



Software-Defined Vehicle Support and Coordination Project

D2.2 Second Report of High-level Requirement List

Author: Mario Driussi

October 2025

Deliverable	D2.2 – Second Report of High-level Requirement List
Work Package(s)	WP2 – Technology and high-level requirements solicitation
Dissemination Level	Public
Due Date	30 September 2025
Actual Submission Date	30 September 2025
WP Leader	VIF
Deliverable Leader	VIF
Contact Person	Mario Driussi
Email	Mario.driussi@v2c2.at

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	19-08-2025	Mario Driussi	Initial version
V0.2	26-08-2025	Mario Driussi	Update
V0.3	28-08-2025	Mario Driussi	Update
V0.4	29-08-2025	Mario Driussi	Update
V0.5	02-09-2025	Mario Driussi	Update
V0.6	09-09-2025	Mario Driussi	Update
V0.7	11-09-2025	Mario Driussi	Ready for review
V0.8	19-09-2025	Phillipe Cuenot	Review
V0.9	23-09-2025	Mario Driussi	Incorporating comments
V0.10	26-09-2025	Mario Driussi	Incorporating comments
V1.0	29-09-2025	Mario Driussi	Released
V2.0	29-09-2025	Alexandra Pinto	Final formatting

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

<i>1 Executive Summary</i>	5
<i>2 Introduction</i>	6
2.1 Introduction – Purpose of This Document.....	6
2.2 Requirements against Open-source Collaboration Project in an Industrial Environment (with Respect to Safety, Security, the EU AI Act and Others)	6
2.3 Requirements Template	7
2.4 Definitions.....	8
2.3.1 Building Block Tags	9
2.3.2 The following definitions from the RFC 2119 are applied.	9
<i>3 Status of the Requirements Collection from the Year-1</i>	10
3.1 High-level Requirements General Requirements	10
3.2 High-level Requirements of BB-SC (Building Block Stack Components)	19
<i>4 Overview of Collected Requirements Year-2</i>	30
4.1 Automotive Grade – Open-source Software Projects	30
<i>5 Conclusions</i>	43
<i>6 References</i>	44

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 deliverable consists of two sections:

- The requirements collected in Year-1 with additional information about implementation in research projects, open-source initiatives or other projects, and
- a new section dealing with requirements for open-source software development projects for use in safety and security-relevant industrial environments.

While the first-year report summarised the general requirements collected in WP2, focusing primarily on generic elements and BB-SC requirements (stack components), this new section focuses on requirements for open-source software components aimed to developed in collaborative projects and intended for use in the automotive industry, i.e. can be used without modification in series production projects. To pass automotive product quality gates, the developed artefacts must be demonstrably of high quality: well-documented, thoroughly tested, and developed under processes aligned with Automotive SPICE and the relevant safety and cybersecurity standards (ISO 26262, ISO/SAE 21434), among others. Developing production-ready code collaboratively in open-source is still relatively new to the automotive industry; accordingly, this report adds a second set of high-level requirements that describe the needs for such projects to be successful.

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 high-level requirements around 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 stacks, and furthermore, the approach to cross-company collaboration in open source software implementation projects. There are also no restrictions regarding the environment in which a BB can exist (part of the inVehicle stack, the development toolchain, the coordinated processes, etc).

Section 3 contains the BB requirements from the first annual report, including a focus on (planned) implementation projects.

Section 4 introduces a new set of high-level requirements that should be met by a collaboration project for open source software development in the automotive sector.

2.2 Requirements against Open-source Collaboration Project in an Industrial Environment (with Respect to Safety, Security, the EU AI Act and Others)

The second report of this deliverable addresses the discussions and challenges of collaborative open-source development targeting an automotive industrial setting that produces software for safety-critical systems. Such initiatives must comply with safety and cybersecurity standards while demonstrating high implementation quality. (See ISO 26262 for functional safety and ISO/SAE 21434 for cybersecurity engineering in road vehicles or Automotive SPICE for process maturity measures.) To that end, we compile the cross-cutting requirements that make open-source development successful in this context, offering practical guidance to partners working on SDVoF and the future of digital mobility.

