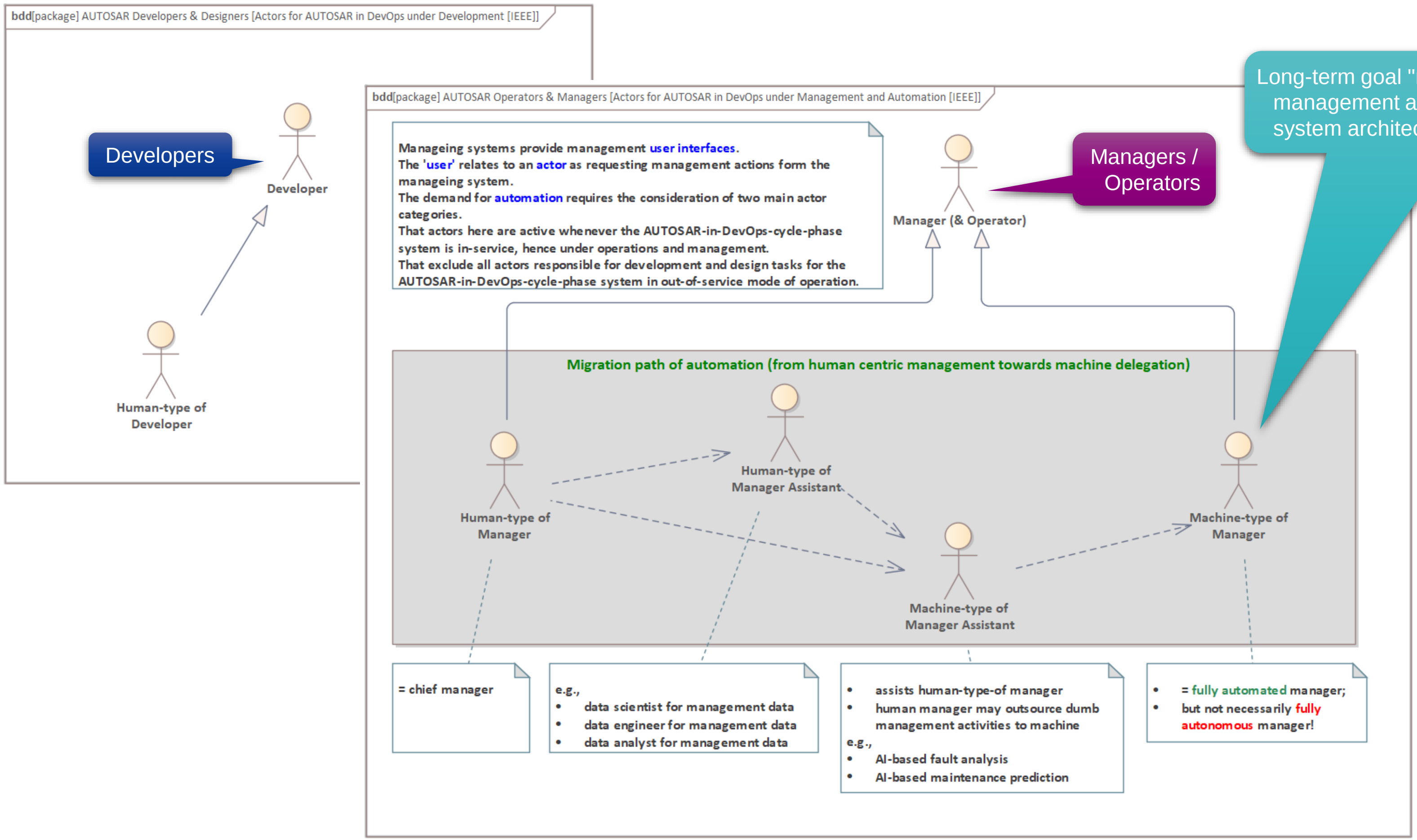
… for planing

… for build

… for operation

Actors

Developers, operators, managers, … as humans or/and machines


bdd[package] AUTOSAR Developers & Designers [Actors for AUTOSAR in DevOps under Development [IEEE]]
Developers
Developer
Human-type of Developer
bdd[package] AUTOSAR Operators & Managers [Actors for AUTOSAR in DevOps under Management and Automation [IEEE]]
Manager (& Operator)
Managers / Operators
Migration path of automation (from human centric management towards machine delegation)
Human-type of Manager
Human-type of Manager Assistant
Machine-type of Manager Assistant
Machine-type of Manager

Manageing systems provide management user interfaces.
The 'user' relates to an actor as requesting management actions form the manageing system.
The demand for automation requires the consideration of two main actor categories.
That actors here are active whenever the AUTOSAR-in-DevOps-cycle-phase system is in-service, hence under operations and management.
That exclude all actors responsible for development and design tasks for the AUTOSAR-in-DevOps-cycle-phase system in out-of-service mode of operation.

= chief manager

e.g.,
- data scientist for management data
- data engineer for management data
- data analyst for management data

- assists human-type-of manager
- human manager may outsource dumb management activities to machine

e.g.,
- AI-based fault analysis
- AI-based maintenance prediction

- = fully automated manager;
- but not necessarily fully autonomous manager!

