SOVD – Service Oriented Vehicle Diagnostics

Bernd Wenzel
ASAM e.V.

Tobias Weidmann
Vector Informatik GmbH

Dresden
17.05.2022

VECTOR

ASAM Association for Standardization of Automation and Measuring Systems

Agenda

1 Motivation

2 Concepts

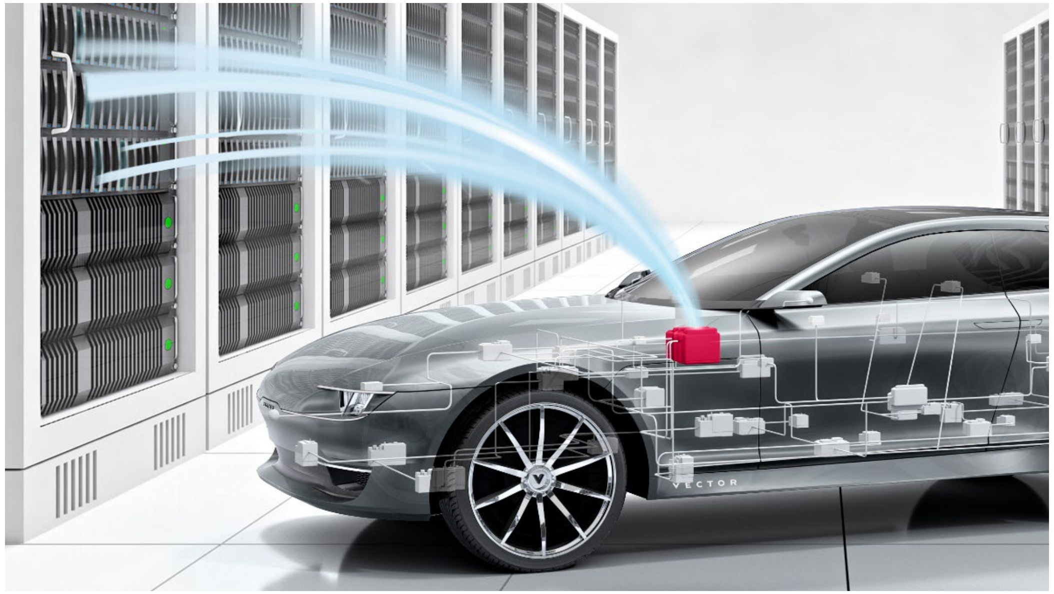
3 Methods Overview

4 Standardization activities based on SOVD

Motivation

Challenge

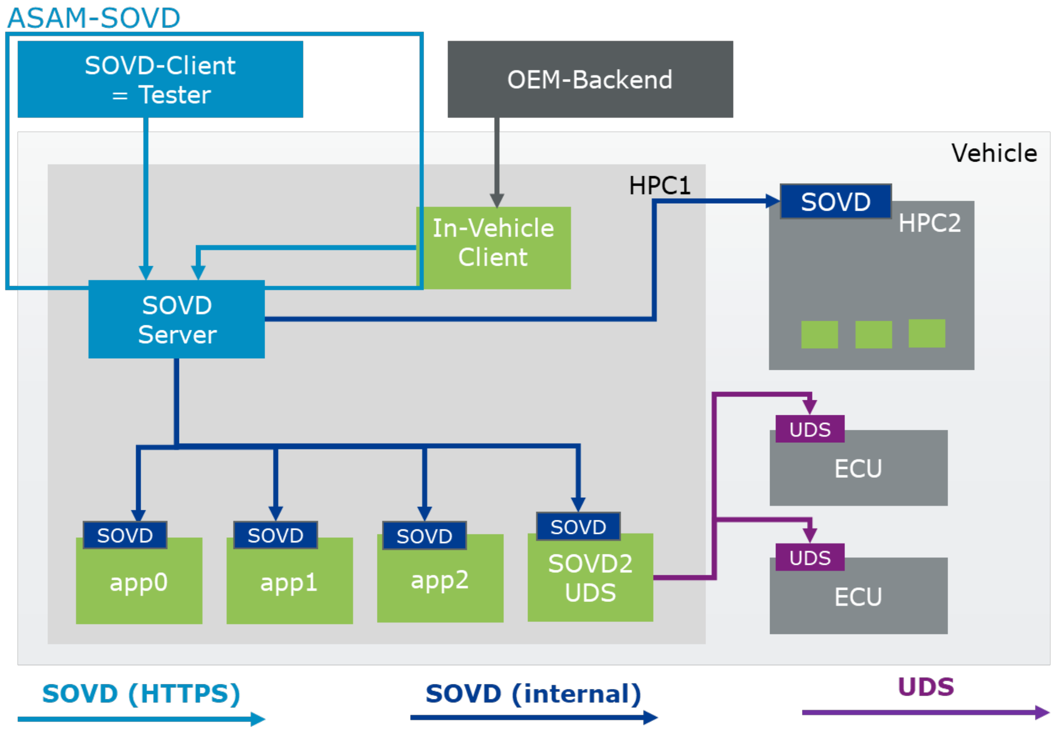
- New architectures based on HPCs, multiple OS, the different applications and their dependencies put a major challenge also to diagnostics.
- Focus extends from identifying hardware errors to analyzing software issues.
- SW-analysis requires different type of data
  - Logs, traces, process information, stack traces
- Diagnostic content in the vehicles will change dynamically, this contrasts with the static approach of UDS.
- SW-update will change from transferring individual bits and bytes to controlling a complex update procedure in the vehicle.



