

ASAM SOVD

Implementation Example

Reena Parekh

21 March, 2023
Fürstenfeldbruck



Agenda

SOVD V1.0.0 Implementation Example

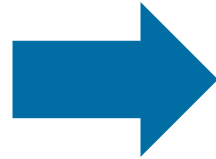
1	Motivation
2	Technology Stack
3	Define SOVD APIs
4	Implement SOVD Server
5	Deploy SOVD Server
6	Demonstration - Remote Diagnosis

SOVD

Motivation for new Standard

Why SOVD ?

- Software-defined vehicles
- Dynamic software updates
- Zone-based E/E architectures
- HPCs with virtualization
- Guest Operating Systems
- Heterogeneous Middleware



What is SOVD ?

- Next generation diagnostics through Http/REST based APIs
- Allows diagnostics of new age entities like HPCs, Apps
- Analysis of software and systems in addition to UDS based diagnostics
- Uniform APIs for proximity, remote and in-vehicle use cases
- Allows diagnosis for the entire lifecycle of vehicles

Technology Stack

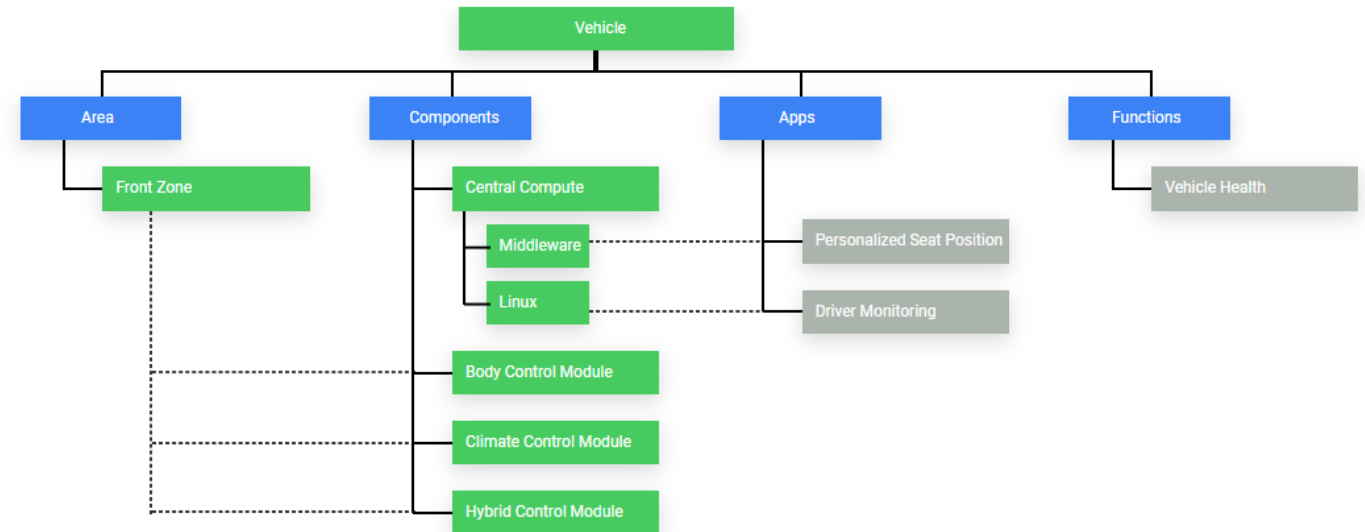
Used for SOVD Implementation

- API with REST principles
- Open API specifications to define APIs
- Http 1.1/Http 2.0 protocol
- Diagnostic information in form of resources
- Http Methods to access resources
- JSON data types
- OpenID Connect and OAuth 2.0 for security
- TLS for securing end-to-end communication

Define SOVD API

Using Vehicle Topology information

- SOVD API and the hierarchy of APIs are defined by Capability Descriptions
- Clients traverse the topology and discover entities, its resources and methods to interact with them.
- Clients access resources via entity path, derived from element's entity collection name and the entity identifier
e.g.,
{base_url}/components/Central_Compute/subcomponents/Linux