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

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

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

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

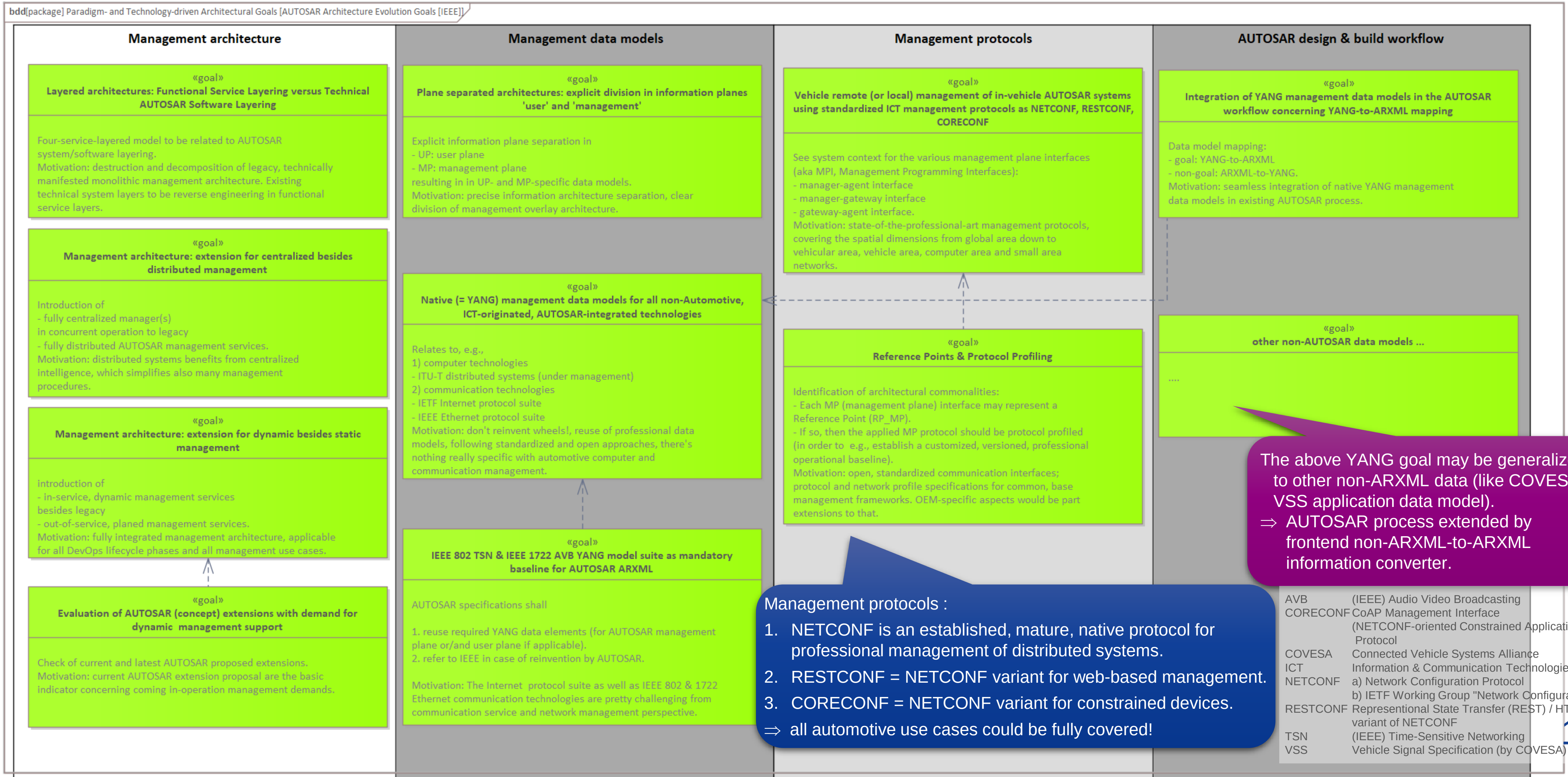
NOTE 3 – Terminology: *Assistance* translates in *Automation* support. And Automation should not be confused with *Autonomy* (which would imply that humans grant machine actors intelligent independence beyond the human assigned constraints; autonomy relates to a concept of freedom).

AI	Artificial Intelligence
ETSI	European Telecommunications Standards Institute

Top goals & non-goals

(I.5.1) Requirements viewpoint

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



bdd[package] Paradigm- and Technology-driven Architectural Goals [AUTOSAR Architecture Evolution Goals [IEEE]]

Management architecture

«goal»
Layered architectures: Functional Service Layering versus Technical AUTOSAR Software Layering

Four-service-layered model to be related to AUTOSAR system/software layering.
Motivation: destruction and decomposition of legacy, technically manifested monolithic management architecture. Existing technical system layers to be reverse engineering in functional service layers.

«goal»
Management architecture: extension for centralized besides distributed management

Introduction of
- fully centralized manager(s)
in concurrent operation to legacy
- fully distributed AUTOSAR management services.
Motivation: distributed systems benefits from centralized intelligence, which simplifies also many management procedures.

«goal»
Management architecture: extension for dynamic besides static management

introduction of
- in-service, dynamic management services
besides legacy
- out-of-service, planed management services.
Motivation: fully integrated management architecture, applicable for all DevOps lifecycle phases and all management use cases.

«goal»
Evaluation of AUTOSAR (concept) extensions with demand for dynamic management support

Check of current and latest AUTOSAR proposed extensions.
Motivation: current AUTOSAR extension proposal are the basic indicator concerning coming in-operation management demands.

Management data models

«goal»
Plane separated architectures: explicit division in information planes 'user' and 'management'

Explicit information plane separation in
- UP: user plane
- MP: management plane
resulting in in UP- and MP-specific data models.
Motivation: precise information architecture separation, clear division of management overlay architecture.

«goal»
Native (= YANG) management data models for all non-Automotive, ICT-originated, AUTOSAR-integrated technologies

Relates to, e.g.,
1) computer technologies
- ITU-T distributed systems (under management)
2) communication technologies
- IETF Internet protocol suite
- IEEE Ethernet protocol suite
Motivation: don't reinvent wheels!, reuse of professional data models, following standardized and open approaches, there's nothing really specific with automotive computer and communication management.

«goal»
IEEE 802 TSN & IEEE 1722 AVB YANG model suite as mandatory baseline for AUTOSAR ARXML

AUTOSAR specifications shall

1. reuse required YANG data elements (for AUTOSAR management plane or/and user plane if applicable).
2. refer to IEEE in case of reinvention by AUTOSAR.

Motivation: The Internet protocol suite as well as IEEE 802 & 1722 Ethernet communication technologies are pretty challenging from communication service and network management perspective.

