# ASAM SOVD

## Diagnostic API for software-defined vehicles

**Bernd Wenzel**
**ASAM e.V.**

Oct, 23, 2025
Novi

Association for Standardization of Automation and Measuring Systems

# Agenda

# Agenda

| | |
|---|---|
| 1 | ASAM SOVD |
| 2 | ASAM SOVD – Development Timeline |
| 3 | Proposed content V 1.2.0 |
| 4 | ASAM OTX Extension for ASAM SOVD |

# ASAM SOVD

Data access to SDVs:

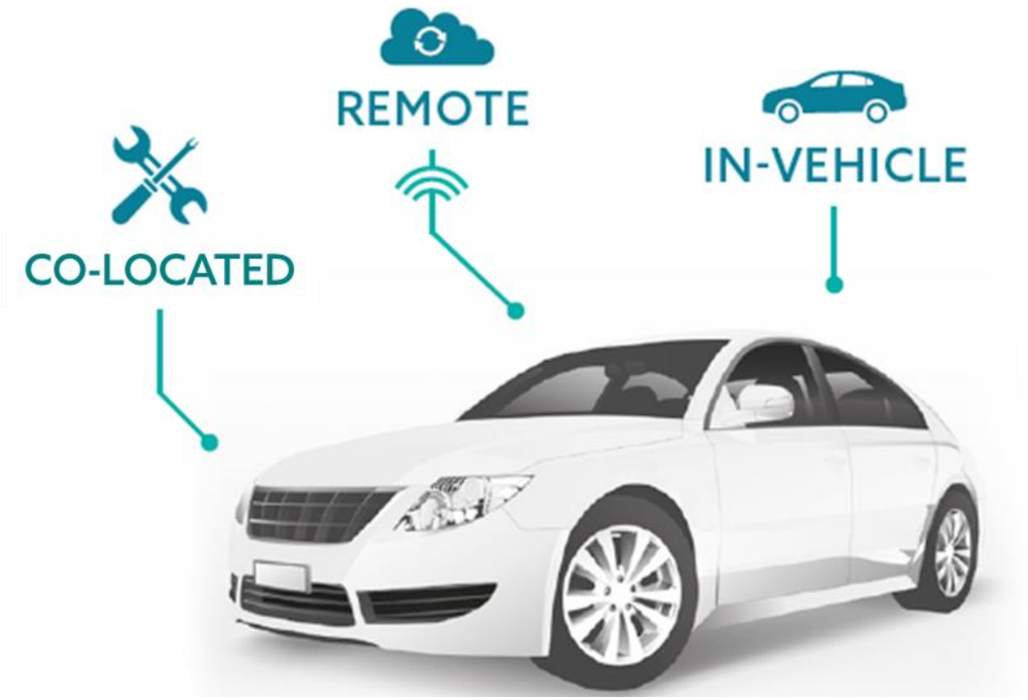
## Designed for diagnostics

## – Built for more!

# ASAM SOVD

## Why ASAM SOVD

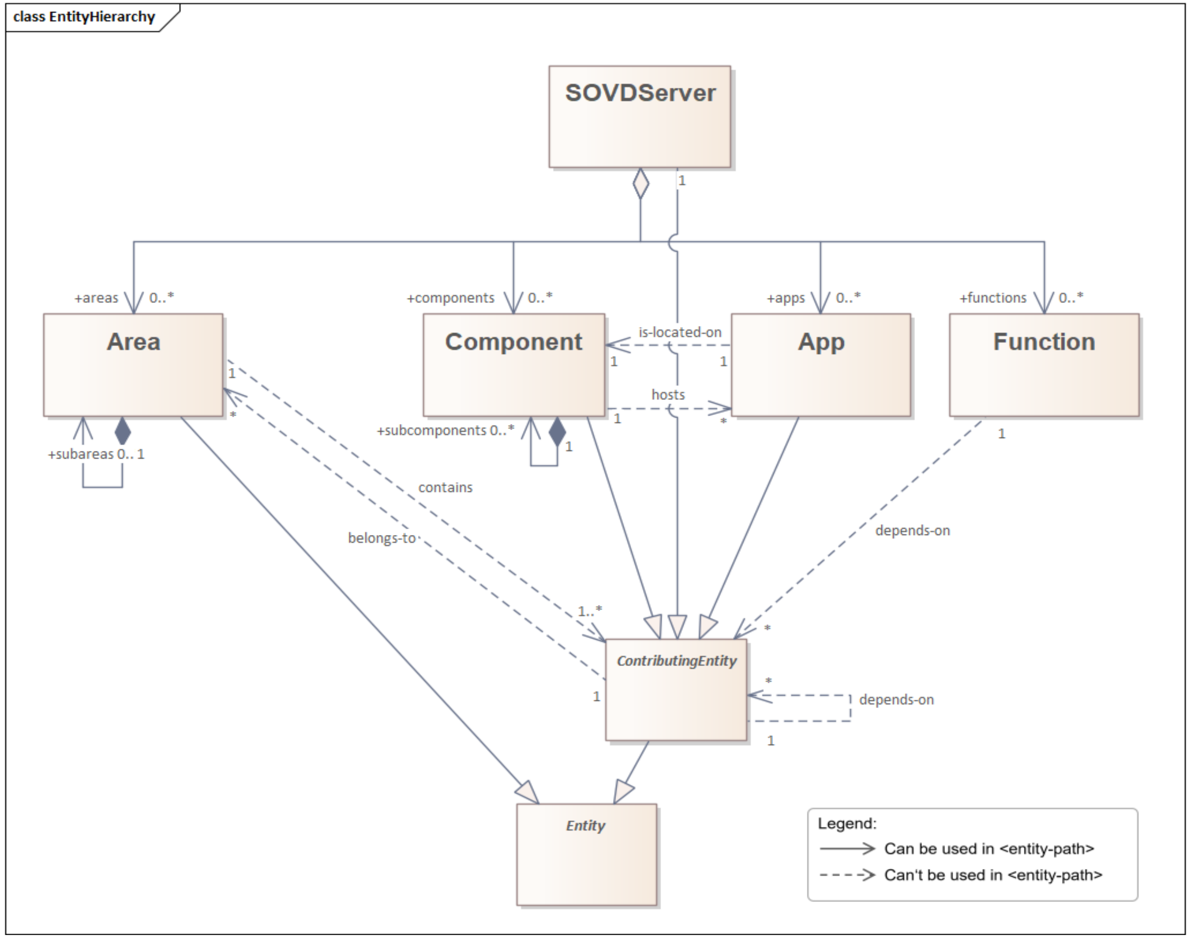
- Support for next generation software architectures
- Uniform diagnostics of High-Performance Computers (HPCs)
- Continuous update of software in the vehicle, which may also include new functionalities
- Vehicle as IOT device
- Software analysis during operation (not just simple reading of error codes)
- Support of interactive generic diagnostics