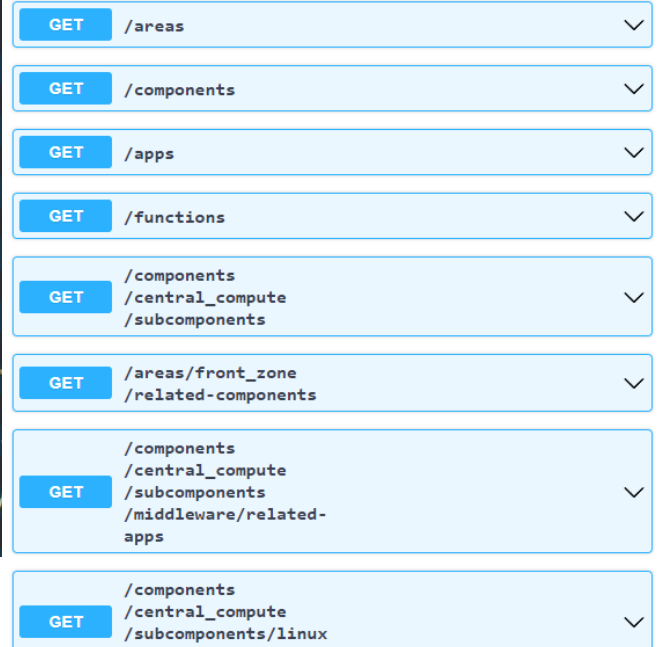
Define SOVD API

Using Capability Descriptions

- OpenAPI specifications 3.1.0 and above can be used for capability description creation in JSON/YAML formats
- OpenAPI fields like servers, paths, components are used for defining SOVD API
- OpenAPI extension mechanisms are used for defining SOVD specific extensions e.g., x-sovd-unit

```
1 {
2   "openapi": "3.1.0",
3   "info": {
4     "title": "ASAM SOVD API",
5     "description": "This is a simple API for SOVD",
6     "version": "1.0.0"
7   },
8   "servers": [
9     {
10    "url": "https://tcssovdserver.com/TCS-SOVD/API/1.0.0",
11    "description": "ASAM Technical Seminar SOVD demo"
12    }
13  ],
14  "paths": {
15    "/docs": { ...
34  },
35    "/areas": { ...
77  },
78    "/components": { ...
135  },
136    "/apps": { ...
185  },
186    "/functions": { ...
228  },
229    "/components/central_compute/subcomponents": { ...
275  },
276    "/areas/front_zone/related-components": { ...
319  },
320    "/components/central_compute/subcomponents": { ...
358  },

```

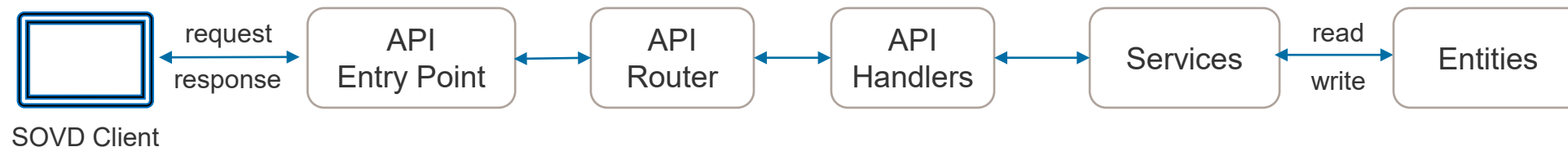


GET	/areas	▼
GET	/components	▼
GET	/apps	▼
GET	/functions	▼
GET	/components/central_compute/subcomponents	▼
GET	/areas/front_zone/related-components	▼
GET	/components/central_compute/subcomponents/middleware/related-apps	▼
GET	/components/central_compute/subcomponents/linux	▼

Implement SOVD server

Using Capability Descriptions

- The capability descriptions can be used to generate SOVD framework
- Services can be implemented as per requirements
- Services shall read and write resources from/to entities
- In case of classical components,
 - CDA converts SOVD request to UDS request
 - Its UDS client sends UDS request to components and receives back the response