Management protocols

«goal»
Vehicle remote (or local) management of in-vehicle AUTOSAR systems using standardized ICT management protocols as NETCONF, RESTCONF, CORECONF

See system context for the various management plane interfaces (aka MPI, Management Programming Interfaces):
- manager-agent interface
- manager-gateway interface
- gateway-agent interface.
Motivation: state-of-the-professional-art management protocols, covering the spatial dimensions from global area down to vehicular area, vehicle area, computer area and small area networks.

«goal»
Reference Points & Protocol Profiling

Identification of architectural commonalities:
- Each MP (management plane) interface may represent a Reference Point (RP_MP).
- If so, then the applied MP protocol should be protocol profiled (in order to e.g., establish a customized, versioned, professional operational baseline).
Motivation: open, standardized communication interfaces; protocol and network profile specifications for common, base management frameworks. OEM-specific aspects would be part extensions to that.

AUTOSAR design & build workflow

«goal»
Integration of YANG management data models in the AUTOSAR workflow concerning YANG-to-ARXML mapping

Data model mapping:
- goal: YANG-to-ARXML
- non-goal: ARXML-to-YANG.
Motivation: seamless integration of native YANG management data models in existing AUTOSAR process.

«goal»
other non-AUTOSAR data models ...

....

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 :

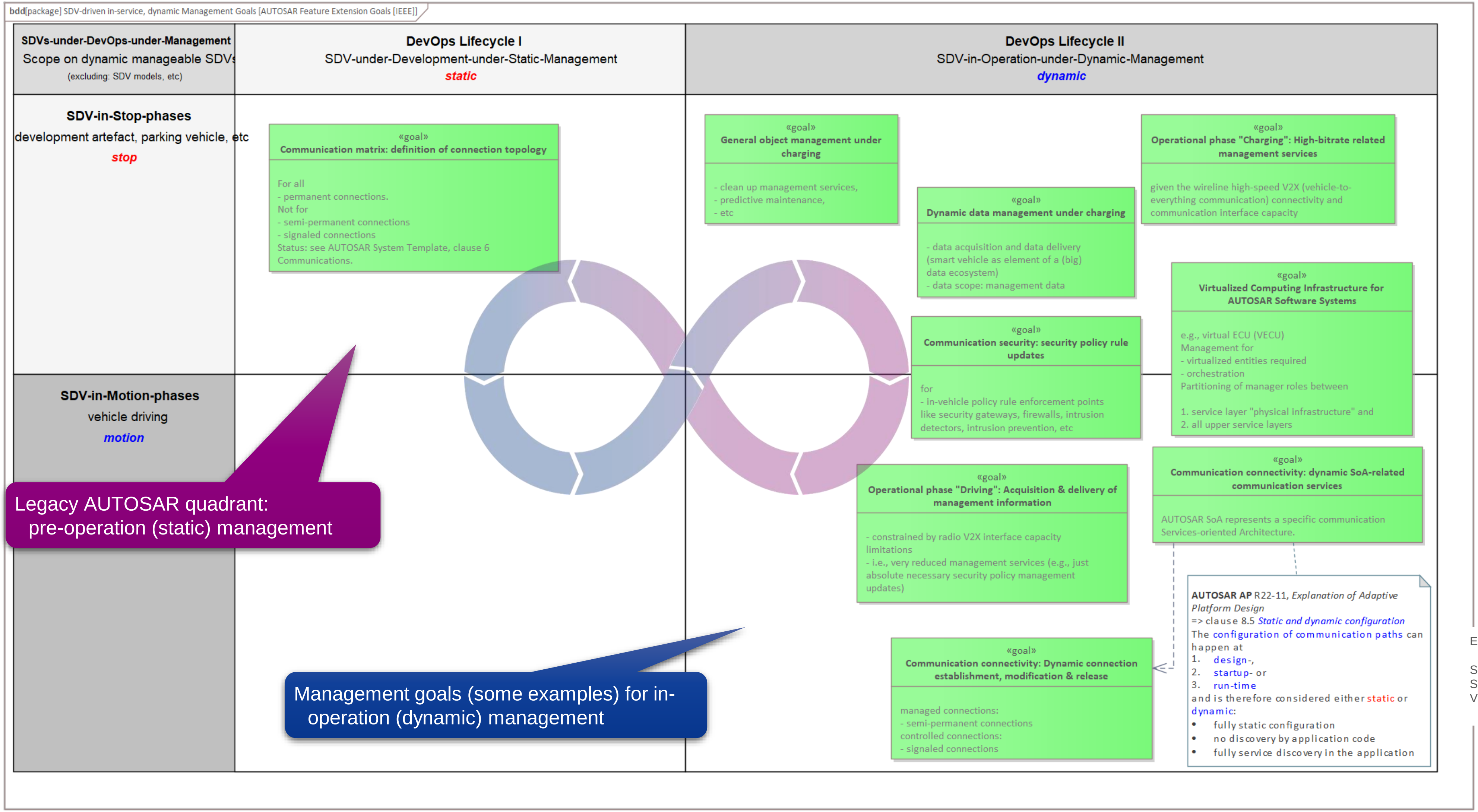
1. NETCONF is an established, mature, native protocol for professional management of distributed systems.
2. RESTCONF = NETCONF variant for web-based management.
3. CORECONF = NETCONF variant for constrained devices.

⇒ all automotive use cases could be fully covered!