# ASAM SOVD

## Architectural concept

- HTTP REST-based API for diagnosing SDVs
- Data is provided as resources in entities
- Support of both simple and complex data types
- Data description in JSON format
- Data is presented on a physical level
- Data uses SI units
- Data is decoupled from meta data



# ASAM SOVD

Feature overview version 1.1.0

## Basic services

- Capability discovery
- Capability description
- Access to legal diagnostics

## Classic features

- Fault handling (incl. environment data)
- Reading and writing data
- Configuration
- Control of operations

## HPC-related features

- Software update
- Execution of scripts
- Handling of bulk-data
- Enhanced Locking
- Access to logging information
- In-Vehicle Comm Logging
- Start, Stop, Restart of Entities

## Basic services

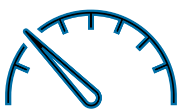
- Security
- Authentication
- Authorization
- Trigger-based access
- Cyclic data access

# ASAM SOVD

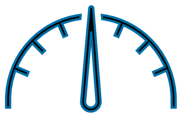
## Cyclic data access

- Server Sent Events (SSE) via Event Source API used to support publish / subscribe pattern
- Query for cyclic-subscriptions from an entity
- Issue a new cyclic-subscription

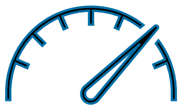
- Supported Intervals



Slow:
250 .. 500 ms



Normal:
100 .. 250 ms



Fast:
less than 100 ms

- Duration-defined
- Retrieve details of a cyclic-subscription (how client can access, schema of event stream, observed resources)
- Update a cyclic-subscription
- Delete a cyclic-subscription