Motivation

Overview

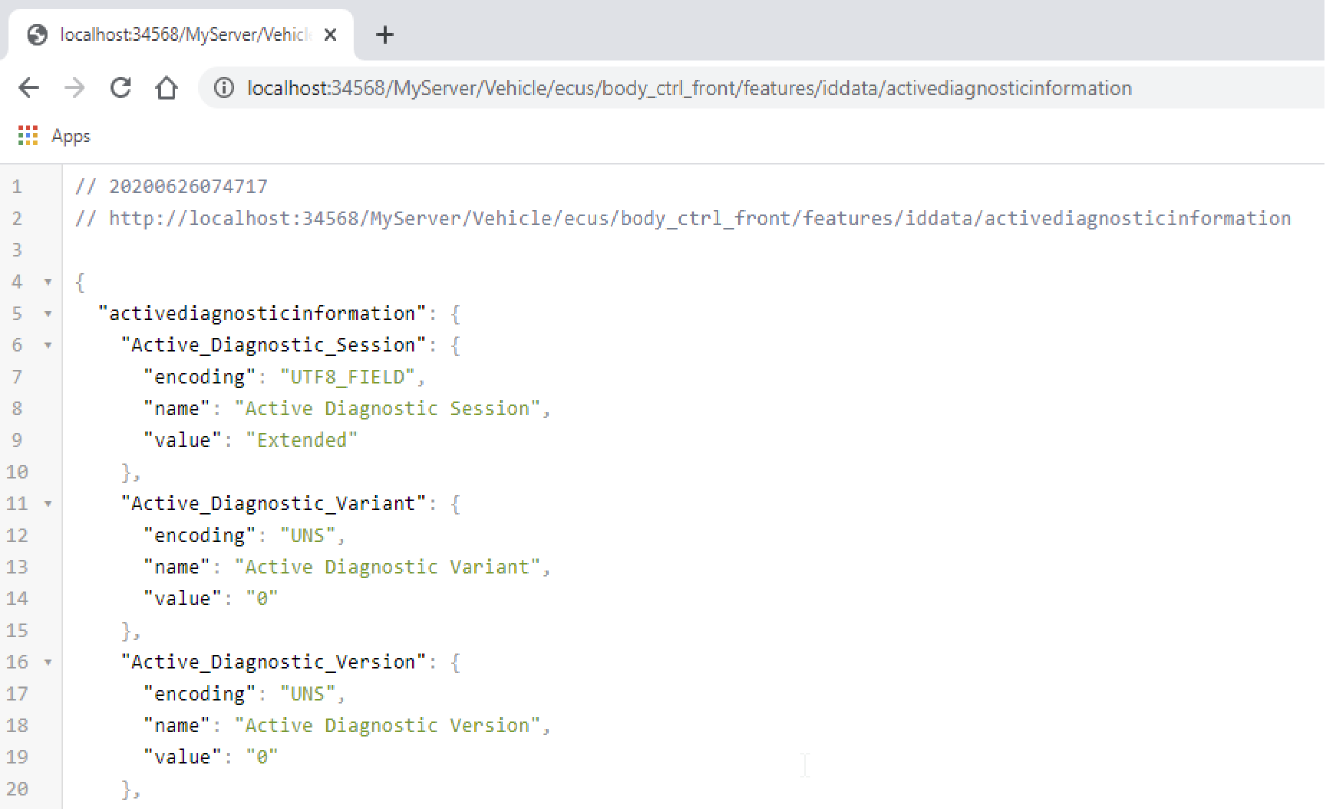
- One API for diagnostics and software update (cross-vehicle)
- SOVD covers traditional use-cases
  - Data access, fault information, control of internal SW-functions
- SOVD covers HPC related diagnostic use-cases
  - vehicle SW-update, logging, tracing, access to system information, dynamic discovery of content
- Based on IT-technology (HTTP/REST, JSON, OAuth)
- SOVD encapsulates UDS but does not replace it
- Focus of ASAM SOVD is the API, discussions for an implementation started with AUTOSAR



