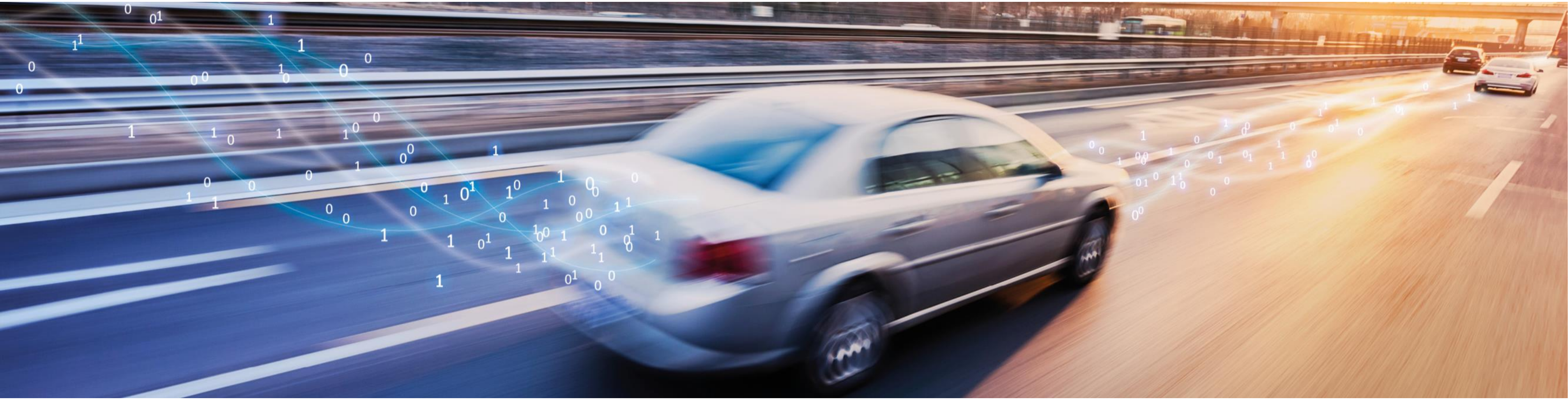
SOVD Overview

Tobias Weidmann
Vector Informatik

28. October 2020
Stuttgart



ASAM Association for Standardization of Automation and Measuring Systems

Agenda

1	Next Generation Diagnostics
2	Use Case Overview
3	What could SOVD look like

Next Generation Diagnostics

Approach

- UDS is still the choice for diagnostics of embedded systems, but it is insufficient for several new use cases.

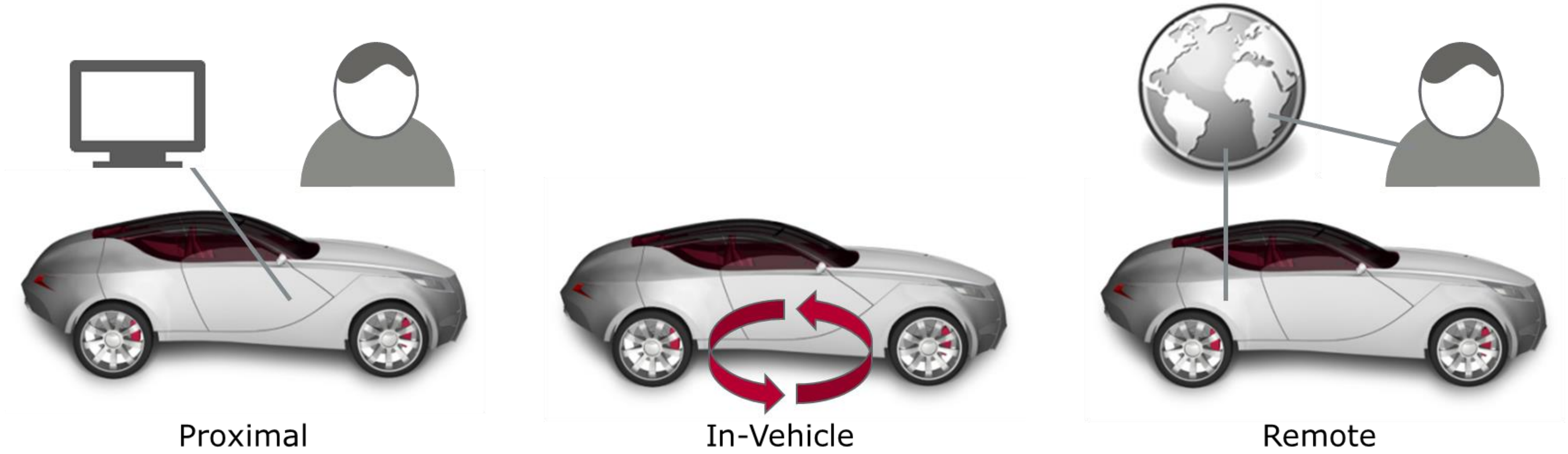
Standardize the next generation of diagnostics beyond UDS:
SOVD (Service-oriented Vehicle diagnostics)