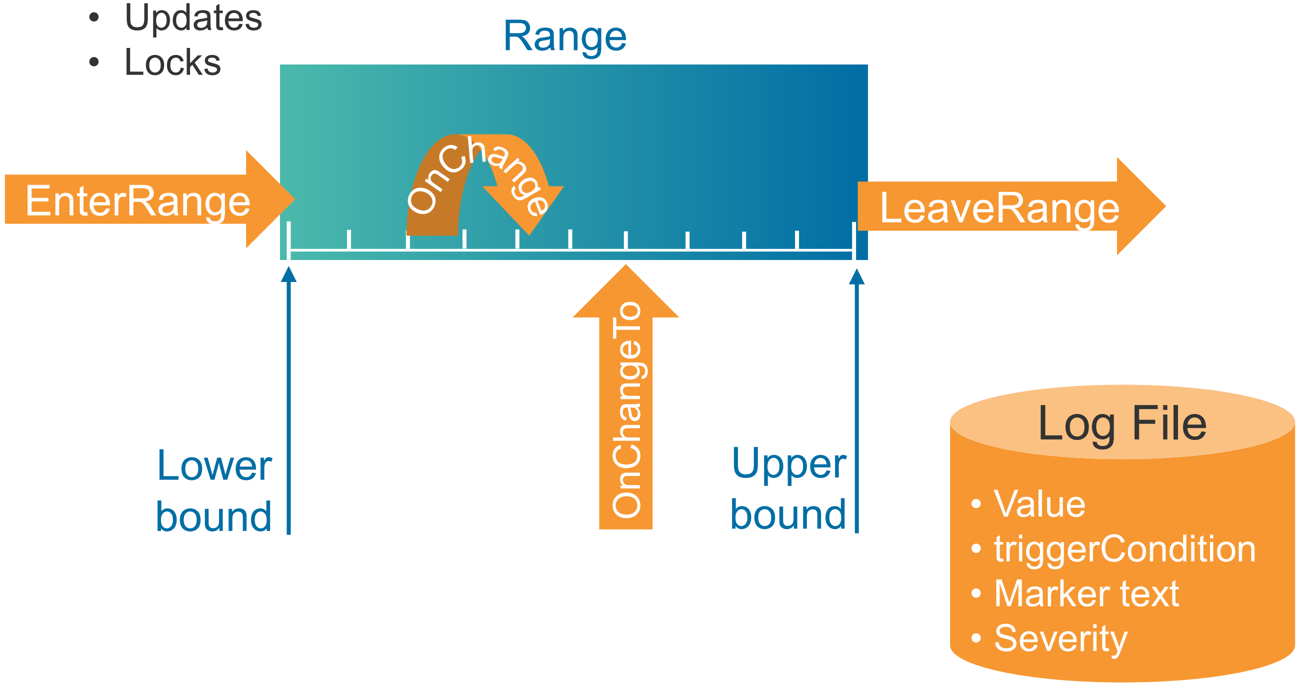
# ASAM SOVD

## Trigger-based access

- Server Sent Events (SSE) used, to notify the client when a trigger fires
- Single-shot and multi-shot triggers
- Persistent triggers
- Lifetime definition
- Issue a new trigger
- Query for triggers from an entity
- Retrieve details of a trigger
- Update a trigger
- Delete a trigger

**Observable objects:**

- Data
- Faults
- Execution of operations
- Execution of scripts
- Updates
- Locks



# ASAM SOVD

## Control of operations

| Execution | Description |
|---|---|
| synchronous | returns with result after the operation has been completed |
| asynchronous | returns immediately and a client has to query the status of the execution |

| I/O Control Capability | Description |
|---|---|
| execute | Short term adjustment |
| freeze | Freeze current state |
| reset | Reset to default |

**Proximity Proof**

- Challenge – Response

**Target Modes**

- Explicit
- Implicit

**Locking**

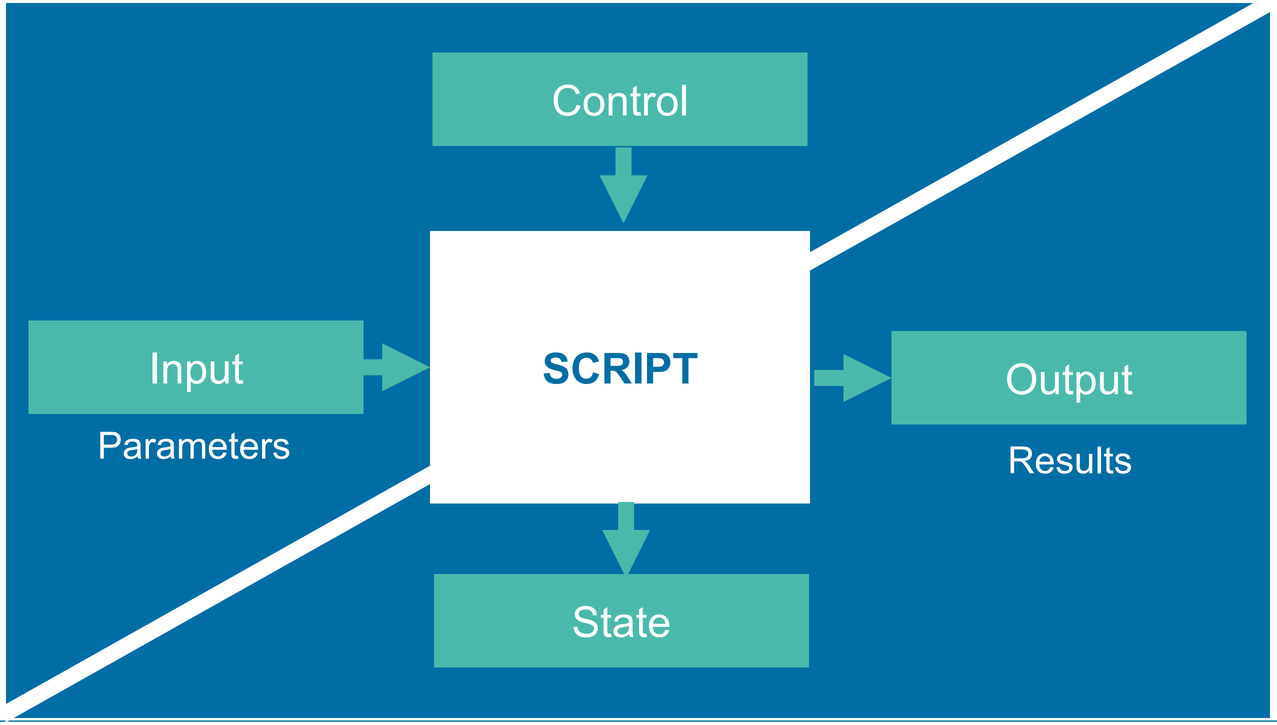
- Controlled with expiration time

# ASAM SOVD

Execution of scripts

**General functions**

- List available scripts
- Upload a script
- Retrieve details of a script
- Remove a script