Concepts

HTTP/REST in a nutshell

- REST is based on HTTP, basically a web browser is sufficient to execute
- Resources are the core element
- HTTP verbs on the resources represent the different operations
- Knowledge of the initial URL is sufficient, further links are provided to discover the API
- REST is stateless, i.e.
  every request contains all the relevant information that the server can process it


localhost:34568/MyServer/Vehicl
localhost:34568/MyServer/Vehicle/ecus/body_ctrl_front/features/iddata/activediagnosticinformation
Apps

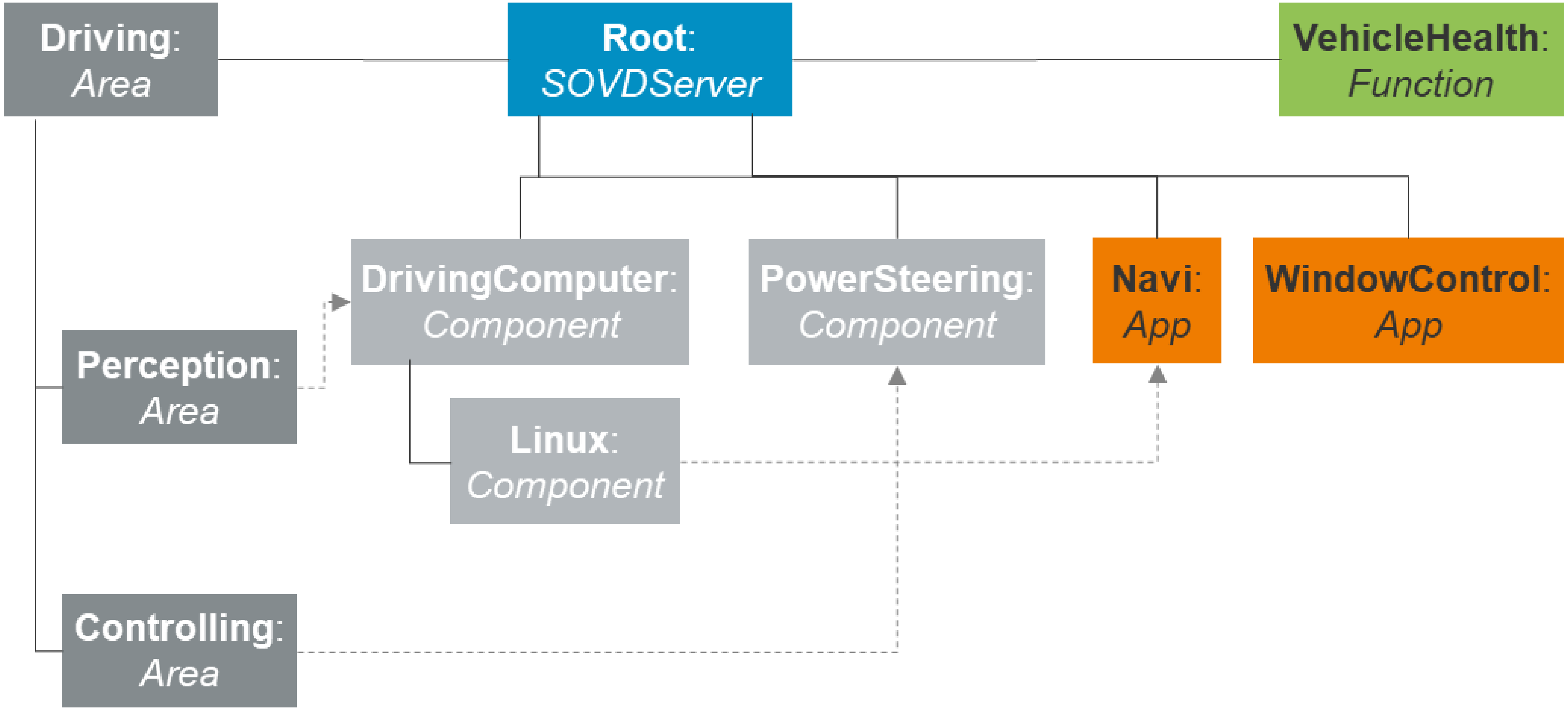
```
// 20200626074717
// http://localhost:34568/MyServer/Vehicle/ecus/body_ctrl_front/features/iddata/activediagnosticinformation

{
  "activediagnosticinformation": {
    "Active_Diagnostic_Session": {
      "encoding": "UTF8_FIELD",
      "name": "Active Diagnostic Session",
      "value": "Extended"
    },
    "Active_Diagnostic_Variant": {
      "encoding": "UNS",
      "name": "Active Diagnostic Variant",
      "value": "0"
    },
    "Active_Diagnostic_Version": {
      "encoding": "UNS",
      "name": "Active Diagnostic Version",
      "value": "0"
    },
```