These requirements cover legal and governance aspects (notably license compliance per OpenChain/ISO/IEC 5230), collaboration models and technical alignments to be agreed upfront, communication among participating parties, tooling, and state-of-the-art best practices (including documented processes, evidence, and quality gates). They also include training and onboarding materials for efficient adoption, as well as the documentation of potential business models and ROI to support further evolution. For Open-Source Software (OSS) maturity and process quality, programs such as the OpenSSF (Open-Source Security Foundation) Best Practices Badge may provide an additional reference framework.

2.3 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 for Year-1 requirements:

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.
Impl Project	Mapping to an implementation project

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.

2.4 Definitions

Table 1: Definitions, Acronyms, Abbreviations

Definitions, Acronyms, Abbreviations	Meaning
AGOSSCP	Automotive Grade Open-Source Software Collaboration Project
ABI	Application Binary Interface
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
AUTOSAR	AUTomotive Open System ARchitecture
AUTOSAR CAPI	AUTOSAR Common Adaptive Platform Implementation
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
OSS	Open Source Software
RFC 2119	Request for comment, Key words for use in RFCs to indicate requirements level
S-BB	Supporting – Building Block
SDV	Software-Defined Vehicle

Definitions, Acronyms, Abbreviations	Meaning
SDVoF	Software Defined Vehicle of the Future
SOA	Service Oriented Architecture

2.3.1 Building Block Tags

The requirements collected in Year 2 fall into the S-BB (Support Building Blocks) category and are compiled into a document entitled Requirements for an Industrial Open-Source Software Development Project in the Automotive Environment, which will be tagged with a new S-BB Type.

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 Status of the Requirements Collection from the Year-1

3.1 High-level Requirements General Requirements

The following high-level requirements in section 3 are from the Year-1 report and relate to SDV Building Block and Toolchains. 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.

These Year-1 high-level requirements were consolidated in Year-2 and mapped to planned implementation and research projects, open-source initiatives or other projects. This mapping is intended to highlight the relevance of the requirements and question those requirements that could not be mapped to any project: Are they obsolete, or is there a lack of an initiative to implement them?

List of projects who discuss the collaboration or cooperation on Building Block implementation and projects which are still in the discussion/alignment phase.

- HAL4SDV
- Shift2SDV
- Eclipse S-CORE (OSS Initiative)
- COVESA
- SOAFEE
- AUTOSAR (Standardization CAPI)
- RIGOLETTO
- Code4EV
- TwinLoop
- UP2DATE4SDV
- Eclipse SDV Autowrx
- Eclipse SDV Ankaïos
- Eclipse Automotive API Framework
- Eclipse Arrowhead

To this end, the table containing the requirements was updated by replacing the “Owner” field with “Impl. Project”. This new field contains no, one or more references to implementation projects that implement or realise the requirement or a similar requirement.

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
Impl. Project	FEDERATE

ID	FEDERATE_HLReq_Gen_2
Name	Template/Shate - 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
Impl. Project	FEDERATE

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
Impl. Project	Eclipse S-CORE (https://eclipse-score.github.io/) Planned HAL4SDV (https://www.hal4sdv.eu/) Planned Shift2SDV

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
Impl. Project	Eclipse Automotive API Framework (https://projects.eclipse.org/projects/automotive.autoapiframework) Eclipse S-CORE (https://eclipse-score.github.io/)

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
Impl. Project	COVESA (https://github.com/COVESA/s2dm ; https://covesa.github.io/vehicle_signal_specification/) Eclipse SDV Autowrx (https://github.com/eclipse-autowrx)

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
Impl. Project	WEBASSEMBLY (https://webassembly.org/ ; https://webassembly.org/specs/ ;) WASI (https://github.com/WebAssembly/WASI) COVESA (https://wiki.covesa.global/display/WIK4/Vehicle+API)

