HPC Diagnostics

List of Use-Cases for Standardization

Autors:	Ansgar Schleicher, DSA GmbH Thomas Falschebner, Porsche AG Bernd Gottschalk, Daimler AG Axel Weckherlin, BMW AG Editor: Thomas Thomsen, ASAM e.V.
Date:	2019-04-10
Version:	1.0

Content

1 Purpose 2

2 Introduction 3

3 Use-Cases 5

3.1 Use-Case 1: Proximity Diagnostics 5
3.1.1 Definition 5
3.1.2 Standardization Proposal 5
3.1.3 Rationale 6

3.2 Use-Case 2: Remote Diagnostics 6
3.2.1 Definition 6
3.2.2 Standardization Proposal 6
3.2.3 Rationale 7

3.3 Use-Case 3: Onboard Diagnostics 7
3.3.1 Definition 7
3.3.2 Standardization Proposal 7
3.3.3 Rationale 8

4 Architectural Overview of HPC-Diagnostics 9

1 Purpose

The purpose of this document is to define a common understanding of diagnostic use-cases for automotive HPC[*)] devices, which are deemed worthwhile for industry-wide standardization. The use-cases define a specific, diagnostics-related work situation within a vehicle's lifecycle, their fundamental requirements, to-be-standardized interfaces, and their reasons for standardization. This might also include initial ideas and concepts, how the use-cases could be implemented in a future standard.

The diagnostic use-case descriptions, together with an architectural overview description of HPC-diagnostics, shall then serve as input for the ASAM proposal process. Experts in this area can read the document prior to a proposal workshop, decide to participate in this workshop and prepare their own input for the future standardization process. Based on the diagnostic use-cases defined in this document, the participants of the proposal workshop have an opportunity to present their requirements, features or existing solutions that could potentially be used as a starting point for future standard development.

There is common agreement among the authors that public standardization of the use-cases defined in this document is more beneficial rather than to have proprietary tools and interfaces. This allows end-users to setup diagnostic tool chains without major integration efforts, even across multiple companies. Tool vendors can develop products for a global market, instead of putting their efforts in OEM-specific solutions. The proposed standardization for HPC-diagnostics is pro-active and early, i.e. before proprietary solutions become de-facto inhouse standards. This avoids major migration efforts in the future and the costly need of maintaining multiple diagnostic tool chains in parallel for many years.

[*)] HPC - High Performance Computer

2 Introduction

Today's vehicles are diagnosed through the OBD-connector of the vehicle and serial bus communication. The most widely spread bus system in use is the CAN-bus. First vehicle platforms supporting an Ethernet access to the ECUs entered the market. The most widely spread diagnostic protocol across both bus systems is UDS.

This protocol and today's diagnostics serve the following main purposes:

1. Read Event (Trouble) status from ECU (i.e. read out result of ECU's self diagnosis)
2. Read current sensor values (i.e. values of the respective ECU sampled from connected sensors and stored in memory)
3. Read or write static values from/to the ECU (i.e. static identifiers like VIN, part number etc.)
4. Perform actuator tests with the ECU (i.e. check whether an ECU can correctly control its attached actuators as anticipated)
5. Parameterize/code the ECU to the appropriate behavior and equipment status (i.e. parameterize/code engine characteristics, display language, front light type etc.)
6. Reprogram the ECU to update its software

Use-cases 1 - 4 are clearly and solely directed at the diagnosis of the electrical and electronical system controlled by the respective ECU, while the software executing on the ECU cannot be analyzed or diagnosed.

Use-case 5 is not a true diagnostic use-case, but diagnostic protocols have been applied for this task even way before UDS.

The same holds true for use-case 6. It is not a diagnostic use-case. Reprogramming is one possible repair action that can be performed, if vehicle behavior is not correct.

With the introduction of HPCs to the vehicle network, the diagnostic capabilities of UDS seem to reach its limitations.

