



Software-Defined Vehicle Support and Coordination Project

D2.1 First Report of high level requirement list

**Author: Mario Driussi
October 2024**

Deliverable	Dx.x – Deliverable name
Work Package(s)	WP2 – Technology and high-level requirements solicitation
Dissemination Level	
Due Date	30-09-2024
Actual Submission Date	15-10-2024
WP Leader	VIF
Deliverable Leader	VIF
Contact Person	Mario Driussi
Email	

Document History			
Revision No.	Date of the review	Name of the reviewer	Status of the document (<i>in progress, ready for review, released</i>)
V0.1	16-09-2024	Mario Driussi	done
V0.1	18-09-2024	Daniel Krippner	reviewed
V0.2	25.09.2024	Mario Driussi	released

This document and the information contained within may not be copied, used, or disclosed, entirely or partially, outside of the **FEDERATE** consortium without prior permission of the project partners in written form

The project has been accepted for funding within the Chips Joint Undertaking (CHIPS JU), a public-private partnership in collaboration with the Horizon Europe (HORIZON) Framework Programme under Grant Agreement No. **101139749**

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the granting authority can be held responsible for them.

Table of Contents

1 Executive Summary.....	5
2 Introduction	6
2.1 Introduction - Purpose of this document	6
2.2 Requirements Template	6
2.3 Definitions.....	7
2.3.1 Building Block Tags	7
2.3.2 The following definitions from the RFC 2119 are applied.	8
3 List of Requirements	9
3.1 High Level Requirements General Requirements.....	9
3.2 High Level Requirements of BB-SC (Building Block Stack Components)	16
4. List of Tables	Fehler! Textmarke nicht definiert.
5. List of Figures	Fehler! Textmarke nicht definiert.
6. Conclusions	24
7. References.....	24

Tables

Table 1: Definitions, Acronyms, Abbreviations	7
---	---

Figures

Figure 1 Snapshot of the Requirements Template.....	6
---	---

1 Executive Summary

This deliverable reports on the FEDERATE WP2 High-level Requirements solicitation.

The requirements, that a successful transformation from ECU-centred vehicle architectures to software-defined vehicles places on the vehicle industry, are extensive and varied. These relate to a wide variety of building blocks. This applies to processes as well as tools, infrastructure or the software components that are combined into software stacks in the vehicle and its environment in the same way.

This is the first of an annual deliverable series, based on the collected high level requirements in WP2.

As described in Deliverable D2.5 Glossary, corresponding tags have been defined for building blocks, which are intended to simplify the categorisation of a building block.

This tagging concept was also applied to the requirements template in order to better categorize the requirements.

In the first version, general requirements for the Software Defined Vehicle and its IT Infrastructure (Cloud Services, Tools, Processes, Interfaces, ...) were collected, as well as special high-level requirements related to BB-SC (Building Block Stack Components).

2 Introduction

2.1 Introduction - Purpose of this document

One of the tasks of WP2 is to collect high-level requirements related to the SDV.

This document lists requirements for the SDV. There are no limitations, borders or restrictions regarding where the requirements come from or what they refer to. This includes processes or toolchains as well as software components or software stack. Neither are there restrictions regarding the environment in which a BB can exist.

This document provides a first set of the high-level requirements solicitation.

The requirements are targeting different Areas and Building Block(s).

2.2 Requirements Template

To collect the high-level requirements an excel template was provided and filled in by the partners.

The minimum level of description is defined via the fields:

Name	Unique name for the requirement
Description	Concise definition of the requirement. One sentence max per one requirement.
Priority	Requirement priority (Must, Must Not, Should, Should Not, May)
Rationale	Explanation why this requirement is necessary.
Metrics	Proof to assess if the requirement is satisfied or not.
Owner	Partner who created this requirement.