**Control**

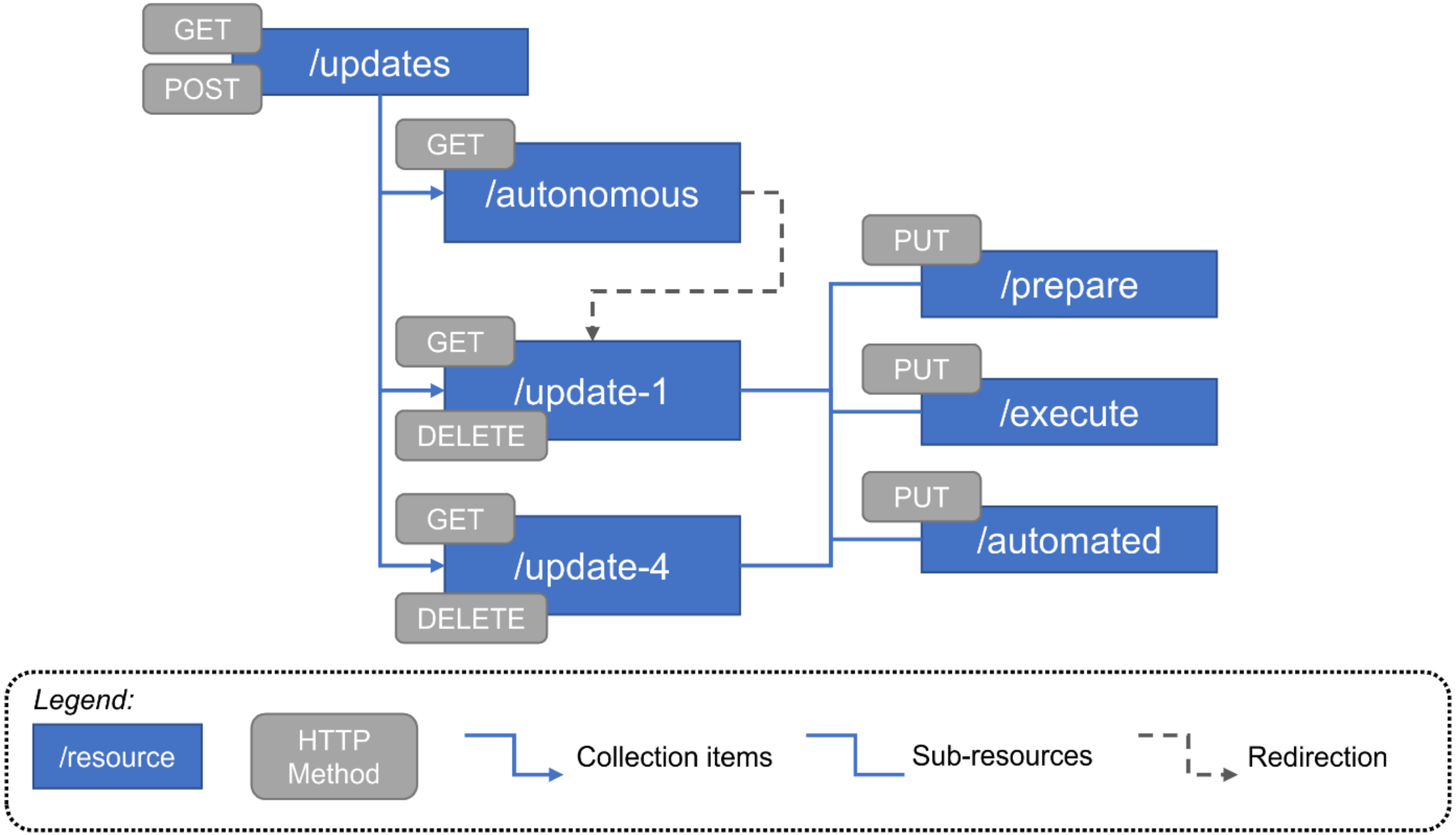
- Start
  + now
  + now and on restart
  + on restart
  + once on restart
- Terminate (stop, forcedTermination)