HPCs allow for:

- multi-core, multi-threaded computing
- virtualized ECUs sharing resources on the same hardware
- high band-width low-latency communication to other HPCs (e.g. via Automotive Ethernet)
- complex operating systems
- complex software and AI-engines that are far beyond regular control functions of today's ECUs

Hence, mechanisms are required to analyze and diagnose not only the electronic aspects of the vehicle network but also the software executing on HPCs.

In addition to HPCs, new connectivity technology enters the market place, that may in the long run obsolete the OBD connector. In any case, it provides new possibilities to access the vehicle. Today, diagnosis is focused on proximity-based use-cases, where a technician or worker is located close to the vehicle, connects an external device and performs his/her diagnostic job. Wi-Fi and mobile broadband network access (4G, 5G) allow for remote and OTA[*)] use-cases (OTA: over-the-air) like remote diagnostics, coding/parametrization, reprogramming. At the same time, HPCs allow to perform diagnostic tasks on-board the vehicle without any permanent external access. We thus have to distinguish between use-cases for *onboard*, *proximity* and *remote* diagnostics, which is a new challenge not covered by today’s prevailing diagnostic stacks, based on UDS (ISO 14229), D-PDU API (ISO 22900-2), ODX (ASAM MCD-2D) and ASAM MCD-3D.

The following three sections will give an idea of the use-cases as seen by the authors of this paper as of today and as a basis of discussion and elaboration. The final section will provide a coarse-grained overview of a possible system architecture.

3 Use-Cases

3.1 Use-Case 1: Proximity Diagnostics

3.1.1 Definition

In this use-case, the person performing vehicle diagnostics is close to the vehicle. This use-case can apply in several occasions:

- Technician in a workshop
- Roadside assistance with a technician present
- Periodical technical Inspection
- Emissions check
- Production
- Development

“Diagnostics” covers the following use-cases:

- Support for detection and localization of faults
  - Reading actual values
  - Reading error or fault Information
  - Reading debugging information (e.g. log files)
- Checking operational status (OK/NOK)
- Performing actuations or stimulations for functional checks
- To be discussed: software updates
- Doing software configuration

Safety, data privacy and security aspects have to be addressed.

3.1.2 Standardization Proposal

The diagnostic API for HPCs shall be standardized. This API provides the interface to test equipment outside the vehicle for the mentioned use-cases.

The equipment used by the technician for diagnostics is connected

- either directly to the vehicle or
- to an entity in the internet which is connected to the vehicle.

The vehicle is connected

- to the Internet or
- directly to the tool to the equipment used by the technician.

The vehicle is connected via one or several of the following connections:

- Wired Ethernet
  - Via OBD connector
  - Via separate connector
- Wi-Fi
- Wide range communication (3G, 4G, 5G)

In the vehicle, one or several HPCs are the counterpart for diagnostic communication.

They provide diagnostic access to

- the HPCs themselves,
- vehicle services (service oriented architecture) and
- ECUs (maybe using UDS protocol) which are behind the HPCs.

The API shall be designed using existing state of the art technologies whenever possible.

The API could be designed as a RESTful API, e. g. using HTTP(S).

3.1.3 Rationale

The existing UDS (ISO14229) diagnosis reaches some limitations when it comes to diagnosing HPCs. This leads to a need for defining a successor.

A standard will be beneficial for several stakeholders:

- OEMs
  - Diagnosis tools from the market can be used.
  - Development costs for the interface can be shared.
- Diagnosis tool manufacturers
  - Multi-brand tools can be offered.
  - Tools with a common interface can be offered to different manufacturers.
- Independent aftermarket
  - Multi-brand test tools can be offered.
- Inspection authorities
  - Easier integration of multiple brands.
- Legislator
  - Possibility to open up the market for HPC diagnosis to the independent aftermarket and inspection authorities.