WP2 High Level Requirements General							
Leader: VIF							
WP2 (WP2#) & High Level (HL#) & General(G#)	Please use FEDERATE_ as prefix for the code identifying the requirement		Unique name for the requirement	Concise definition of the requirement. One sentence max per one requirement.	Requirement priority (Must, Should, Could, Won't)	Explanation why this requirement is necessary.	Proof to assess requirement is satisfied or not
BB-SC	Nr	ID	Name	Description	Priority	Rationale	Metrics
HLReq_Gen	1	FEDERATE_HLReq_Gen_1	Template/Shape - to describe Building Blocks	A template/shape with a minimal set of descriptive properties should be present.	Must	A Building Block description has to be clear, unique and easy to interpret.	0 or 1
HLReq_Gen	2	FEDERATE_HLReq_Gen_2	Template/Shape - Concept to Classify Building Blocks	Ensure that we are talking about the same thing, a concept should clearly define where BB are located and what kind of BB it is	Must	Tagging concept should help to classify the type of BB.	0 or 1
HLReq_Gen	3	FEDERATE_HLReq_Gen_3	Template/Shape - Building Block must take into account the execution environment/area.	Building blocks are executed in different environments/areas, this has the consequence that e.g. different access controls and authorisation concepts are required or that a public interface is required.	Must	There must be a possibility to refer to different governance documents or to non-functional requirements. e.g. POSIX PSES1, ISO-26262, ISO-21434, ...	0 or 1
HLReq_Gen	4	FEDERATE_HLReq_Gen_4	API/IF - PublicAPI and standardized interface descriptions are mandatory for Building Blocks	Each Building Block in a open source EcoSystem should provide a public API/IF	Should	To use open source and proprietary Building Block together at least an open API/If should be provided	0 or 1
HLReq_Gen	5	FEDERATE_HLReq_Gen_5	Data Centric Architecture should be considered for SDV Building Blocks	A change from application centric to a more data centric architecture must be considered	Must	In order to use the data efficiently, data should be at the centre and tools should be provided accordingly on	0 or 1

Figure 1 Snapshot of the Requirements Template

This template made it possible to assign requirements to the corresponding building block tags defined in the project. This makes it easier to link the requirements to the collected Building Blocks at a later stage.

2.3 Definitions

Table 1: Definitions, Acronyms, Abbreviations

Definitions, Acronyms, Abbreviations	Meaning
API	Application Programming Interface
Area	An area describes a logically and/or technically connected domain (telecommunications, infrastructure, IT services, etc.) which is directly or indirectly involved in the development, operation, maintenance or decommissioning of an SDV and the vehicle industry
ASIL	Automotive Safety Integrity Level
BB	Building Block
BB-CEST	Building Block – Cloud Engineering & Support Tool
BB-CEST-TC	Building Block – Cloud Engineering & Support Tool – Tool Chain
BB-CMU	Building Block – Cloud Mock-up Unit
BB-CSC	Building Block - Cloud Stack Component
BB-EST	Building Block – Engineering & Support Tool
BB-EST-TC	Building Block – Engineering & Support Tool – Tool Chain
BB-MU	Building Block – Mock-up Unit
BB-SC	Building Block - Stack Component
DMR	Dual Modular Redundancy
ECU	Electronic Control Unit
OS	Operating System
RFC 2119	Request for comment, Key words for use in RFCs to indicate requirements level
S-BB	Supporting – Building Block
SDV	Software Defined Vehicle
SDVoF	Software Defined Vehicle of the Future
SOA	Service Oriented Architecture

2.3.1 Building Block Tags

As mentioned at the beginning of the document in the Executive Summary **High Level General Requirements** and **High Level BB-SC Requirements** have been collected.

- General Requirements are on SDV level
- BB-SC Requirements refer to:

BB-SC: Building Blocks Stack Component(s) were defined as executable code or usable library within the car itself or within the development environment.

This could include proprietary implementations as well as open-source software or interfaces and application programming interfaces (APIs), which need to be implemented by specific components.

2.3.2 The following definitions from the RFC 2119 are applied.

1. **MUST** This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.
2. **MUST NOT**: This phrase, or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.
3. **SHOULD**: This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
4. **SHOULD NOT**: This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
5. **MAY**: This word, or the adjective "OPTIONAL", mean that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option **MUST** be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option **MUST** be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

3 List of Requirements

3.1 High Level Requirements General Requirements

The following high-level requirements relate to SDV in general. It is about uniform interfaces and APIs that should be standardised. General architectural features are also discussed here, as well as new approaches such as data-driven applications.