**State**

- prepared
- running
- completed
- failed
- terminated

# ASAM SOVD

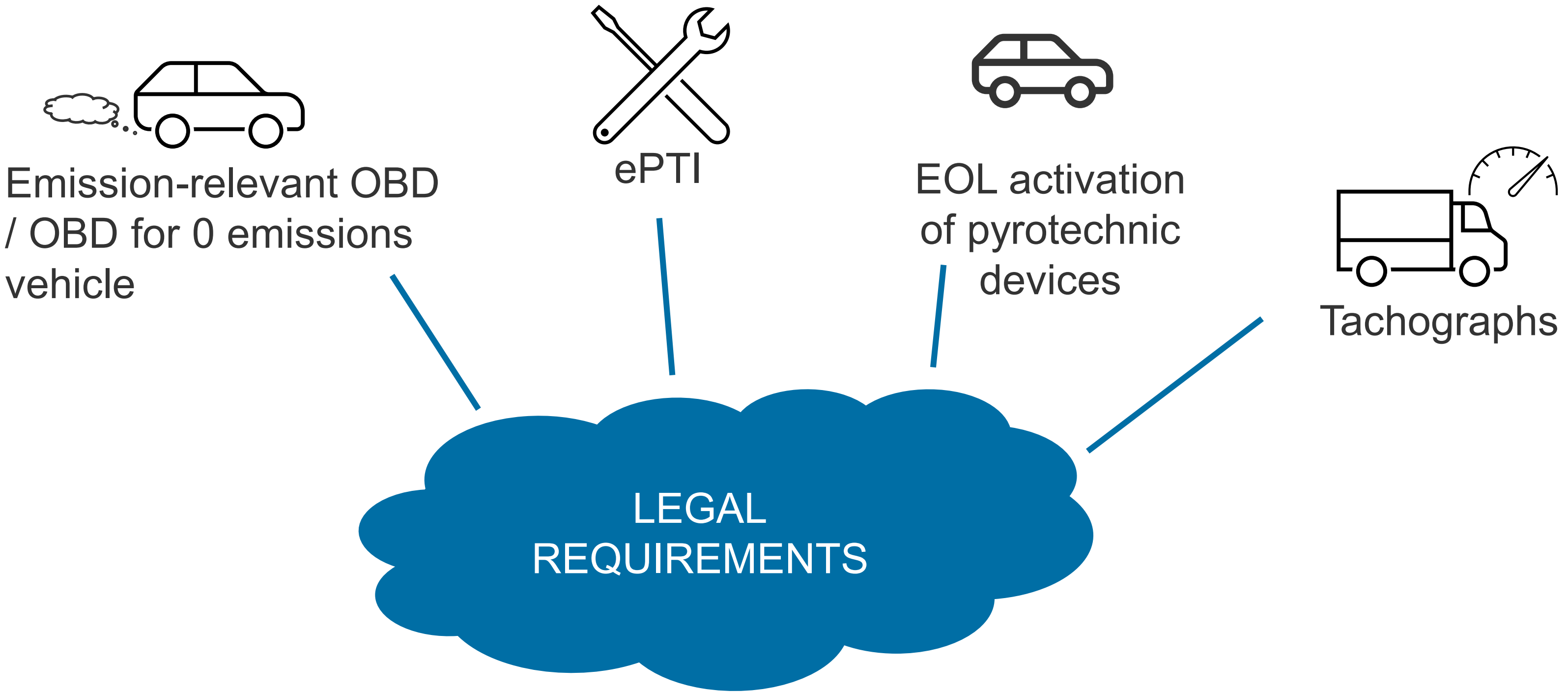
Software update - example



# ASAM SOVD

Access to legal diagnostics -tagging of entities and resources

## Examples




## Standard solution

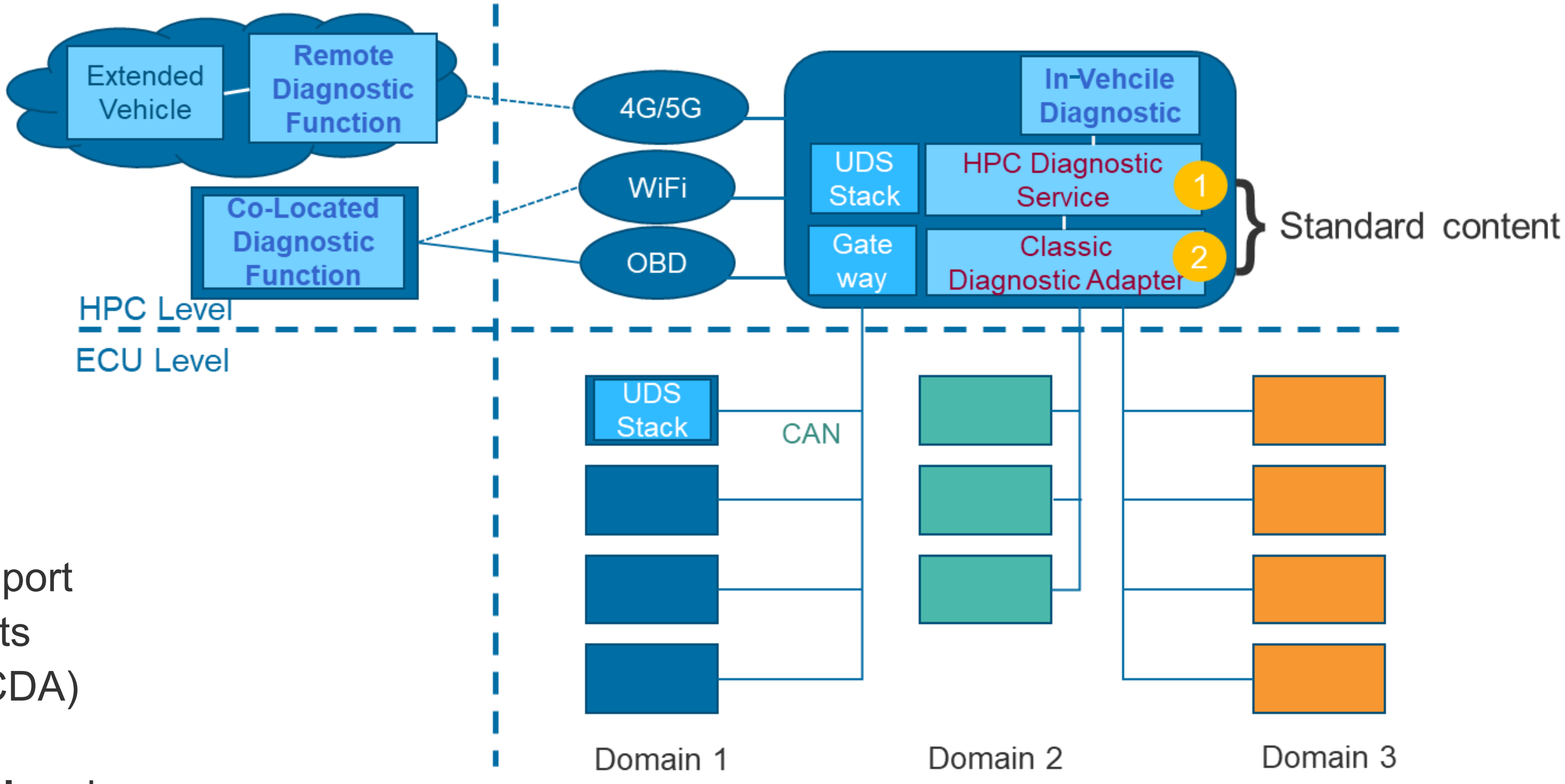
Standard-specific prefix for the resource (e.g., ePTI)