AVB	(IEEE) Audio Video Broadcasting
CORECONF	CoAP Management Interface (NETCONF-oriented Constrained Application Protocol
COVESA	Connected Vehicle Systems Alliance
ICT	Information & Communication Technologies
NETCONF	a) Network Configuration Protocol b) IETF Working Group "Network Configuration"
RESTCONF	Representional State Transfer (REST) / HTTP variant of NETCONF
TSN	(IEEE) Time-Sensitive Networking
VSS	Vehicle Signal Specification (by COVESA)

Goals & non-goals

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



bdd[package] SDV-driven in-service, dynamic Management Goals [AUTOSAR Feature Extension Goals [IEEE]]

SDVs-under-DevOps-under-Management
Scope on dynamic manageable SDVs
(excluding: SDV models, etc)

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
stop

SDV-in-Motion-phases
vehicle driving
motion

«goal»
Communication matrix: definition of connection topology

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

«goal»
General object management under charging

- clean up management services,
- predictive maintenance,
- etc

«goal»
Dynamic data management under charging

- data acquisition and data delivery (smart vehicle as element of a (big) data ecosystem)
- data scope: management data

«goal»
Operational phase "Charging": High-bitrate related management services

given the wireline high-speed V2X (vehicle-to-everything communication) connectivity and communication interface capacity

«goal»
Communication security: security policy rule updates

for
- in-vehicle policy rule enforcement points like security gateways, firewalls, intrusion detectors, intrusion prevention, etc

«goal»
Virtualized Computing Infrastructure for AUTOSAR Software Systems

e.g., virtual ECU (VECU)
Management for
- virtualized entities required
- orchestration
Partitioning of manager roles between

1. service layer "physical infrastructure" and
2. all upper service layers

«goal»
Operational phase "Driving": Acquisition & delivery of management information

- constrained by radio V2X interface capacity limitations
- i.e., very reduced management services (e.g., just absolute necessary security policy management updates)

«goal»
Communication connectivity: dynamic SoA-related communication services

AUTOSAR SoA represents a specific communication Services-oriented Architecture.

«goal»
Communication connectivity: Dynamic connection establishment, modification & release

managed connections:
- semi-permanent connections
controlled connections:
- signaled connections

AUTOSAR AP R22-11, *Explanation of Adaptive Platform Design*
=> clause 8.5 *Static and dynamic configuration*
The configuration of communication paths can happen at
1. design-,
2. startup- or
3. run-time
and is therefore considered either static or dynamic:
- fully static configuration
- no discovery by application code
- fully service discovery in the application

Legacy AUTOSAR quadrant:
pre-operation (static) management

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

ECU	Electronic Control Unit = vehicle computer
SDV	Software-defined Vehicle
SoA	Service-oriented Architecture
V2X	Vehicle-to-Everything (communication)

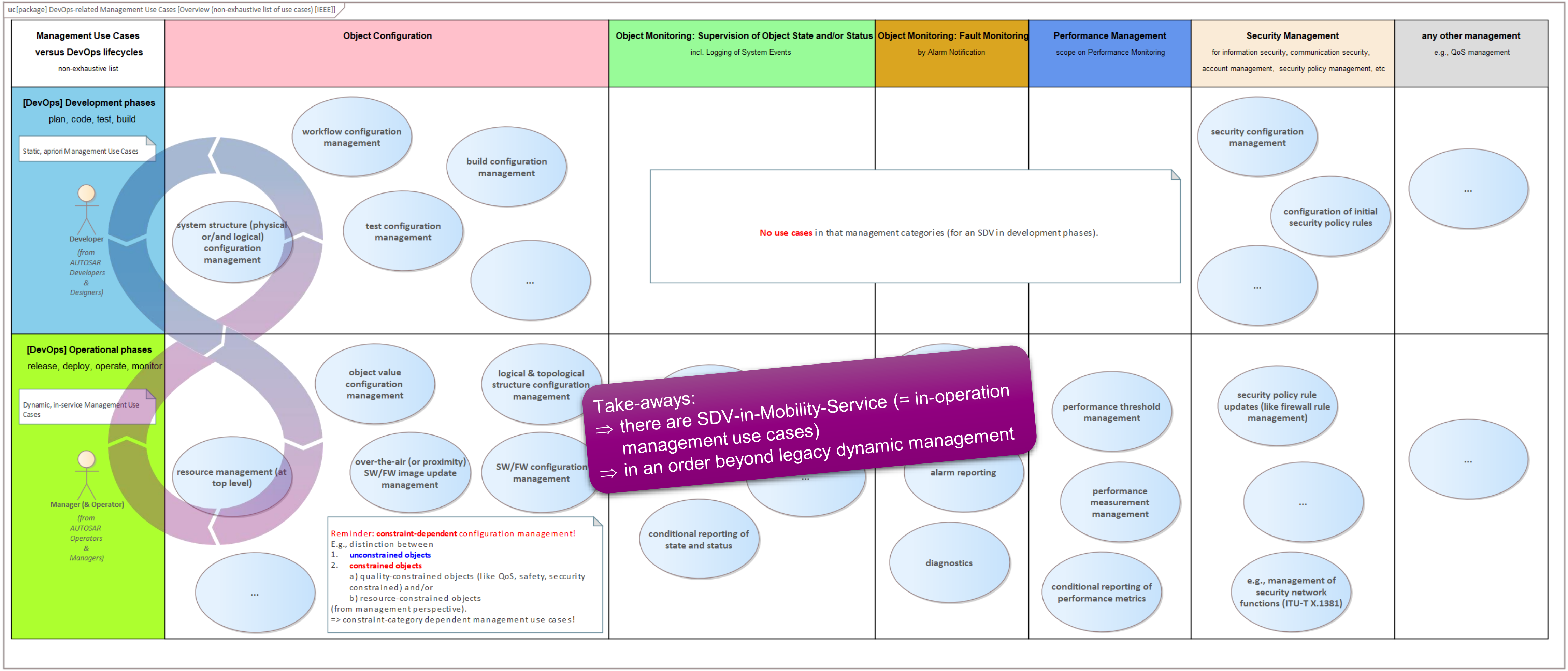
Operational Viewpoint (II)

for an SDV-under-DevOps capable AUTOSAR management architecture

AUTOSAR	AUTomotive Open System ARchitecture
DevOps	Lifecycle model "Development / Operations"
SDV	Software-defined Vehicle

DevOps-related management use cases

example use cases (non-exhaustive)



uc [package] DevOps-related Management Use Cases [Overview (non-exhaustive list of use cases) [IEEE]]