ID	FEDERATE_HLReq_Gen_1
Name	Template/Shape - to describe Building Blocks
Description	A template/shape with a minimal set of descriptive properties should be present.
Priority	Must
Rationale	A Building Block description has to be clear, unique and easy to interpret.
Metrics	n/a
Owner	VIF

ID	FEDERATE_HLReq_Gen_2
Name	Template/Shape - Concept to Classify Building Blocks
Description	Ensure that we are talking about the same thing, a concept should clearly define where BB are located and what kind of BB it is
Priority	Must
Rationale	Tagging concept should help to classify the type of BB.
Metrics	n/a
Owner	VIF

ID	FEDERATE_HLReq_Gen_3
Name	Template/Shape - Building Block must take into account the execution environment/area.
Description	Building blocks are executed in different environments/areas, this has the consequence that e.g. different access controls and authorization concepts are required or that a public interface is required.
Priority	Must
Rationale	There must be a possibility to refer to different governance documents or to non-functional requirements. e.g. POSIX PSE51, ISO-26262, ISO-21434, ...
Metrics	n/a
Owner	VIF

ID	FEDERATE_HLReq_Gen_4
Name	API/IF - PublicAPI and standardized interface descriptions are mandatory for Building Blocks
Description	Each Building Block in an open source Ecosystem should provide a public API/IF
Priority	Should
Rationale	To use open source and proprietary Building Block together at least an open API/If should be provided
Metrics	n/a
Owner	VIF

ID	FEDERATE_HLReq_Gen_5
Name	Data Centric Architecture should be considered for SDV Building Blocks
Description	A change from application centric to a more data centric architecture must be considered
Priority	Must
Rationale	In order to use the data efficiently, data should be at the center and tools should be provided accordingly.
Metrics	n/a
Owner	VIF

ID	FEDERATE_HLReq_Gen_6
Name	API agnostic of programming language
Description	The API/IF shall be defined agnostic to any programming language and operating Systems
Priority	Must
Rationale	Portability across various language (C, C++, RUST, etc..) and OS (Android, Linux, ROS, etc..)
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_Gen_7
Name	Layered architecture with Building Block
Description	The SDV architecture shall establish a layered architecture to facilitate portability of application on execution platforms
Priority	Must
Rationale	Architecture shall be layered with Hardware Abstraction and Middleware to abstract E/E architecture
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_Gen_8
Name	Building Block organization
Description	The SDV architecture shall enable modular design technique to facilitate development of application
Priority	Must
Rationale	Architecture shall be based on Building Blocks organization
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_Gen_9
Name	Building Block enabling Cloud native application
Description	The SDV architecture shall enable cloud native application for future mobility solutions
Priority	Must
Rationale	Architecture shall consider Building Block for Cloud interaction
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_Gen_10
Name	Building Block enabling data driven application
Description	The SDV architecture shall enable data driven application in car and outside
Priority	Must
Rationale	Building shall provide services for data driven applications development not affecting vehicle behavior
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_Gen_11
----	-----------------------

Name	Building Block update after SOP
Description	The SDV architecture shall support Post SOP over the air updates with guarantee on privacy, security, safety, performance and efficiency
Priority	Must
Rationale	Building Block for OTA are mandatory
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_Gen_12
Name	Tool Framework for development
Description	The SDV architecture and Building Block organization shall enable collaborative environment for development of application
Priority	Must
Rationale	Tool framework for development
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_Gen_13
Name	Container execution
Description	The Building Block shall provide services to execute application in container to ensure isolation
Priority	Must
Rationale	Containers bring benefit of isolation
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_Gen_14
Name	Micro-services architecture
Description	The Building Block shall provide micro-services to control the application
Priority	Must
Rationale	Microservices is a state of art solution for controlling MW layer
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_Gen_15
-----------	------------------------------

Name	Logging and tracing services
Description	The Building Block shall provide services for monitoring, logging, tracing
Priority	Must
Rationale	Control of services is mandatory to supervise the platform execution
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_Gen_16
Name	Load balancing services
Description	The Building Block shall provide services for routing, load balancing applications
Priority	Must
Rationale	Load balancing across architect in Cloud is state of the art to be also for embedded part
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_Gen_17
Name	Safe distribution of application
Description	The building block shall enable secure and fail-safe distribution of application
Priority	Must
Rationale	The platform shall provide services to guarantee safe and secure distribution to ensure automotive grade quality of the platform
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_Gen_18
Name	Services architecture
Description	The messaging between building block includes service-oriented architecture and supports publication-subscription and request-response patterns
Priority	Must
Rationale	SOA is state of art for SW platform
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_Gen_19
-----------	------------------------------