3.2 Use-Case 2: Remote Diagnostics

3.2.1 Definition

In this use-case the vehicle diagnostics is performed remotely, i.e. over-the-air (OTA diagnostics). This use-case can appear with several sub-use-cases, for example:

- Service advisor in preparation of vehicle service
- Workshop technician for remote trouble shooting on customer demand
- Roadside assistance remotely with no technician present at the vehicle
- Information retrieval via diagnostics for monitoring purposes (e.g. fuel level, battery health check)
- Remote activation of vehicle functions (e.g. remote door opening, remote heating)
- Fleet management

3.2.2 Standardization Proposal

Ideally the diagnostic API itself is the same as described in Use-Case 1: Proximity Diagnostics. The differentiation lies in the different sub-use-cases and their connectivity to the vehicle.

The equipment for diagnostics of the above listed use-cases is connected

- either to the vehicle over the air,

- or to an entity in the internet which is connected to the vehicle over the air.

The vehicle is connected

- over the air via Wide Range communication (3G, 4G, 5G).

Whenever the vehicle is connected to the Internet via Wi-Fi, that is described as part of 3.1.2.

All other aspects of 3.1.2 apply.

In addition, the API shall incorporate boundary conditions and requirements such as

- limited bandwidth and/or bandwidth changes during a diagnostic session,
- breakdown and resumption of communication during a diagnostic session,
- sufficient power level throughout a diagnostic session,
- differentiation of diagnostics capabilities between safe and secure vehicle status and vehicle in driving mode.

3.2.3 Rationale

The same as aspects as for 3.1.3 apply, extended to the additional use-cases of Remote Diagnostics.

3.3 Use-Case 3: Onboard Diagnostics

3.3.1 Definition

In this case, there is no external device or remote application connected to the vehicle and no external operator (service technician, remote help desk) interacts with the vehicle or an application communicating with the vehicle. Rather, diagnostic functions run autonomously within the vehicle. This use case can appear in the following scenarios:

- Implementation of on-board monitors of critical components
- Implementation of predictive/preventive maintenance scenarios
- Implementation of fleet monitoring scenarios (e.g. collecting vehicle status information periodically)

3.3.2 Standardization Proposal

The diagnostic API for HPCs shall be standardized. This API provides the interface to test equipment outside the vehicle for the use-cases 2 and 3 and for the communication inside the vehicle (inter-HPC diagnostic communication, e.g. between the HPCs or a central diagnostic unit), that provides the diagnostic access to the whole vehicle or that has some onboard intelligence to calculate specific diagnostic information onboard for use-cases like predictive diagnostics.

No external equipment needs to be connected to the vehicle for this use case.

-

The vehicle does not necessarily have to be connected or networked to implement this use case.

In the vehicle, one or several HPCs can implement onboard diagnostic use cases.

These may need diagnostic access to

- the HPCs themselves,
- vehicle services (service oriented architecture) and
- ECUs (maybe using UDS protocol) which are connected to the HPCs.

The API shall be designed using existing state of the art technologies whenever possible.

The API could be designed as a RESTful API, e. g. using http(s).

It shall also be evaluated, whether standardized monitoring, heartbeat, watchdog applications should be considered to operate the full diagnostic environment safely and enable standardized overview of all running components.

3.3.3 Rationale

The existing UDS (ISO14229) diagnosis reaches some limitations when it comes to diagnosing HPCs. This leads to a need for defining a successor to access all diagnostic information in the car.

New technologies like SOA architecture with distributed functions over many HCPs and ECUs in a car require a new standard for diagnostics to provide an effective way for diagnostics within software centric systems. This has to be done by every OEM to be able to provide an diagnostic interfaces, that fulfills the need of the future HCPs and vehicle architectures.