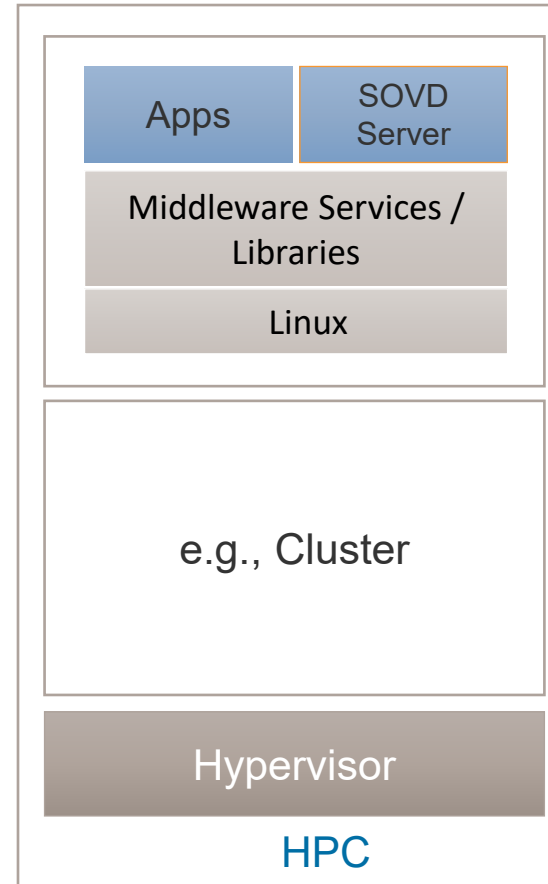


SOVD Implementation framework

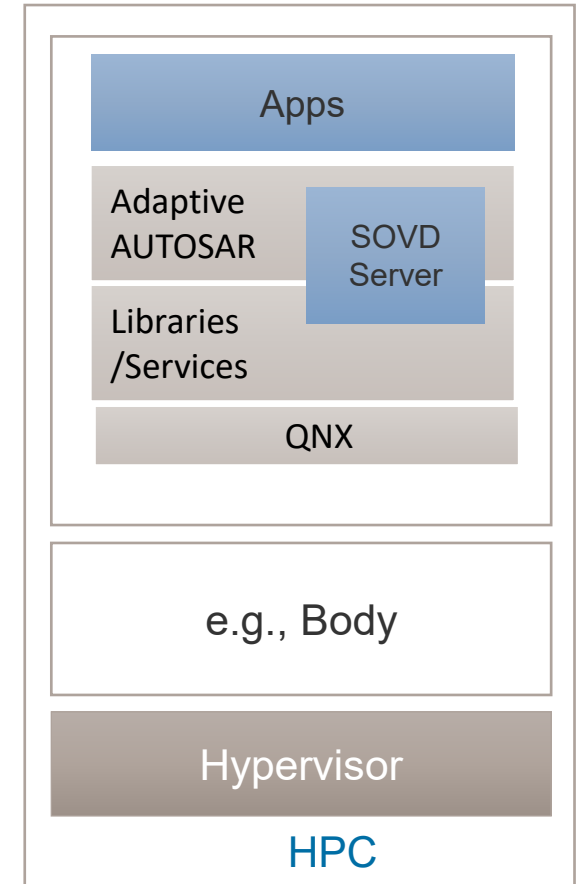
Deploy SOVD in Vehicle

Security & Discovery

- The deployment of SOVD can be in a AUTOSAR or non-AUTOSAR environment
- For remote use cases, the communication path for server discovery can be :
 - configured at startup or
 - discovered dynamically at runtime, using service registry.
- For Proximity, ASAM defines mechanism of mDNS and DNS-SD protocol for server discovery.
- OAUTH 2.0 can be used for client authentication & authorization. TLS for securing end-to-end communication channel