- Specify an API which allows to diagnose and update new and traditional systems
- Standardization committee in ASAM (registered association), target to work out proposal for ISO
- Handle obvious dependencies to ISO (UDS, ODX, ...) and AUTOSAR (Classic and Adaptive)

Next Generation Diagnostics

Subject of Standardization

- Provide 3 scenarios: Proximal, in-vehicle, remote diagnostics
- Specify one consistent API for new systems and traditional sensor/actuator diagnostic
- Stay technology agnostic: Reuse/adapt suitable technologies, do not reinvent a new technology!
- Support content discovery, so that you don't need an external data specification in the future
  (but you still need a content description to implement inspection procedures)



Agenda

1 Motivation for SOVD

2 Use Case Overview

3 What could SOVD look like

Use Case Overview

SOVD Use Cases

Data Access

- Reading / writing
- Single Shot / Periodic Trigger based, Push/Pull
- Bulk/Single value
- File/Stream
- Configuration

Vehicle faults

- Read out lists of vehicle fault information
- provide system information around the appearance of a specific vehicle fault.
- **Audience**: Workshop

SW Update

- Check Pre/Post-Conditions
- Push/Pull update package
- "Installation"
- Activate Update

Logging

- Different sources
- Configurable aggregation in a standardized format (syslog-like)
- **Audience**: 1st Level Support

Tracing

- Activation of detailed tracing on demand
- Collect data onboard for later offline analysis
- "Network traffic"
- Application traces
- **Audience**: Developer

System Information

- (Cyclic) Reading of HPC system information
- Trigger based access
- Start/Stop Processes on the HPC

Capability discovery

- Determine diagnostic capabilities interactively

Capability description

- Provide offline description for development of test sequences in P, AS
- Provide a specification for the implementation of the Entity

Multi-Client-Access, Security, Authorization

Agenda

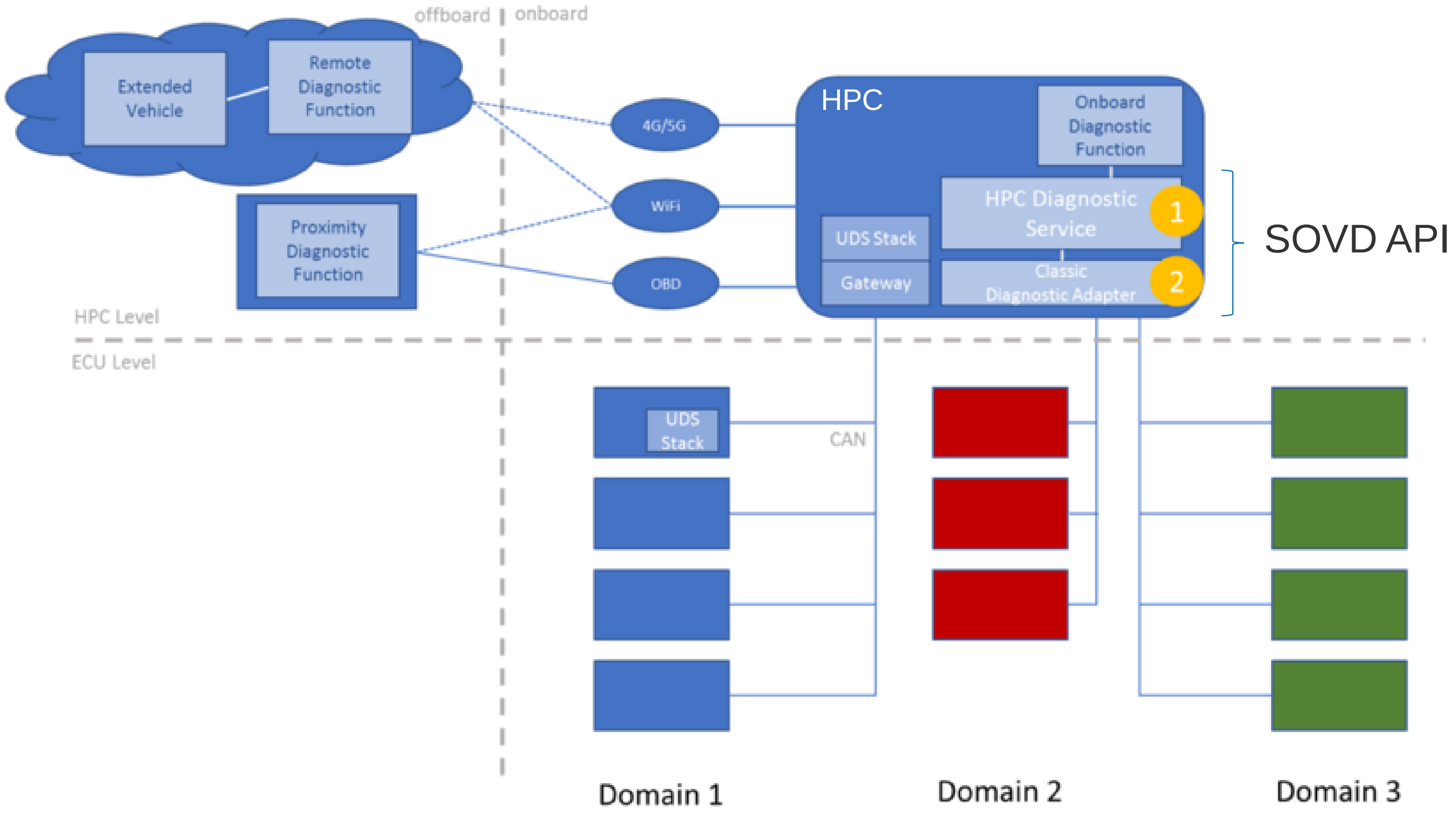
1. Motivation for SOVD
2. Use Case Overview
3. What could SOVD look like