No automotive specific stack needed on client side

Agenda

1	Motivation
2	Concepts
3	**Methods Overview**
4	Standardization activities based on SOVD

SOVD Structure



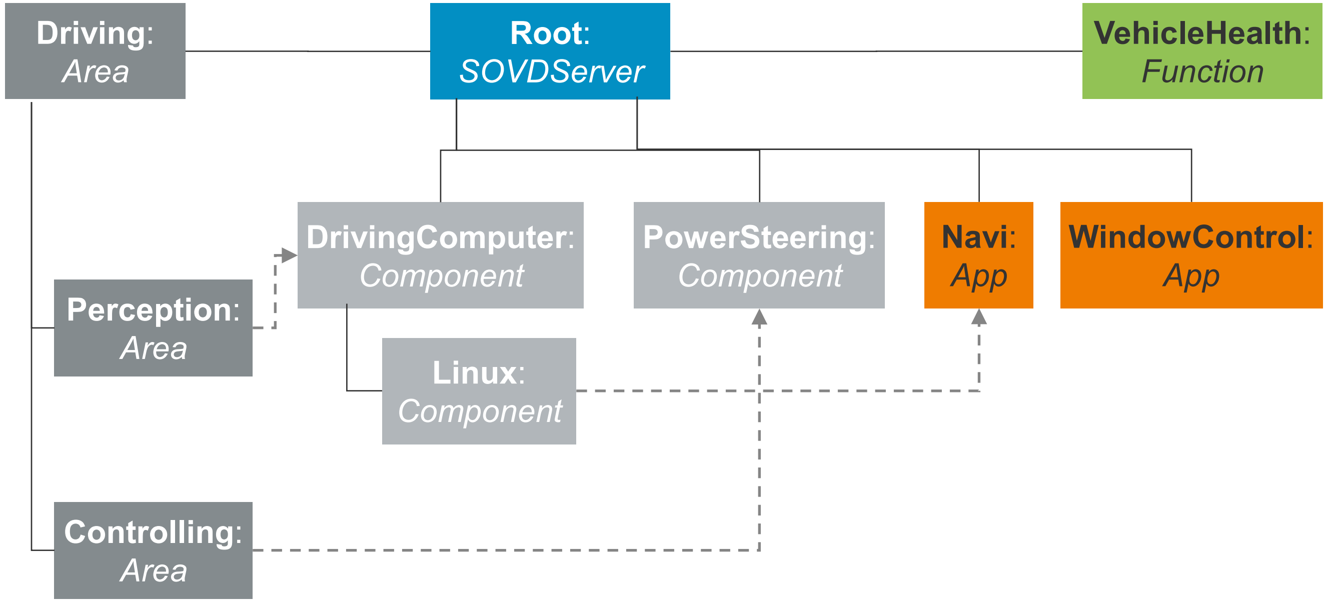
Method for capability discovery

Discovering of entities and resources

- Discovery of contained entities
- Query sub-entities of an entity
- Query related entities of an entity
- Query entity capabilities
- Areas represent a topological view on the entities

Access to capability description content

- Query an online capability description
- Query schema information for content processing



Identical format for offline and online capability description used, based on OpenAPI an JSON Schema

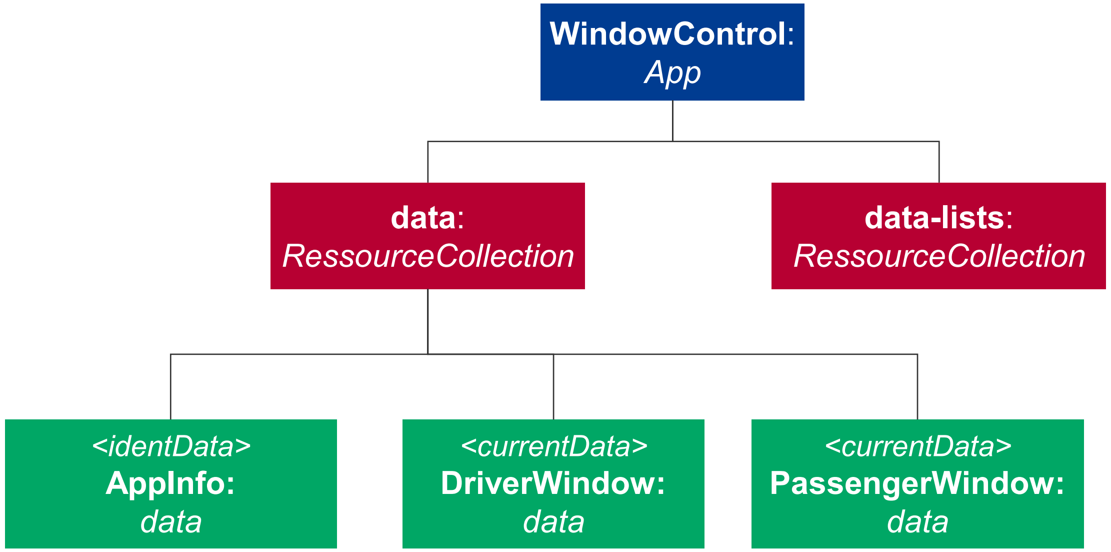
Method for data resource read / write access