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
Impl. Project	Eclipse S-CORE(https://eclipse-score.github.io/) Planned HAL4SDV Architecture (https://www.hal4sdv.eu/) Planned Shift2SDV Architecture aligned with HAL4SDV and Eclipse S-CORE

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
Impl. Project	Eclipse S-CORE (https://eclipse-score.github.io/) Planned HAL4SDV (https://www.hal4sdv.eu/) Planned Shift2SDV

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
Impl. Project	SOAFEE (https://www.soafee.io/) Support planned in HAL4SDV (https://www.hal4sdv.eu/) Support planned in Shift2SDV

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
Impl. Project	Eclipse S-CORE (https://eclipse-score.github.io/score/main/features/communication/docs/requirements/index.html) Planned in Shift2SDV

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
Impl. Project	Planned HAL4SDV (https://www.hal4sdv.eu/) Planned Shift2SDV Planned UP2DATE4SDV (https://www.dlr.de/de/se/forschung-transfer/projekte/up2date4sdv)

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
Impl. Project	Planned HAL4SDV (https://www.hal4sdv.eu/) Planned Shift2SDV Eclipse SDV Projects

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
Impl. Project	Eclipse S-CORE (https://eclipse-score.github.io/score/main/features/orchestration/index.html) Eclipse Ankaaios Planned HAL4SDV SOAFEE (https://www.soafee.io/)

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
Impl. Project	Planned HAL4SDV (https://www.hal4sdv.eu/) Planned Shift2SDV

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
Impl. Project	Eclipse S-CORE (https://eclipse-score.github.io/score/main/features/diagnostics/index.html) Eclipse OpenSOVD https://github.com/eclipse-opensovd

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
Impl. Project	SOAFEE (https://www.soafee.io/) Eclipse S-CORE (https://github.com/eclipse-score)

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
Impl. Project	Planned HAL4SDV (https://www.hal4sdv.eu/) Planned Shift2SDV AUTOSAR CAPI (https://www.autosar.org/capi)

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
Impl. Project	Planned HAL4SDV (https://www.hal4sdv.eu/) Planned Shift2SDV Eclipse Arrowhead (https://github.com/eclipse-arrowhead) AUTOSAR CAPI (https://www.autosar.org/capi)

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
Impl. Project	Planned HAL4SDV (https://www.hal4sdv.eu/) Planned Shift2SDV Planned AUTOSAR CAPI (https://www.autosar.org/capi)

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
Impl. Project	tbc

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
Impl. Project	Planned Shift2SDV

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
Impl. Project	AUTOSAR CAPI (https://www.autosar.org/capi) Eclipse S-CORE (https://github.com/eclipse-score)

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
Impl. Project	Planned HAL4SDV (https://www.hal4sdv.eu/) Planned Shift2SDV Eclipse S-CORE (https://github.com/eclipse-score) AUTOSAR CAPI (https://www.autosar.org/capi)

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
Impl. Project	Planned HAL4SDV (https://www.hal4sdv.eu/) Planned Shift2SDV AUTOSAR CAPI (https://www.autosar.org/capi)

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
Impl. Project	Planned HAL4SDV (https://www.hal4sdv.eu/) Planned Shift2SDV Eclipse S-CORE (https://github.com/eclipse-score) AUTOSAR CAPI (https://www.autosar.org/capi)

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
Impl. Project	Planned HAL4SDV (https://www.hal4sdv.eu/) Planned Shift2SDV Eclipse S-CORE (https://eclipse-score.github.io/score/main/features/diagnostics/index.html#) AUTOSAR CAPI (https://www.autosar.org/capi)

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
Impl. Project	Planned HAL4SDV (https://www.hal4sdv.eu/) Planned Shift2SDV Eclipse S-CORE (https://github.com/eclipse-score) AUTOSAR CAPI (https://www.autosar.org/capi)