Management Use Cases versus DevOps lifecycles non-exhaustive list	**Object Configuration**	**Object Monitoring: Supervision of Object State and/or Status** incl. Logging of System Events	**Object Monitoring: Fault Monitoring** by Alarm Notification	**Performance Management** scope on Performance Monitoring	**Security Management** for information security, communication security, account management, security policy management, etc	**any other management** e.g., QoS management
[DevOps] Development phases plan, code, test, build Static, apriori Management Use Cases Developer (from AUTOSAR Developers & Designers)	workflow configuration management build configuration management system structure (physical or/and logical) configuration management test configuration management ...	**No use cases** in that management categories (for an SDV in development phases).			security configuration management configuration of initial security policy rules ...	...
[DevOps] Operational phases release, deploy, operate, monitor Dynamic, in-service Management Use Cases Manager (& Operator) (from AUTOSAR Operators & Managers)	object value configuration management logical & topological structure configuration management resource management (at top level) over-the-air (or proximity) SW/FW image update management SW/FW configuration management ...	... conditional reporting of state and status	alarm reporting diagnostics	performance threshold management performance measurement management conditional reporting of performance metrics	security policy rule updates (like firewall rule management) ... e.g., management of security network functions (ITU-T X.1381)	...

Reminder: **constraint-dependent** configuration management!
E.g., distinction between
1. **unconstrained objects**
2. **constrained objects**
   a) quality-constrained objects (like QoS, safety, seccurity constrained) and/or
   b) resource-constrained objects
(from management perspective).
=> constraint-category dependent management use cases!

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

QoS	Quality of Service

Functional viewpoint (III)

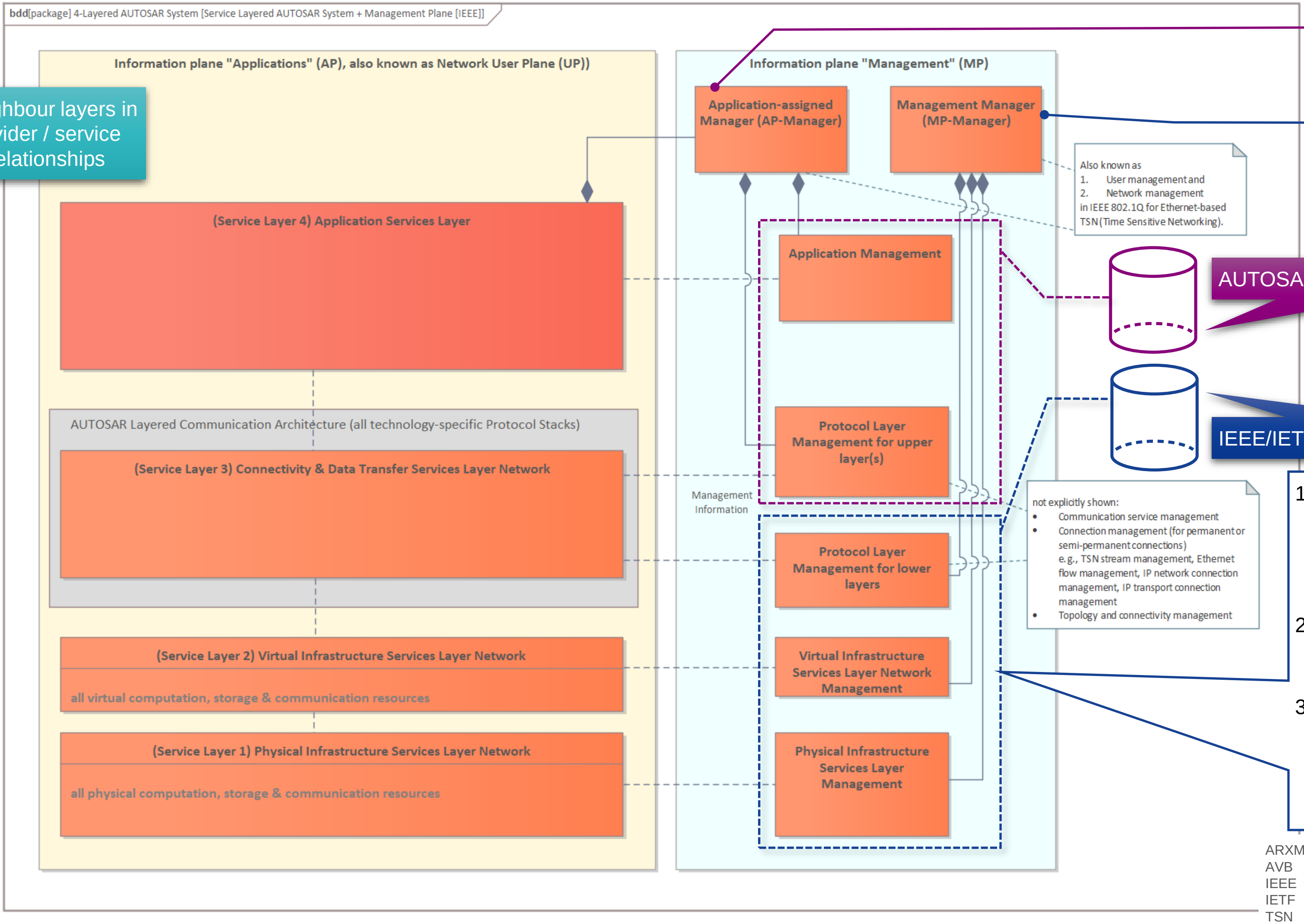
⇒ Functional service layering versus technical AUTOSAR software layering

⇒ Functional management architecture for vehicle local and/or remote management

Service layering

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

vertical neighbour layers in service provider / service consumer relationships


bdd[package] 4-Layered AUTOSAR System [Service Layered AUTOSAR System + Management Plane [IEEE]]
Information plane "Applications" (AP), also known as Network User Plane (UP))
(Service Layer 4) Application Services Layer
AUTOSAR Layered Communication Architecture (all technology-specific Protocol Stacks)
(Service Layer 3) Connectivity & Data Transfer Services Layer Network
(Service Layer 2) Virtual Infrastructure Services Layer Network
all virtual computation, storage & communication resources
(Service Layer 1) Physical Infrastructure Services Layer Network
all physical computation, storage & communication resources
Information plane "Management" (MP)
Application-assigned Manager (AP-Manager)
Management Manager (MP-Manager)
Application Management
Protocol Layer Management for upper layer(s)
Management Information
Protocol Layer Management for lower layers
Virtual Infrastructure Services Layer Network Management
Physical Infrastructure Services Layer Management

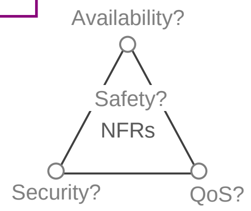
Also known as
1. User management and
2. Network management

in IEEE 802.1Q for Ethernet-based TSN (Time Sensitive Networking).

not explicitly shown:
- Communication service management
- Connection management (for permanent or semi-permanent connections)
  e.g., TSN stream management, Ethernet flow management, IP network connection management, IP transport connection management
- Topology and connectivity management

AP-Manager:
- distributed in legacy AUTOSAR
- more centralization in future?