Methods

- Retrieve the list of data available for an entity
- Data is categorized according to its semantic
  - E.g. currentData, identData, storedData, sysInfo
- Read/write access to data
- Possibilities to group data

- Possibilities to create aggregated data sets on entity level

- Periodic / Trigger Based data access is planned for v1.1

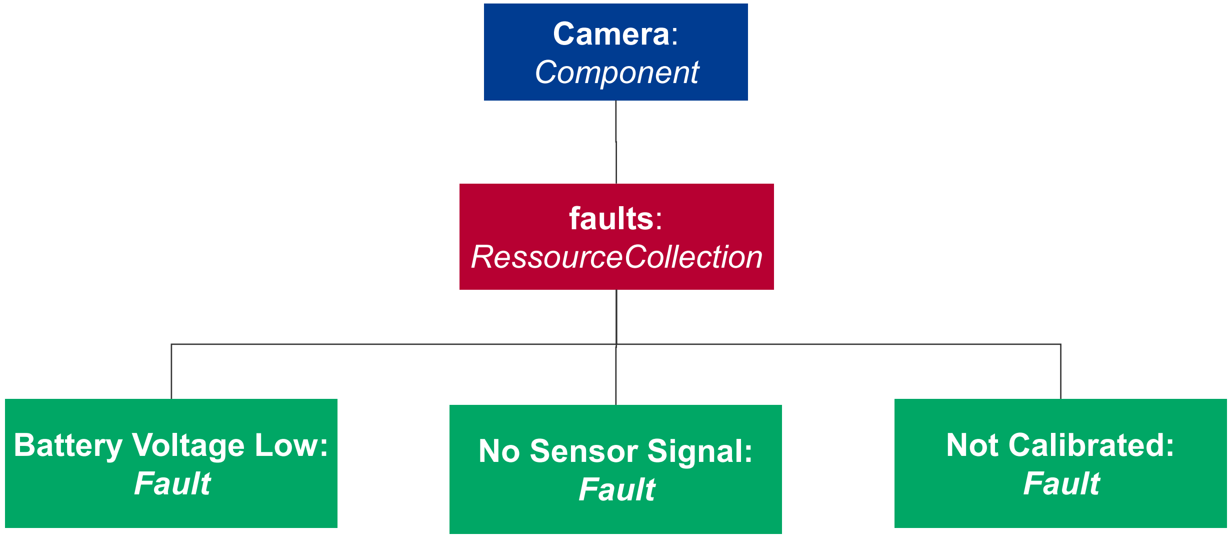


WindowControl: *App*
data: *RessourceCollection*
data-lists: *RessourceCollection*
<identData> **AppInfo:** *data*
<currentData> **DriverWindow:** *data*
<currentData> **PassengerWindow:** *data*

Method for fault handling

Methods

- Read faults from an entity
- Read details for a fault
- Delete all faults of an entity
- Delete single fault of an entity


Camera: *Component*
faults: *RessourceCollection*
Battery Voltage Low: ***Fault***
No Sensor Signal: ***Fault***
Not Calibrated: ***Fault***

Query parameters

- Status, based on key value pair
- Severity

Access to environment data for a single fault code

- OEM specific key value pairs

Method for control of operations

Operations (SW-internal functions, actuators)