A standard will be beneficial for several stakeholders:

- OEMs
  - Diagnosis tools from the market can be used.
  - Development costs for the interface can be shared.
- Suppliers of ECUs and HPCs

4 Architectural Overview of HPC-Diagnostics

The authors’ aim is to define and standardize interfaces and services to cover all of the above use cases in one flexible architecture, instead of defining interfaces and services separately for each of them. It is also important to define an architecture that allows for the integration of classic diagnostics for regular ECUs that will be integrated with the HPCs to a complete vehicle network. The architecture needs to leave enough room for OEM design decisions and needs to cover vehicle platforms that may only have one HPC to those that have multiple.

The general idea of an architecture is, that every HPC can provide a diagnostic service that fully covers the diagnostic capabilities of the domain this HPC controls. This service is based on technology that can be used by

- processes executing onboard the vehicle (onboard use case)
- processes executing on an external test device that is connected to the vehicle via a local network technology (proximity use case)
- processes executing remotely connected to the vehicle via a mobile broadband network or through any internet-based access (remote use case)

In addition, the architecture is intended to leave room for designs where only one HPC is the central diagnostic access for the complete vehicle (even if multiple HPCs are present) and for designs where multiple or even all HPCs within the same vehicle provide a separate diagnostic service for their respective subdomain.