MP-Manager:
- centralized!
- non-functional requirements (NFR)?



AUTOSAR-based ARXML data models

IEEE/IETF YANG-based ARXML data models

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.

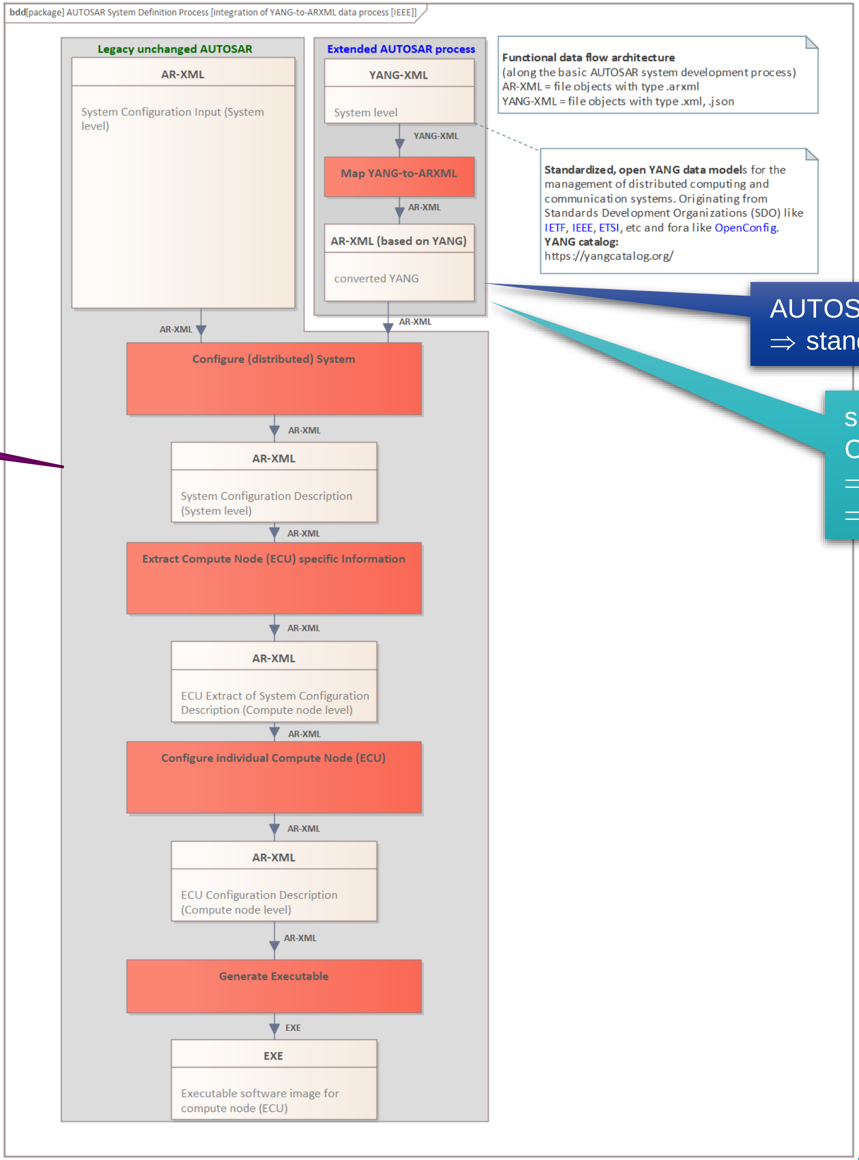
ARXML	AUTOSAR XML
AVB	(IEEE) Audio Video Broadcasting
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
TSN	(IEEE) Time-Sensitive Networking

AUTOSAR system development process

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!

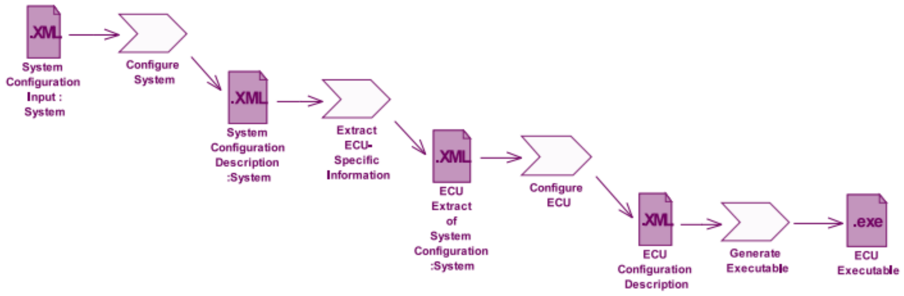

bdd[package] AUTOSAR System Definition Process [integration of YANG-to-ARXML data process [IEEE]]
Legacy unchanged AUTOSAR
AR-XML
System Configuration Input (System level)
Extended AUTOSAR process
YANG-XML
System level
YANG-XML
Map YANG-to-ARXML
AR-XML
AR-XML (based on YANG)
converted YANG
AR-XML
AR-XML
Configure (distributed) System
AR-XML
AR-XML
System Configuration Description (System level)
AR-XML
Extract Compute Node (ECU) specific Information
AR-XML
AR-XML
ECU Extract of System Configuration Description (Compute node level)
AR-XML
Configure individual Compute Node (ECU)
AR-XML
AR-XML
ECU Configuration Description (Compute node level)
AR-XML
Generate Executable
EXE
EXE
Executable software image for compute node (ECU)