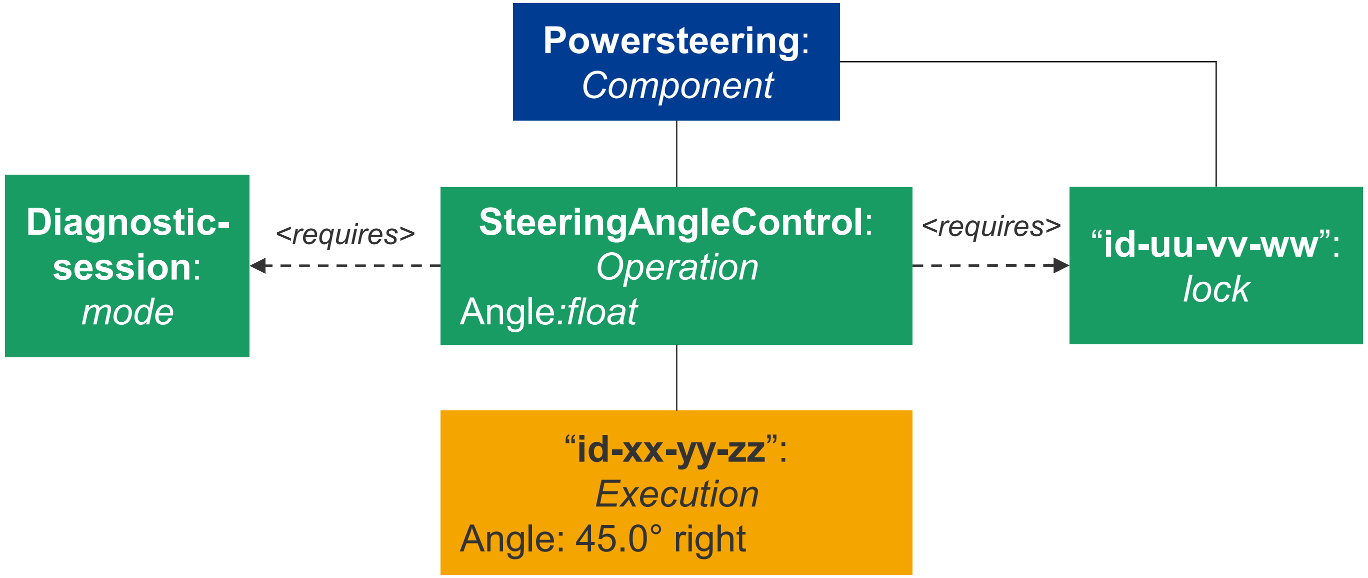
- List available operations
- Initiate the execution (potentially multiple)
- Monitor the status, adjust the execution
- Terminate the execution

Target modes

- Retrieve list of all supported modes of an entity
- Explicit control of entity states via their defined modes

Locking

- Goal: avoid parallel usage of entities in certain sequences
- Acquire a lock on an entity
- Release an acquired lock on an entity



Powersteering: *Component*
Diagnostic-session: *mode*
<requires>
SteeringAngleControl: *Operation*
Angle:*float*
<requires>
"id-uu-vv-ww": *lock*
"id-xx-yy-zz": *Execution*
Angle: 45.0° right

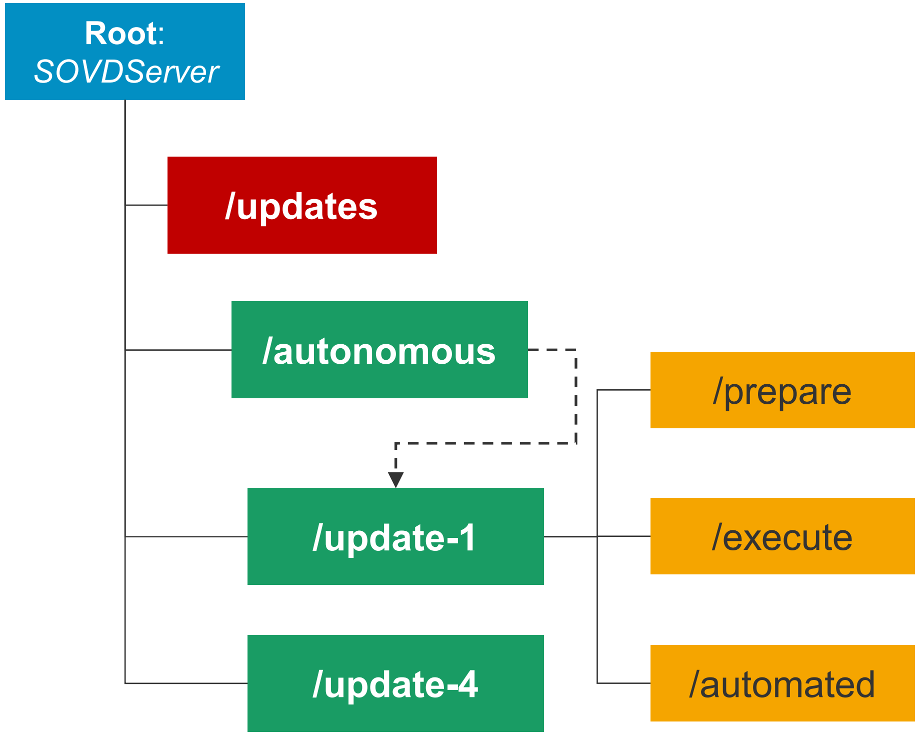
Method for software update

Basics

- It is assumed that there is a central component in the vehicle which performs the software update
- ASAM SOVD provides an API to trigger this central software update component
- Update procedure itself is not subject to ASAM SOVD