What could SOVD look like?

Disclaimer

As standardization activities are still in progress, and no results are published the following slides show only an interpolation of current activities.

What could SOVD look like?



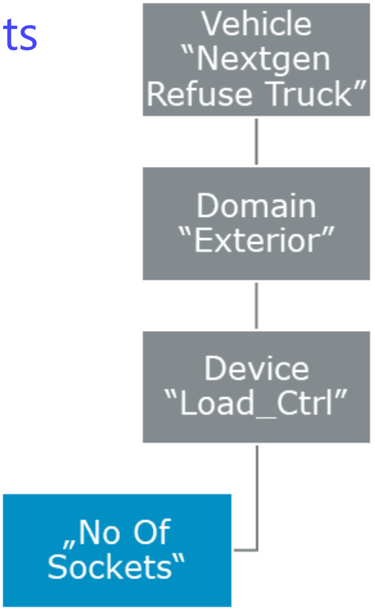
https://www.asam.net/project-detail/sovd-service-oriented-vehicle-diagnostics/

What could SOVD look like?

RESTish approach

SOVD is investigating a HTTP-REST based approach

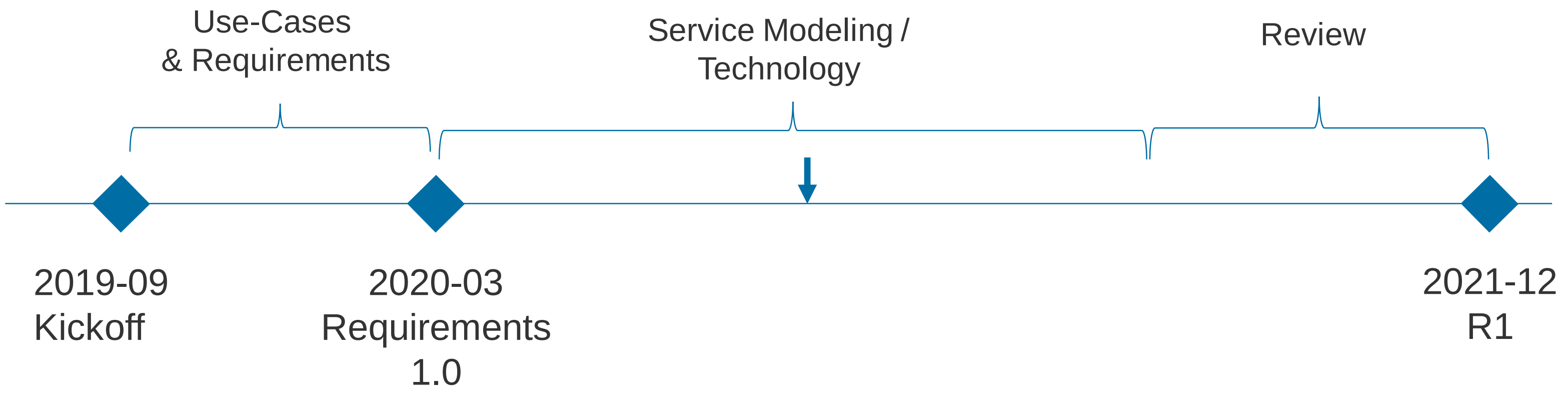
```
GET <vehicle.ip>/domains/exterior/devices/load_ctrl/system_info/no_of_sockets
…
{
  “Number of Sockets used”:152
}
```



- a REST Call is composed by a Resource and HTTP Verb
- Resources identify entity in the hierarchy
- The Verb defines the operation executed on the resource
    - GET
        - Returns either the specific information (e.g. Parameter-Value)
        - In case of a collection, returns the collection of child nodes
    - OPTIONS
        - Returns the list of operations available on a specific resource

What could SOVD look like?

Current state of standardization



- Use-Cases defined
- Requirements defined which the standard shall fulfill

- Current work items
  - Basic technology decisions
  - API design principles
  - API contents for the individual use-cases

Questions?

Tobias Weidmann

Vector Informatik
Manager Customer Services
Tobias.Weidmann@vector.com