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
Impl. Project	COVESA (https://covesa.github.io/vehicle_signal_specification/ ; https://github.com/COVESA/vehicle-information-service-specification ; https://github.com/eclipse-kuksa/kuksa-databroker ; COVESA VSS/VISS) Eclipse KUKSA (Databroker; Vehicle-facing APIs and signal semantics align with VSS/VISS; KUKSA is a VSS data broker with VISS/gRPC access) Eclipse Automotive API Framework (https://github.com/eclipse-autoapiframework) Planned HAL4SDV (https://www.hal4sdv.eu/)

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
Impl. Project	Planned HAL4SDV (https://www.hal4sdv.eu/) Eclipse S-CORE (https://github.com/eclipse-score) Eclipse Automotive API Framework (https://github.com/eclipse-autoapiframework)

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
Impl. Project	Planned HAL4SDV (https://www.hal4sdv.eu/)Eclipse S-CORE (https://github.com/eclipse-score)Planned Shift2SDV Eclipse S-CORE (https://github.com/eclipse-score) AUTOSAR CAPI (https://www.autosar.org/capi)

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
Impl. Project	Planned Shift2SDV ECLIPSE S-CORE (https://github.com/eclipse-score) AUTOSAR CAPI (https://www.autosar.org/capi)

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
Impl. Project	AUTOSAR CAPI (https://www.autosar.org/capi)

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
Impl. Project	AUTOSAR CAPI (https://www.autosar.org/capi)

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
Impl. Project	Planned HAL4SDV (https://www.hal4sdv.eu/) COVESA VSS (timestamped signals) SOAFEE (https://www.soafee.io/) Eclipse SDV SOVD (https://github.com/eclipse-opensvd) AUTOSAR CAPI (https://www.autosar.org/capi)

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
Impl. Project	Eclipse S-CORE (https://github.com/eclipse-score) AUTOSAR CAPI (https://www.autosar.org/capi)

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
Impl. Project	Eclipse S-CORE (https://github.com/eclipse-score) AUTOSAR CAPI (https://www.autosar.org/capi)

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
Impl. Project	Planned Shift2SDV Eclipse S-CORE (https://github.com/eclipse-score) Eclipse OpenSOVD (https://github.com/eclipse-opensvd) AUTOSAR CAPI (https://www.autosar.org/capi)

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
Impl. Project	Planned HAL4SDV (https://www.hal4sdv.eu/) Eclipse S-CORE (https://github.com/eclipse-score) Planned HAL4SDV (https://www.hal4sdv.eu/) SOAFEE (https://www.soafEE.io/)

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
Impl. Project	Eclipse S-CORE (https://github.com/eclipse-score) AUTOSAR CAPI (https://www.autosar.org/capi)

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
Impl. Project	Planned HAL4SDV (https://www.hal4sdv.eu/) Planned Shift2SDV Eclipse S-CORE (https://github.com/eclipse-score)

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
Impl. Project	Planned HAL4SDV (https://www.hal4sdv.eu/)Planned Shift2SDV Eclipse S-CORE (https://github.com/eclipse-score) AUTOSAR CAPI (https://www.autosar.org/capi)

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
Impl. Project	Planned HAL4SDV (https://www.hal4sdv.eu/) Planned Shift2SDV Eclipse S-CORE (https://github.com/eclipse-score) AUTOSAR CAPI (https://www.autosar.org/capi)

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
Impl. Project	tbcPlanned HAL4SDV (https://www.hal4sdv.eu/) Planned Shift2SDV Eclipse S-CORE (https://github.com/eclipse-score) AUTOSAR CAPI (https://www.autosar.org/capi)

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
Impl. Project	Eclipse S-CORE (https://github.com/eclipse-score) Planned HAL4SDV (https://www.hal4sdv.eu/)Planned Shift2SDV AUTOSAR CAPI (https://www.autosar.org/capi)

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
Impl. Project	Eclipse S-CORE (https://github.com/eclipse-score) Planned HAL4SDV (https://www.hal4sdv.eu/) Planned Shift2SDV AUTOSAR CAPI (https://www.autosar.org/capi)

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
Impl. Project	Eclipse S-CORE (https://github.com/eclipse-score) Planned HAL4SDV (https://www.hal4sdv.eu/)Planned Shift2SDV AUTOSAR CAPI (https://www.autosar.org/capi)