Tag the resource (e.g., ePTI)
=> Prefix and tag are defined by the referencing legislated standard

ASAM SOVD allows to query tagged entities and resources (filtering with tag)

# ASAM SOVD

## Classic diagnostic adapter




ASAM SOVD is capable to support legacy diagnostics in UDS via its “Classic Diagnostic Adapter” (CDA)

The CDA encapsulates **stateful** and **signal-based** communication

# ASAM SOVD

## Key advantages

- **Diagnostics of next-gen architectures:** For diagnosing HPCs (as used in SDVs) AND for traditional diagnostics
- **Little implementation effort:** No automotive-specific stack required on client-side
- **Life-cycle support:** Used for all phases of the vehicle life-cycle, from development to end-of-life
- **Usage scenarios beyond diagnostics:** Service-defined data access also for fleet management, predictive maintenance, consumer applications, etc.)

- **Flexible implementation:** API specification only; implementation can be freely defined
- **Consistent capability descriptions:** Identical format for offline and online use, based on OpenAPI
- **Consumer-driven insights:** Data evaluation fully controlled by the consumer

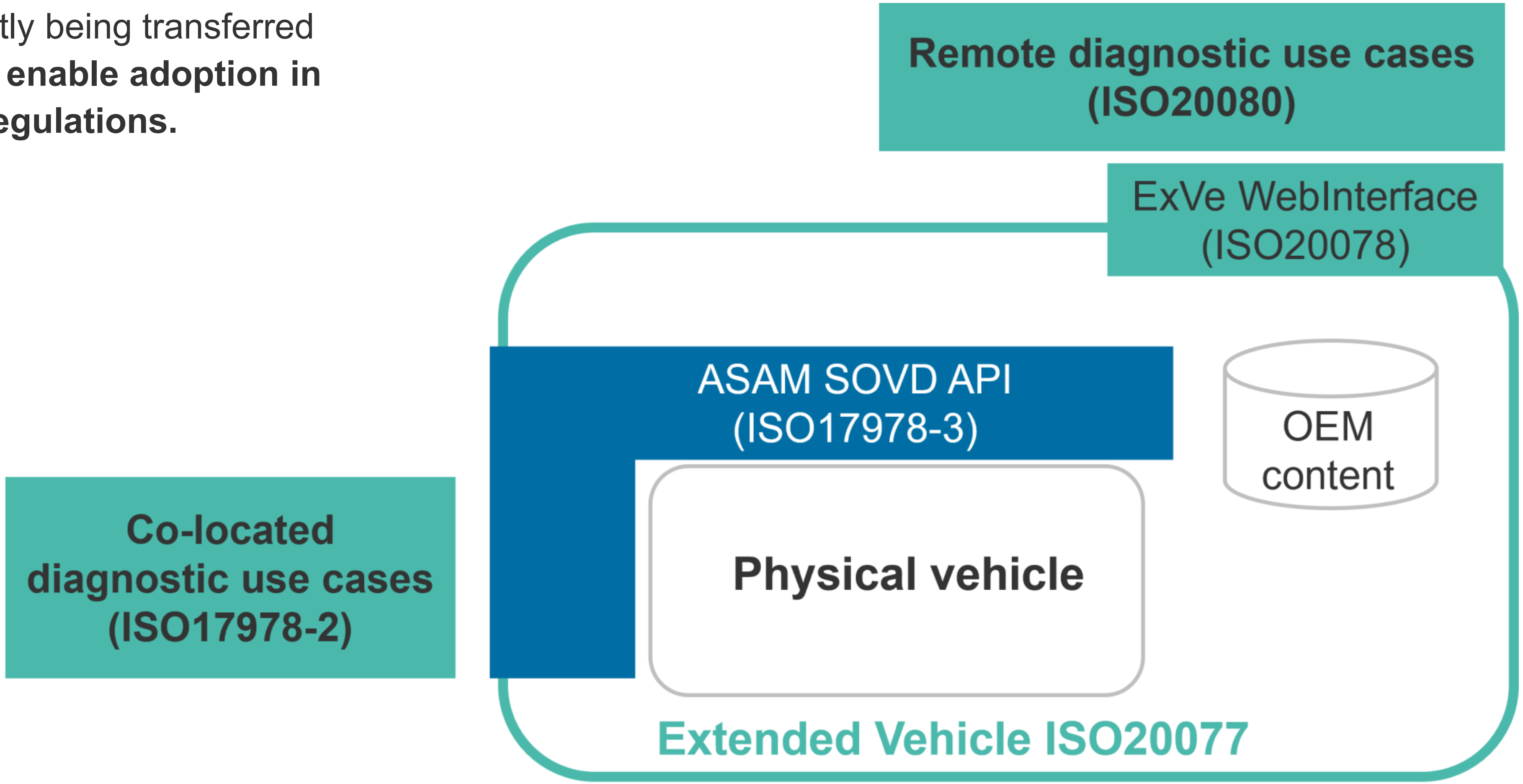
# Agenda

| | |
|---|---|
| 1 | ASAM SOVD |
| 2 | ASAM SOVD – Development Timeline |
| 3 | Proposed content V 1.2.0 |
| 4 | ASAM OTX Extension for ASAM SOVD |

# ASAM SOVD

Relationship between ASAM SOVD and ISO Extended Vehicle

ASAM SOVD is currently being transferred to ISO with the goal **to enable adoption in future international regulations.**



# ASAM SOVD

Timeline inside the standardization landscape

**Sep 2019**
Kick-off
ASAM SOVD 1.0.0

**Jul 2022**
Release
ASAM SOVD 1.0.0

**Nov 2025**
Kick-off
ASAM SOVD 1.2.0

**Jan 2026**
Release
ISO 17978-3

**Mar 2028**
Release
ASAM SOVD 1.2.0

# Agenda

| | |
|---|---|
| 1 | ASAM SOVD |
| 2 | ASAM SOVD – Development Timeline |
| 3 | Proposed content V 1.2.0 |
| 4 | ASAM OTX Extension for ASAM SOVD |

# ASAM SOVD

Cluster: Project overview

| | |
|---|---|
| Proposal workshop: | Aug 05, 2025 |
| Content Definition workshop: | Oct 14 – 15, 2025 |
| Enrollment: | **Oct 20 – Nov 14, 2025** |