Functional data flow architecture
(along the basic AUTOSAR system development process)
AR-XML = file objects with type .arxml
YANG-XML = file objects with type .xml, .json

Standardized, open YANG data models for the management of distributed computing and communication systems. Originating from Standards Development Organizations (SDO) like IETF, IEEE, ETSI, etc and fora like OpenConfig.
YANG catalog:
https://yangcatalog.org/

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


.XML
System Configuration Input : System
Configure System
.XML
System Configuration Description :System
Extract ECU-Specific Information
.XML
ECU Extract of System Configuration :System
Configure ECU
.XML
ECU Configuration Description
Generate Executable
.exe
ECU Executable

API	Application Programming Interface
ARXML	AUTOSAR XML
AUTOSAR	AUTomotive Open System ARchitecture
COVESA	Connected Vehicle Systems Alliance
ECU	Electronic Control Unit (= vehicle computer)
ETSI	European Telecommunications Standards Institute
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
VSS	Vehicle Signal Specification (by COVESA)
XML	Extensible Markup Language
YANG	Yet-Another-Next-Generation = name of a management data modeling language

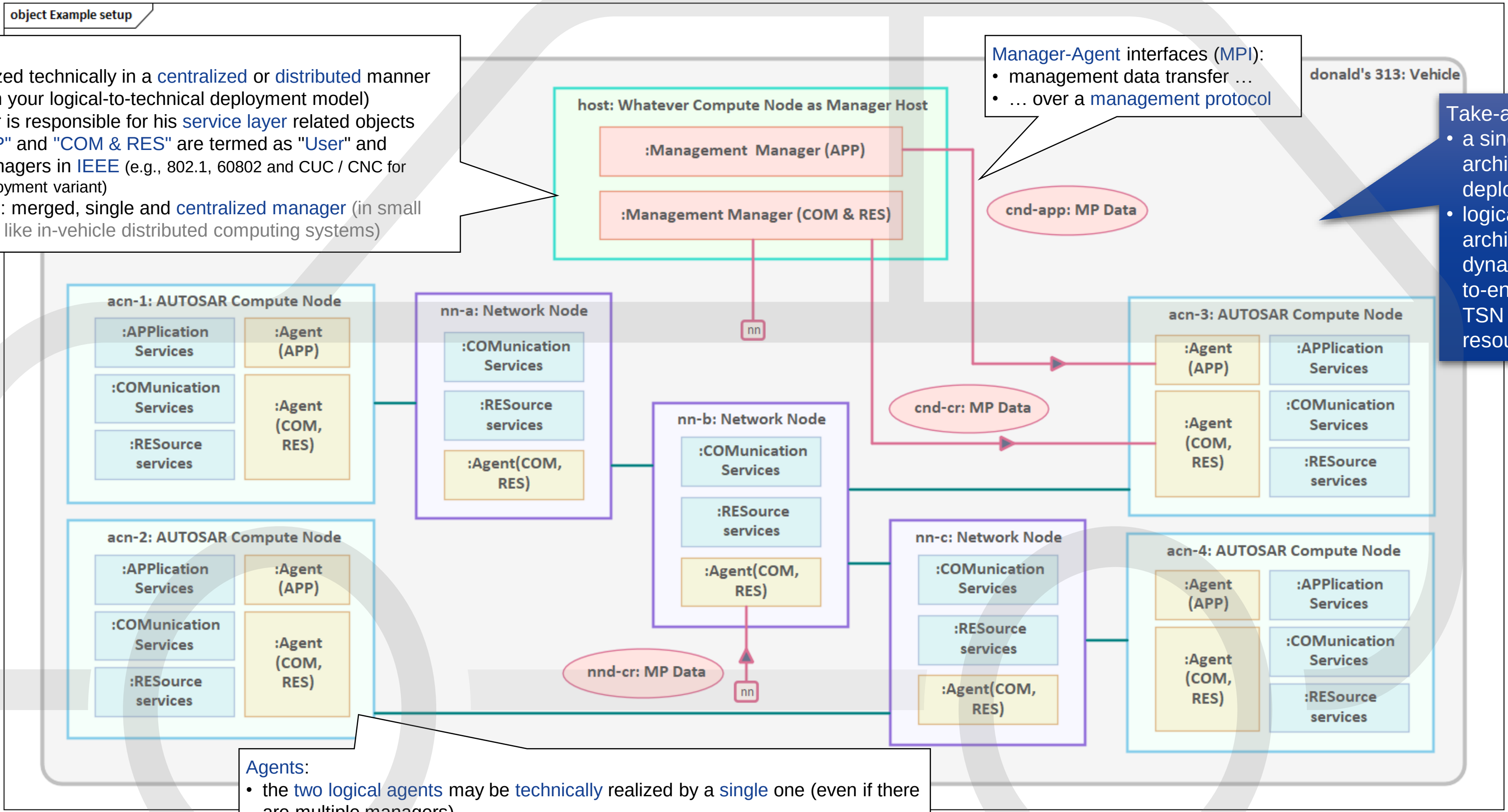
NOTE - The notion of AR-XML and YANG-XML intends to underline the common syntactical language baseline.

Logical viewpoint (IV)

Functional management architecture
for vehicle local and/or remote management

Logical AUTOSAR management architecture

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

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)

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)

Reminder!
Exemplification indicates only a few example, important model elements (in order not to overload the illustration).