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
Impl. Project	Planned Shift2SDV Eclipse S-CORE (https://github.com/eclipse-score) AUTOSAR CAPI (https://www.autosar.org/capi)

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
Impl. Project	Eclipse S-CORE (https://github.com/eclipse-score) Planned HAL4SDV (https://www.hal4sdv.eu/)Planned Shift2SDV AUTOSAR CAPI (https://www.autosar.org/capi)

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
Impl. Project	Eclipse S-CORE (https://github.com/eclipse-score) Planned HAL4SDV (https://www.hal4sdv.eu/) Planned Shift2SDV AUTOSAR CAPI (https://www.autosar.org/capi)

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
Impl. Project	Planned Shif2SDV AUTOSAR CAPI (https://www.autosar.org/capi)

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
Impl. Project	Eclipse S-CORE (https://github.com/eclipse-score) Planned HAL4SDV (https://www.hal4sdv.eu/) Planned Shift2SDV

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
Impl. Project	Planned HAL4SDV (https://www.hal4sdv.eu/) AUTOSAR CAPI (https://www.autosar.org/capi)

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
Impl. Project	Eclipse S-CORE (https://github.com/eclipse-score) Planned HAL4SDV (https://www.hal4sdv.eu/) SOAFEE (https://www.soafee.io/)

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
Impl. Project	Eclipse S-CORE (https://github.com/eclipse-score) Planned Shift2SDV AUTOSAR CAPI (https://www.autosar.org/capi)

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
Impl. Project	Eclipse S-CORE (https://github.com/eclipse-score) Planned HAL4SDV (https://www.hal4sdv.eu/)

4 Overview of Collected Requirements Year-2

4.1 Automotive Grade – Open-source Software Projects

The question of how to develop solutions for required building blocks as quickly and efficiently as possible was answered with a possible approach: the joint development of basic building blocks in open-source projects. In order to successfully launch and manage an open-source project in an automotive context, various aspects, must be taken into account, which are explained in the following requirements.

The collected requirements consider the following aspects to be relevant and worth considering to successfully implement an open-source software project together, and are used to type the topic of the collected requirements:

- **Collaboration** refers to the organized cooperation of developers, researchers, and users in open-source projects. It includes joint code development, repository maintenance, and pull requests, as well as knowledge sharing through documentation and workshops. Successful collaboration forms the foundation for sustainable and long-term project development, as it creates synergies and enables progress beyond individual organizations.
- **Communication** is essential to ensure transparency, traceability, and consensus in open-source projects. It covers asynchronous channels such as mailing lists, forums, or issue trackers, as well as synchronous formats like meetings or chats. Effective communication structures help enabling attractiveness, avoid misunderstandings, clarify priorities, and jointly define project goals.
- **Community**, the community represents the social backbone of an open-source project. It consists of developers, users, supporters, and organizations that contribute to usage and further development. An active and inclusive community is critical for innovation, dissemination, and sustainability. Community-building includes measures such as mentoring, onboarding new members, and promoting diversity.
- **Compliance** refers to adherence to legal, organizational, and technical requirements in open-source projects. Of particular importance are license compliance, data protection, and security regulations. A well-structured compliance management approach prevents legal risks, strengthens partner trust, and facilitates transfer to industrial or commercial application contexts.
- **Efficiency** means making optimal use of resources within the project, whether time, computing power, or personnel. Through automation, continuous integration/delivery (CI/CD), and clear project structures, development cycles can be accelerated and redundancies avoided. Efficiency improvements are not only a technical but also an organizational goal, directly influencing the attractiveness and competitiveness of the project.
- **Exploitation** refers to the strategic use and valorisation of the results of an open-source project. This includes economic exploitation strategies, transfer to industrial applications, as well as the support of start-ups or spin-offs. At the same time, it aims to create added value for users and to increase the visibility of the project beyond traditional open-source channels.