Name	Services integrity
Description	The SDV architecture shall define building block to support dynamic SDV capabilities (aligned with SOA architecture) by providing system integrity check services supporting communication and execution integrity (QoS, runtime configuration, etc.), secured and mastered configuration, etc.
Priority	Should
Rationale	Support implementation of full dynamic SOA service for SDVoF
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_Gen_20
Name	Services for data lifecycle
Description	The SDV architecture shall define solution for separation of data plane and control plane to enable flexibility data access and distribution
Priority	Should
Rationale	Support the control data for software Data Define Communication
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_Gen_21
Name	Stateless service concept
Description	The service of the Building Block shall be defined stateless
Priority	Should
Rationale	Independence and scaling of service (that can be a parameter at the end for example)
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_Gen_22
Name	SDV E2E pattern architecture
Description	The architecture shall define a concept for End to End pattern independent of data transport
Priority	Should
Rationale	Concept important for communication
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_Gen_23
-----------	------------------------------

Name	Architecture pattern
Description	The use of SDV architecture services shall be defined with pattern to define interaction between services
Priority	Should
Rationale	Enabling to understand the use of building block in the architecture for a dedicated configuration
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_Gen_24
Name	Safety observation
Description	The SDV architecture shall provide means to observe errors and failures of BBs and underlying hardware components
Priority	Should
Rationale	Means to enable safety of the entire function / system
Metrics	n/a
Owner	FZI

ID	FEDERATE_HLReq_Gen_25
Name	Building Block implementation supporting safety patterns
Description	Safety patterns such as dual modular redundancy (DMR) potentially require multiple instantiations of the same entities. The BB implementations should support that.
Priority	Should
Rationale	Means to improve safety of the entire system
Metrics	n/a
Owner	FZI

ID	FEDERATE_HLReq_Gen_26
Name	Monitoring interfaces
Description	The BBs should provide interfaces for monitoring their behavior for e.g. safety observation purposes
Priority	Should
Rationale	Means to enable safety of the entire function / system
Metrics	n/a
Owner	FZI

ID	FEDERATE_HLReq_Gen_27
-----------	------------------------------

Name	Analysis services
Description	The SDV architecture should provide anomaly analysis, for e.g. functional decisions, monitoring data or safety-related events or event patterns
Priority	Should
Rationale	Means to support analysis of the running system and its functionality
Metrics	n/a
Owner	FZI

3.2 High Level Requirements of BB-SC (Building Block Stack Components)

The requirements described here relate to stack components (Building Block Stack Component) in particular. These describe features and interfaces, extended functionalities which are expected from SDV InVehicle Stack Components.

ID	FEDERATE_HLReq_SC_1
Name	VehicleAPI and BB-SC(s)
Description	Each Stack Component should be analyzed with respect to the VehicleAPI
Priority	Should
Rationale	Every SW component in the vehicle has states, provide information on events, or resource requirements (e.g. data throughput), etc. This (actual) status information should be accessible via a standardized VehicleAPI if required.
Metrics	n/a
Owner	VIF

ID	FEDERATE_HLReq_SC_2
Name	Middleware API
Description	The embedded middleware shall provide standardized interface to application layer and hardware abstraction layer to enable porting of application and distribution of application over the different ECUs on the E/E architecture
Priority	MUST
Rationale	The middleware API shall be standardized
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_SC_3
-----------	----------------------------