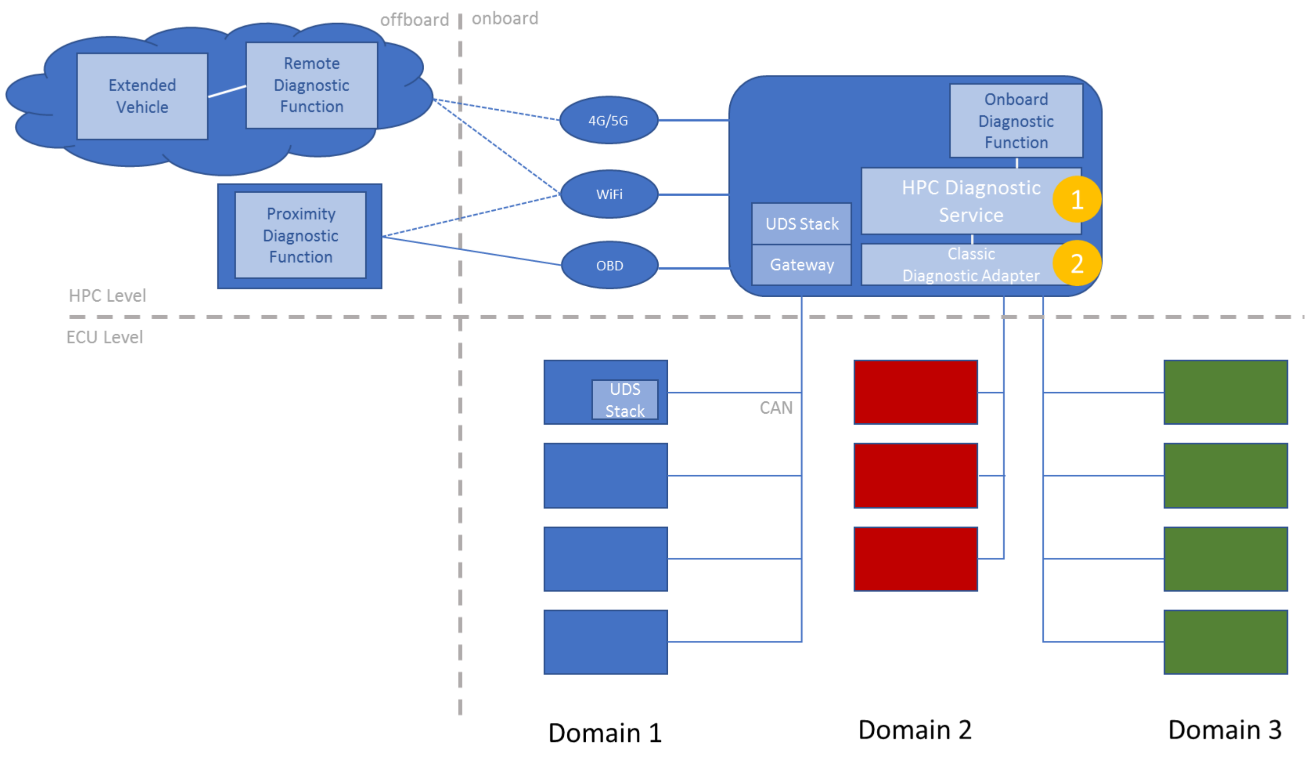

Figure 1 - General Architecture

Figure 1 shows the general idea of the concept within a single-HPC vehicle network. Today’s common ECU diagnostics via protocols like UDS are implemented by a Classic Diagnostic Adapter that connects to the common ECUs to integrate their diagnostic capabilities into the overall architecture and make them accessible for the HPC Diagnostic Service, while simultaneously permitting smooth migration into HPC-based architectures. Of course, the HPC could also take over the full functionality of a gateway and thus provide classic diagnostics via OBD-connector for the full vehicle or a common ECU-based gateway could be connected to the OBD connector to achieve the same result. This OBD-based access to the vehicle will have to be maintained for legislative reasons for years to come. Additionally, the HPC itself should also supply standard UDS diagnostic capabilities (ECU-side), so that it can be diagnosed with classic diagnostic means for legislative reasons.

On top of this, the HPC offers an HPC Diagnostic Service, which is implemented with a service-oriented paradigm. This service can uniformly be accessed by Onboard Diagnostic Modules (e.g. performing a monitoring function for a critical component in the vehicle), by a diagnostic tester with Proximity Diagnostic Modules that is connected via Wi-Fi or Ethernet to the vehicle and is used by a service technician to analyze the vehicle or even by Remote Diagnostic Modules that connect to the vehicle via mobile broadband network. To allow for this wide variety of modules to access the same service, it will be crucial to design authorization and authentication mechanisms that will fine-tune the level of access a particular module will have within the HPC Diagnostic Service. For example, it could be prohibited for a Remote Diagnostic Module to perform an actuator test. This authorization mechanism should also take the current vehicle status into account. For example, it could be prohibited for a Remote Diagnostic Module to change the coding of the HPC, while the vehicle is driving. The same use case could be permitted, as long as the vehicle is not moving.

The new standard shall include API level definitions for the *HPC Diagnostic Service* (see item 1 in Figure 1) and the *Classic Diagnostic Adapter* (see item 2 in Figure 1).

The standard shall ensure compatibility with the Extended Vehicle standards, currently in ISO/DIS status. This is especially the case for Remote Diagnostic use cases.

The design of the HPC Diagnostic Service needs to ensure that communication is possible via non-secure networks with varying latency and varying available bandwidth.