CNC	(IEEE) Centralized Network Configuration (Manager)	= (centralized or distributed) Manager "COM & RES"
CUC	(IEEE) Centralized User Configuration (Manager)	= (centralized or distributed) Manager "APP"

Summary

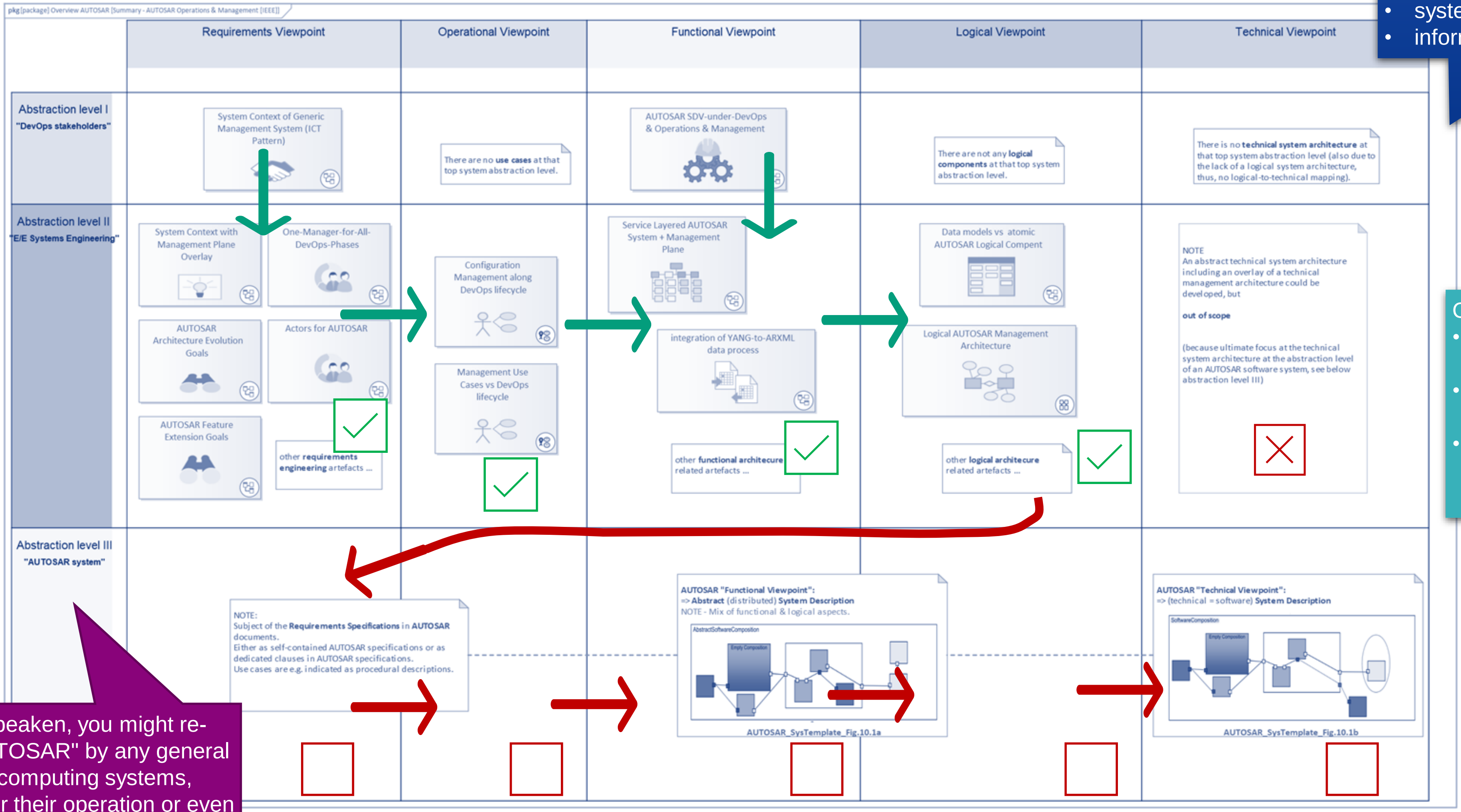
What’s next?

Summary

Model-based operation and management of SDVs = Model-based Systems Engineering!

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

Modeling languages:
- system & software: SysML, UML
- information & data: YANG, ARXML, …


pkg [package] Overview AUTOSAR [Summary - AUTOSAR Operations & Management [IEEE]]
Requirements Viewpoint
Operational Viewpoint
Functional Viewpoint
Logical Viewpoint
Technical Viewpoint
Abstraction level I
"DevOps stakeholders"
Abstraction level II
"E/E Systems Engineering"
Abstraction level III
"AUTOSAR system"
System Context of Generic Management System (ICT Pattern)
There are no use cases at that top system abstraction level.
AUTOSAR SDV-under-DevOps & Operations & Management
There are not any logical components at that top system abstraction level.
There is no technical system architecture at that top system abstraction level (also due to the lack of a logical system architecture, thus, no logical-to-technical mapping).
System Context with Management Plane Overlay
One-Manager-for-All-DevOps-Phases
AUTOSAR Architecture Evolution Goals
Actors for AUTOSAR
AUTOSAR Feature Extension Goals
other requirements engineering artefacts …
Configuration Management along DevOps lifecycle
Management Use Cases vs DevOps lifecycle
Service Layered AUTOSAR System + Management Plane
integration of YANG-to-ARXML data process
other functional architecure related artefacts …
Data models vs atomic AUTOSAR Logical Compent
Logical AUTOSAR Management Architecture
other logical architecure related artefacts …
NOTE
An abstract technical system architecture including an overlay of a technical management architecture could be developed, but
out of scope
(because ultimate focus at the technical system architecture at the abstraction level of an AUTOSAR software system, see below abstraction level III)
NOTE:
Subject of the Requirements Specifications in AUTOSAR documents.
Either as self-contained AUTOSAR specifications or as dedicated clauses in AUTOSAR specifications.
Use cases are e.g. indicated as procedural descriptions.
AUTOSAR "Functional Viewpoint":
=> Abstract (distributed) System Description
NOTE - Mix of functional & logical aspects.
AbstractSoftwareComposition
Empty Composition
AUTOSAR_SysTemplate_Fig.10.1a
AUTOSAR "Technical Viewpoint":
=> (technical = software) System Description
SoftwareComposition
Empty Composition
AUTOSAR_SysTemplate_Fig.10.1b

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 …

Frankly speaken, you might re-place "AUTOSAR" by any general in-vehicle computing systems, whether for their operation or even the development.

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)