| Kick Off = Project start: | Dec 01, 2025 |
| Project duration: | until end of 2027 (release spring 2028) |
| Goal: | Further development of version 1.1.0 |

Minimum Required resources: **50 PD**

**Work principal:**

- work load assigned to participants of clusters
- requirement adjustment inside cluster
- requirement confirmation by project group
- solution finding within clusters
    - if no volunteers are found for a cluster, topic is out of scope
    - cluster members define the work sequences within their clusters.
- the final confirmation of solutions will be given by the project group
- OSD is planned for solution integration

# ASAM SOVD

Overview about clusters and topics: Details see
https://www.asam.net/conferences-events/detail/asam-sovd-120-content-definition-workshop

| Cluster | Topics |
|---|---|
| Generic Features | - Packaging of capabilities description<br>- Binary serialization<br>- Server- / Client communication (WebSockets, HTTP2) |
| Traditional Diagnostic Use Cases | - Enhancements on fault filtering<br>- Software Update Enhancement – White Box approach<br>- Information about a client performing modifications on the vehicle |
| Data Access | - Access to historic data<br>- Value & Unit handling<br>- Mapping |
| SDV Diagnostics | - Functional Diagnostics<br>- Dependencies between services<br>- Client management |

# ASAM SOVD

Overview about clusters and topics: Details see
https://www.asam.net/conferences-events/detail/asam-sovd-120-content-definition-workshop

| Cluster | Topics |
|---|---|
| Scripts and Triggers | - Bidirectional communication<br>- Trigger-based and cyclic start of scripts<br>- Triggers with multiple conditions |
| Bulk Data | - Checking Server Storage Before Bulk Data Upload<br>- Support for Complex Query Expressions in Bulk Data APIs<br>- New bulk-data category creation |
| Logging | - Streaming of Logging Information<br>- Context Information of Logging data |
| Functions | - Define API for SOVD Function<br>- Introduce Broadcast in API |
| Classic Adapter | - Adjustment of UDS<br>- Legislated access with J1979-1<br>- Introduce J1939 adapter |