Methods

- Retrieve list of all updates provided by the entity
- Get details of update
- Automated installation of an update
- Prepare installation of an update
- Execute installation of an update
- Get status of an update
- Delete update package from a SOVD server
- Register an update at the SOVD server



Method for logging

Basics

- Access to aggregated log information
- Evaluation by software experts
- Transport as bulk-data possible

Methods

- Retrieve list of all log information
- Configure SOVD logging
- Retrieve the current SOVD logging configuration
- Reset SOVD Logging configuration to default

Supported Context Types

- RFC 5424 (syslog protocol)
- AUTOSAR diagnostic log and trace

Request:

```
HTTP GET
{base_uri}/components/DrivingComputer/logs/entries
```

Response:

```
HTTP 200 OK
{
  "items": [
    {
      "timestamp": "2021-07-20T00:00:04.387819Z",
      "context":
      {
        "type": "RFC5424",
        "host": "Linux",
        "process": "systemd",
        "pid": 1
      },
      "severity": "info",
      "msg": "Closed D-Bus User Message Bus Socket",
    },
    {
      …
    }
```

Agenda

1 Motivation

2 Concepts

3 Methods Overview

4 Standardization activities based on SOVD

Standardization activities based on SOVD

Overview

ASAM SOVD 1.0

- Released on TSC meeting in July

AUTOSAR Alignment

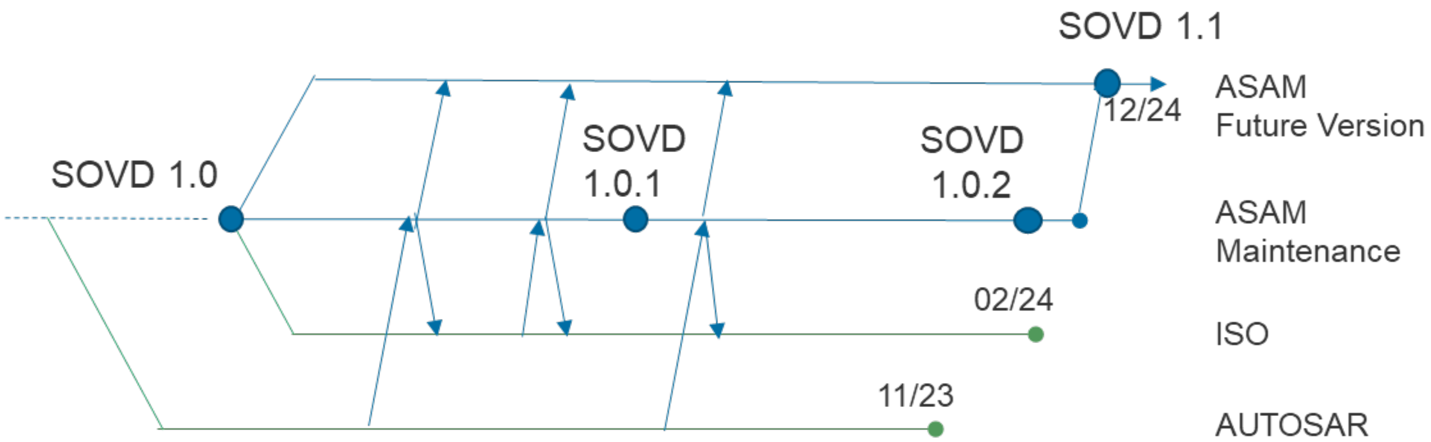
- Involved in internal review
- Handled in concept group 704
- ara::diag extension