Name	Middleware API extension
Description	The embedded middleware shall have standardized API providing extension mechanism for specific extension
Priority	MUST
Rationale	The middleware shall be open to specific competitive extension
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_SC_4
Name	Middleware data control
Description	The embedded middleware shall provide services for data control in the embedded platform such as data collection, data saving with persistency, data logging for debug
Priority	MUST
Rationale	The data control on the platform is necessary and has to provide high level services independent of ECU (abstraction)
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_SC_5
Name	Middleware Power Management
Description	The embedded middleware shall provide services to enable power control of the vehicle on each ECU
Priority	MUST
Rationale	Power management enables to minimize power consumption at the vehicle level.
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_SC_6
Name	Middleware State management
Description	The embedded middleware shall provide services to control and distribute the state of the vehicle to allow application run on dedicated state mode.
Priority	MUST
Rationale	State management enables to control exclusive execution of application in specific vehicle mode (exclusion of condition for application execution)
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_SC_7
-----------	----------------------------

Name	Middleware Time distribution
Description	The embedded middleware shall provide services to control and distribute the time at the vehicle level, support multiple sources of time, their synchronization, diagnosis, integrity check and mitigation of time source in case of failure
Priority	MUST
Rationale	Time distribution is mandatory for all services to ensure common time reference across and outside the vehicle
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_SC_8
Name	Middleware Health Management
Description	The embedded middleware shall provide services to monitor ECU and application execution (runtime and signal failure detection), and provide mitigation for service completion.
Priority	MUST
Rationale	Health management of the execution platform and application is necessary to guarantee safety goals of the system.
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_SC_9
Name	Fail operational Middleware
Description	The embedded middleware shall support context and services of fail operational vehicle.
Priority	MUST
Rationale	Fail operation is mandatory for AD
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_SC_10
-----------	-----------------------------

Name	Middleware for vehicle diagnosis
Description	The embedded middleware shall provide services to perform distributed diagnosis on the E/E architecture.
Priority	MUST
Rationale	Standard feature of a vehicle for garage diagnostics purpose
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_SC_11
Name	Middleware cloud connectivity
Description	The embedded middleware shall provide microservices for Cloud connectivity.
Priority	MUST
Rationale	Standard feature for Cloud connection
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_SC_12
Name	Middleware shadow
Description	The embedded middleware shall provide services to isolate and run vehicle controller or functions using system data, interacting with a supervisor mode but not disturbing real vehicle control.
Priority	MUST
Rationale	Shadowing concept for real time validation of function
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_SC_13
Name	Safe part of middleware
Description	The embedded middleware shall support service to execute up to ASIL D application.
Priority	MUST
Rationale	Safety critical function of function is mandatory and requires specific OS and services measures.
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_SC_14
Name	Open part of middleware
Description	The embedded middleware shall support service to execute QM apps.
Priority	MUST
Rationale	Open world for execution of apps is necessary for SDV.
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_SC_15
Name	Middleware apps control
Description	The embedded middleware shall provide microservices to supervise embedded execution, control of non-critical apps.
Priority	MUST
Rationale	Control of QM apps is necessary for the SDV features.
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_SC_16
Name	Middleware services for reprogramming
Description	The embedded middleware shall provide service for Over the Air reprogramming, controlled by a Master in the vehicle, ensuring authentication and security of the function and rollback in case of problem.
Priority	MUST
Rationale	Secure OTA services is mandatory
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_SC_17
Name	Middleware security services
Description	The embedded shall implement state of the art services to ensure secure communication and control of the platform (cryptographic algorithm, trusted area, secured protocol, etc..)
Priority	MUST
Rationale	Security state of the art solutions are mandatory.
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_SC_18
----	----------------------

Name	Middleware security supervision
Description	The embedded middleware shall provide service to supervise security inside the vehicle and report incidents.
Priority	MUST
Rationale	Detection and supervision of security are mandatory.
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_SC_19
Name	Middleware service oriented
Description	The embedded middleware shall support Services oriented protocol to enable flexible networking.
Priority	MUST
Rationale	SOA is state of the art for SDV.
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_SC_20
Name	Middleware service-oriented integrity
Description	The embedded middleware shall support service to guarantee the integrity, secure and safe, of the dynamic configuration provided by the services oriented architecture.
Priority	MUST
Rationale	Integrity checks of the dynamic configuration bring by SDV SOA middleware.
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_SC_21
Name	Middleware vehicle communication
Description	The middleware shall support standard interface communication and protocol including infrastructure communication (for charging but also for V2X, etc..)
Priority	MUST
Rationale	Support all type of communication protocol requires for SDV and extend feature communicating outside the vehicle.
Metrics	n/a
Owner	Conti