# Agenda

1. ASAM SOVD
2. ASAM SOVD – Development Timeline
3. Proposed content V 1.2.0
4. ASAM OTX Extension for ASAM SOVD

# ASAM OTX (Open Test sequence eXchange format)

## Published by ISO, developed in ASAM

The core and some standard extensions were developed in ISO (parts 1 - 3).
To ensure faster development, benefit from strong organizational expertise, and maintain close alignment with industry needs, expanded extensions are developed within ASAM (ASAM OTX Extensions) before being submitted to ISO as parts 4 and 5.

## Key benefits of OTX

- OTX is a domain specific programming language (DSL) for process reliable **description of exchangeable and executable test logic** inside automotive industry
- OTX ensures that identical, unchanged test logic can be executed on any target system at any time, delivering consistent results
- OTX can be easily extended in compliance with the standard
- OTX can be used for applications inside new areas outside of vehicle diagnostics (beyond traditional vehicle diagnostics)

# ASAM OTX

## Extension overview

| | | | | |
|---|---|---|---|---|
| DiagCom | DiagComPlus | Flash | FlashPlus | ExternalServiceProvider |
| DiagDataBrowsing | DiagConfiguration | i18n | HMI | Measure |
| DiagDataBrowsingPlus | EcuConfiguration | Logging | SQL | DataTime |
| BusMonitoring | | | | Persistence |
| ComInterface | | | | EventHandling |
| VehicleInfo | | Core Processing System | | EventPlus |
| BlackBox | TestResultHandling | DiagComRaw | ZipHandling | StringUtil |
| StateVariable | File | Xml | Quantities | Math |
| DataType | CommonDialogs | Assertion | Util | SateMachineProcedure |
| EcuSateChart | JSON | SOVD | | Range |

Developed by ASAM

Developed by ISO

# ASAM OTX

SOVD Extension

**Purpose:**

- Execute diagnostic operations via the ASAM SOVD API

**Key features:**

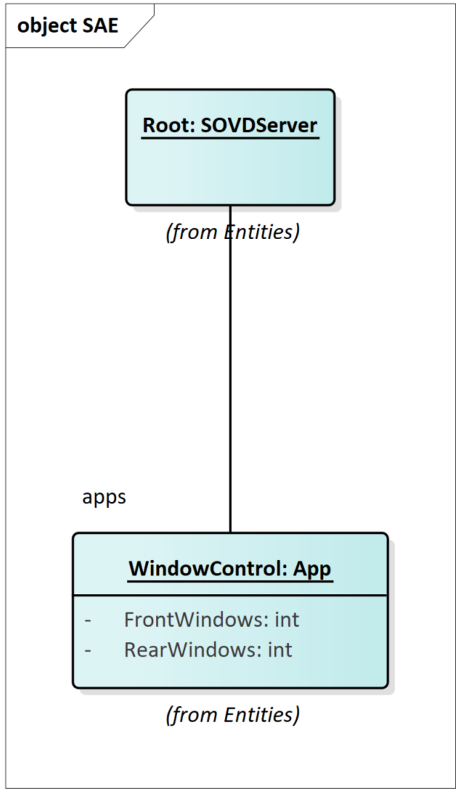
ASAM SOVD Extension is based on JSON extension

- All ASAM OTX Extensions contain
    - DataTypes
    - Enumerations
    - Exceptions
    - Variables
    - Terms
    - Actions

Knowledge of offline capability description at authoring time is presumed

# ASAM OTX

## SOVD Extension – Descriptor Concept




```
ServerDescriptor myServerDescriptor = GetFirstDiscoveredServer();

otx:Map<otx:String, AppDescriptor> myApps =
                        GetAppDescriptors(myServerDescriptor);

AppDescriptor windowControlAppDescriptor =
                        myApps["WindowControl"];

otx:Map<otx:String, DataDescriptor> myDataDescriptors =
                        GetDataDescriptors(windowControlAppDescriptor);

DataDescriptor rearWindowsDataDescriptor =
                        myDataDescriptors["RearWindows"];


Data rearWindows = GetData(rearWindowsDataDescriptor);
```

Descriptor is a JSON object and encapsulates the reference to the entity

# ASAM OTX

## SOVD Extension – Read signature usage

The data resource `rearWindows` is transformed into a structure.

```
org.asam.SovdSignatures.SovdDataStructure rearWindowStruct =
json:GetElementAsStructure(rearWindows, SovdReadValueStructure);

otx:String myId                = rearWindowStruct.id;
json:JsonElement myData = rearWindowsStruct.data;

otx:Integer positionLeft =
                     json:GetElementAsInteger(myData, "/PositionLeft");

quant:Quantity positionRightQuant =
                     GetElementAsQuantity(myData, "/PositionRight");
```

# Thank you for your attention

Bernd Wenzel
Senior Technical Consultant
ASAM e.V.

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