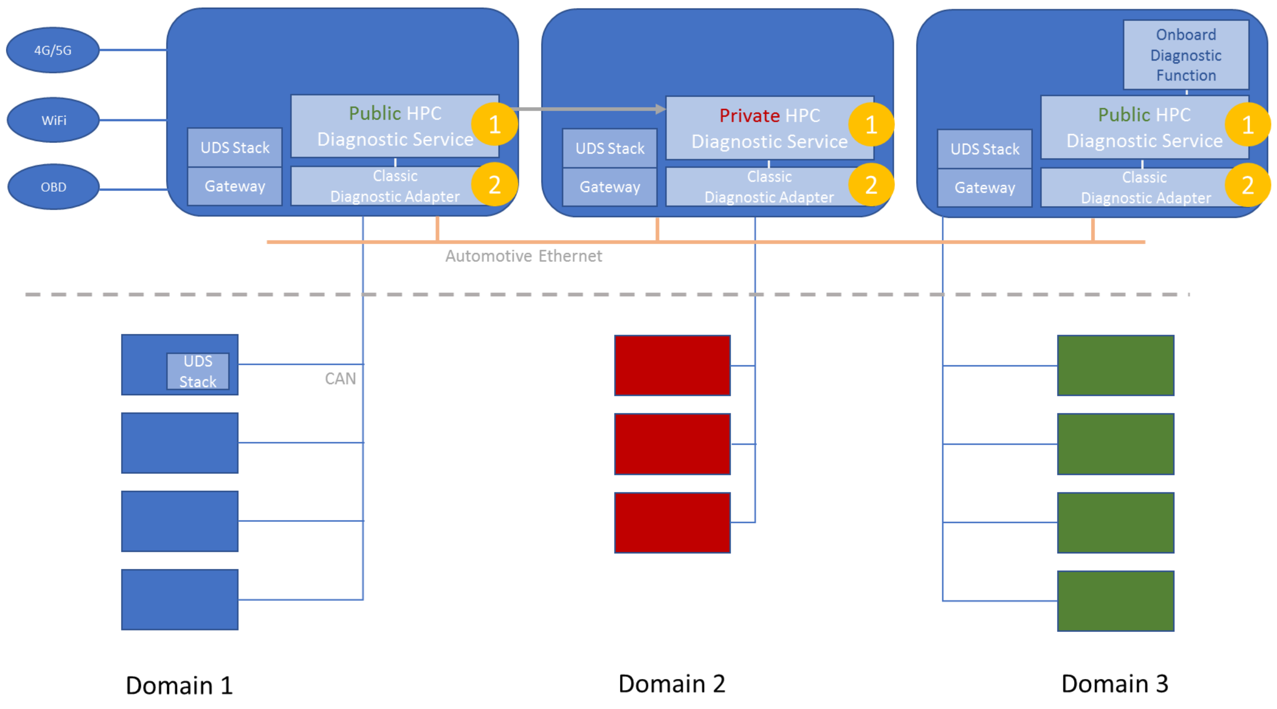
4G/5G
WiFi
OBD
UDS Stack
Gateway
Public HPC Diagnostic Service
Classic Diagnostic Adapter
1
2
UDS Stack
Gateway
Private HPC Diagnostic Service
Classic Diagnostic Adapter
1
2
Onboard Diagnostic Function
UDS Stack
Gateway
Public HPC Diagnostic Service
Classic Diagnostic Adapter
1
2
Automotive Ethernet
UDS Stack
CAN
Domain 1
Domain 2
Domain 3

Figure 2 - Multi-HPC Scenario with multiple HPC Diagnostic Services

In a multi-HPC scenario, the architecture may vary depending on the design goals of the OEM. Figure 2 shows a scenario, where multiple HPCs provide a Diagnostic Service for their domain. Some of these services may be defined as “private” and some as “public”. This terminology is taken from common IT language, where “private” means: non-visible and non-accessible for entities outside a defined context. In this case: non-visible and non-accessible for any application apart from other HPC Diagnostic Services within the same vehicle). “Public” means: visible and accessible for entities outside the defined context, which includes applications outside of the vehicle. Please note that “public” does not imply the access by *any* given entity. Authentication and Authorization procedures apply to fine-tune access levels to the HPC Diagnostic Service. Given sufficient authorization ”public” HPC Diagnostic Services can be accessed by Proximity or Remote Diagnostic Modules. Even Onboard Modules can access services of multiple HPCs to perform their onboard monitoring task. For classic diagnostics to ECUs a regular ECU-based gateway should be employed or one of the HPCs takes over the role of the gateway to provide diagnostics compliant with current legislation.

Depending on OEM’s design rules this concept allows for single-point-of-contact HPCs, where one HPC implements the only public Diagnostic Service, while all other HPCs in the vehicle network implement private Diagnostic Services. At the other extreme, a design could permit a public HPC Diagnostic Service within every HPC. And any combination is also feasible. Figure 2 shows such a mixed approach, where two HPCs publish their own public Diagnostic Service, while one HPC provides a private one.