Deployment Example 1



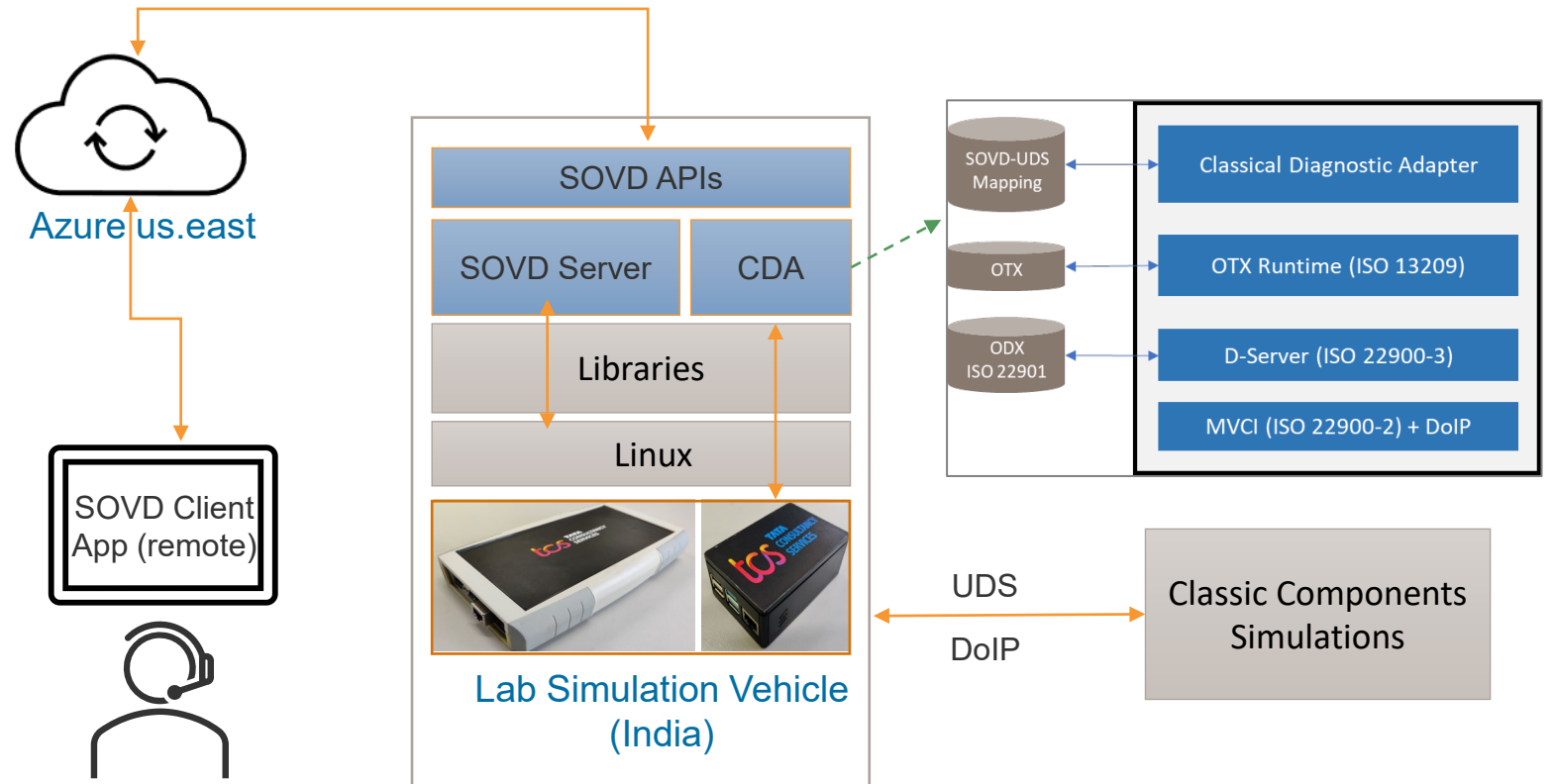
Deployment Example 2

Live Demonstration – Remote Diagnosis

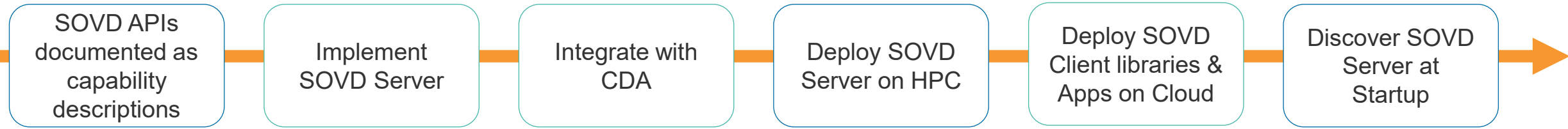
Use Cases and Setup

Use cases –

- Entity and resource discovery
- Fault reading
- Data reading
 - Parameters (classic)
 - System Information (HPC)
- Operations



Workflow



Thank you for your attention !

Reena Parekh
Tata Consultancy Services Ltd.
reena.parekh@tcs.com