ISO Standardization

- Joint NWIP in preparation with other SC31 groups

ASAM Follow Up project

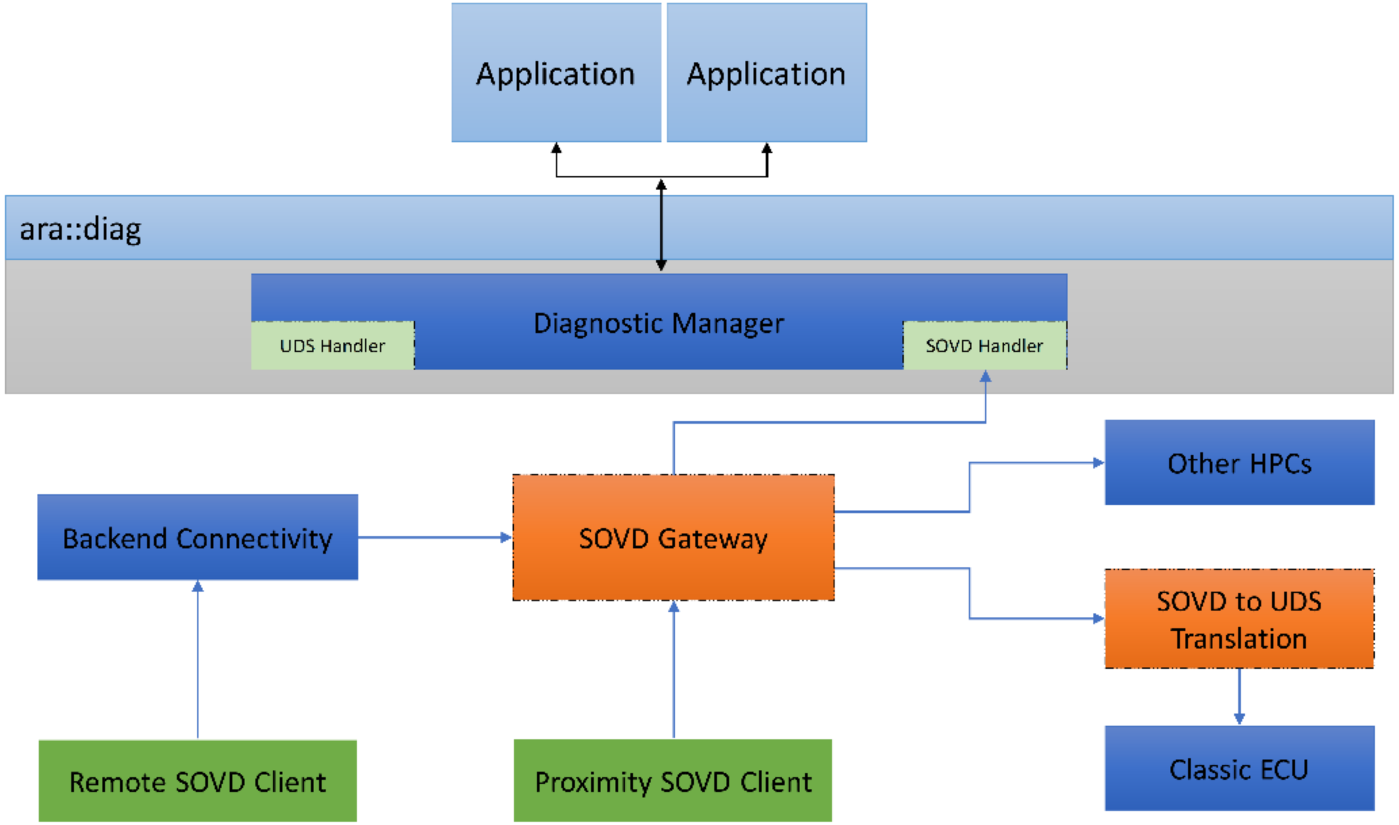
- Started 09/2022, release planned for 12/2024
- Compatible minor version
- Integration of ISO feedback
- Event based communication



VECTOR ASAM

Standardization activities based on SOVD

AUTOSAR concept



source: AUTOSAR SOVD concept group

Standardization activities based on SOVD

SOVD v1.1

Enhancements to Data Access

- Cyclic reception of sensor data or variables values
  - SOVD will support Cyclic access to data resources
  - Reduce communication overhead for data access at a high frequency

- Trigger based data access
  - Especially for sensor data, DTCs and environment data, HPC system information

SOVD for legally required diagnostics

- The project group will discuss and analyze how SOVD could be used to fulfill legally required diagnostics

Standardization activities based on SOVD

SOVD v1.1

Enhanced Diagnostic Use Cases

- In-vehicle data communication logging
  - SOVD will provide a possibility to control an in-vehicle data logger. The use-case is to provide snapshots of the in-vehicle communication for complex failure scenarios

- Execution of diagnostic command sequences or customs scripts
  - SOVD will provide the possibility to control the execution of an in-vehicle diagnostic script.
  - The scripting language will not be standardized.

Enhancements to existing Services

- Extend Logging API to support also continuous transport of log-information
- File Handling via Third Party Providers
- Enhancements to Multi client access to the vehicle
- Reset to Default Setting and/or Restart

Thank you for your attention!

Bernd Wenzel
Senior Technical Consultant
ASAM e.V.

Phone: +49 371 5607 742
Email: bernd.wenzel@asam.net

Tobias Weidmann
Manager Customer Services
Vector Informatik GmbH

Phone: +49 711 80670 2549
Email: tobias.weidmann@vector.com