ID	FEDERATE_HLReq_SC_22
-----------	-----------------------------

Name	Generic Middleware Interface
Description	A generic middleware interface abstracts away all aspects that are specific to a concrete middleware implementation and allows to switch between supported middleware implementations during compile time.
Priority	Should
Rationale	Allows developers to decouple their application logic from specific middleware implementations, enabling easier maintenance and flexibility to switch or upgrade middleware without modifying the core application code, enhancing code portability and reusability across different environments and projects.
Metrics	n/a
Owner	

ID	FEDERATE_HLReq_SC_23
Name	Middleware Sensor API
Description	A general API must be provided to integrate sensors in a generic way into the architecture.
Priority	MUST
Rationale	Mapping data flows on the architecture, making specific sensors exchangeable.
Metrics	n/a
Owner	FZI

ID	FEDERATE_HLReq_SC_24
Name	Middleware transport and communication technologies
Description	The middleware must include means to support intra and inter processor communication for service-oriented architectures.
Priority	MUST
Rationale	Support of communication for SOA.
Metrics	n/a
Owner	FZI

ID	FEDERATE_HLReq_SC_25
Name	Perception monitoring
Description	The middleware should support runtime environment and supporting analyses such as schedulability or memory consumption, for instantiation of the perception system monitoring architecture.
Priority	Should
Rationale	Robustness of perception and hence safety (SotIF)
Metrics	n/a
Owner	FZI

ID	FEDERATE_HLReq_SC_26
-----------	-----------------------------

Name	Middleware Container Abstraction
Description	The middleware should provide a container abstraction so that the orchestrator can manage and execute the container runtime provided by the platform.
Priority	Should
Rationale	Allow Container orchestrator to decouple with available runtime, it provides more flexibility to middleware and also enable easy maintenance.
Metrics	n/a
Owner	Valeo

ID	FEDERATE_HLReq_SC_27
Name	Middleware Service Discovery
Description	The middleware should enable services to dynamically discover each other without any manual configuration, regardless of whether the services are deployed on an edge device or in the cloud. Additionally, the middleware should allow services to move seamlessly between different hosting environments (e.g., from one HCP to another, from cloud to edge, or from edge to cloud) without impacting existing connections or requiring any additional configuration changes. During service discovery, the middleware should also manage access control to ensure that only authenticated services can discover and connect to authorized services.
Priority	Should
Rationale	ensures seamless functionality in dynamic environments by enabling automatic service discovery and reducing manual effort, thereby minimizing downtime.
Metrics	n/a
Owner	Valeo

ID	FEDERATE_HLReq_SC_28
Name	xIPC
Description	The zero copy inter processor communication mechanisms are very HW/SOC-specific and usually each one of them provides its own API and usage model for such mechanism, i.e. IPCF from NXP, MailBox from Qualcomm, IPI from Xilinx, The goal for such BB is to define a standard API to abstract the underline HW specific zero-copy based IPC mechanism. The API shall be suitable to be implemented for different execution environments from RTOS to HLOS. The API shall support dynamic endpoint/port management. through utilizing such API, we could simplify the implementation of the service bus to make it portable for different soc's, supporting different binders for the services.
Priority	Should
Rationale	Interprocessor communication mainly for heterogeneous Multiple processors (HMP) / Asymmetric Multiple Processors (AMP)
Metrics	n/a
Owner	Valeo

4. Conclusions

This is a first list of high-level requirements collected and should be shared with other related projects (e.g. HAL4SDV, or Shift2SDV if excepted) working on this topic or searching for additional requirements which need to be synchronized to get a better understand about gaps and what is needed or what already exists.

In a further step, these requirements are matched and linked to the already collected building blocks. If a requirement is not covered by any building block, this must be assessed, and a new building block created if necessary.

As the list of collected requirements only contains high-level requirements and BB-SC requirements, consideration must be given to how cloud and tool or S-BB requirements can be better queried.

This will have to be discussed in the next work package 2 meetings to be able to take appropriate measures.

5. References

Key words for use in RFCs to Indicate Requirement Levels:

[1] RFC 2119 <https://datatracker.ietf.org/doc/html/rfc2119>

Glossary

[2] D2.3_Glossary_FEDERATE