- **Governance** describes the rules, structures, and processes that steer an open-source project. This includes decision-making mechanisms, role distribution, and conflict-resolution guidelines. A clearly defined governance model ensures that the project remains stable regardless of individuals and that the interests of different stakeholders are taken into account.
- **Technical Development** encompasses the actual software and system development within the project. This includes architecture design, programming, testing, documentation, and release management. Technical excellence and the ability to innovate are essential to retain users in the long term and secure a technological advantage over alternative solutions.
- **Technical Harmonization** refers to the alignment and standardization of interfaces, modules, and processes within and across open-source projects. The goal is interoperability, code reusability, and the prevention of fragmentation. Harmonization makes it easier for organizations to integrate open-source software into existing systems and strengthens acceptance in industrial applications.
- **Quality** assurance is a core aspect of the success of open-source projects. It covers code quality, test coverage, documentation, usability, and security. Through peer reviews, automated testing, and clear development guidelines, it is ensured that the software remains reliable, maintainable, and future-proof. High quality also serves as a trust anchor for the community and for potential industrial adopters.

The following requirements were collected under these aspects.

ID	FEDERATE_AGOSSCP_HLReq_Coll_1
Topic	Collaboration
Name	Open-Source Development Collaboration Models
Description	Models for collaboration must be clearly defined.
Priority	Must
Rationale	In a joint open-source software development project, it must be clearly defined who takes on which role, which resources are provided and who is responsible for financing.

ID	FEDERATE_AGOSSCP_HLReq_Coll_2
Topic	Quality
Name	Open-Source Development Collaboration Technical Harmonization
Description	In an open-source integration project that integrates other open-source projects and merges them into a larger one, clear technical specifications must be made in advance (about API, ABI, Version of Tools and Toolchains, Interfaces/Data models, etc),
Priority	Must
Rationale	The target platform, interfaces, APIs and more must be clearly regulated, as otherwise the integration and maintenance effort will take up a large part of the expenditure.

ID	FEDERATE_AGOSSCP_HLReq_Coll_3
Topic	Efficiency
Name	Open-Source Collaboration - Resource Management
Description	An open-source software development project deliberately planned by several companies requires appropriate resource planning.
Priority	Must
Rationale	Without appropriate planning and provision of resources, the expected outcome will not be available in the specified time.

ID	FEDERATE_AGOSSCP_HLReq_Coll_4
Topic	Efficiency
Name	Open-Source Collaboration - Resource Management
Description	The required resources should meet the requirements of the task.
Priority	Should
Rationale	If resources are provided who first have to acquire the skills required for the task, this will result in delays in the execution and achievement of the objectives.

ID	FEDERATE_AGOSSCP_HLReq_Coll_5
Topic	Technical Harmonization
Name	Open-Source Collaboration - Technical Alignment
Description	An open-source sub-project should coordinate and adapt its APIs, ABIs or the target environment with the integration project from the outset.
Priority	Should
Rationale	If no appropriate coordination is carried out, the integration and maintenance effort increases, which leads to considerable delays and an increase in costs.

ID	FEDERATE_AGOSSCP_HLReq_Coll_6
Topic	Technical Harmonization
Name	Continuous Integration into Core Stack
Description	All building block open-source repositories should be required to demonstrate compatibility with the overall system in a continuous integration pipeline that is at least designed / defined centrally (not by each open-source repository) and included by the building block open-source repositories.
Priority	Should
Rationale	If compatibility is not ensured early on (automatically/algorithmically), it may take too much effort later on, resulting in a useless ecosystem of incompatible software.

ID	FEDERATE_AGOSSCP_HLReq_Coll_7
Topic	Technical Harmonization
Name	Open-Source Collaboration - Architecture view
Description	An open-source project shall provide an architecture view to identify clustering of BB's and BB's design description.
Priority	Should
Rationale	An architecture overview is mandatory to be able to map BB's and identify reusable BB's across a different open-source architecture.

ID	FEDERATE_AGOSSCP_HLReq_Coll_8
Topic	Governance
Name	Open-Source Collaboration - Roadmap
Description	An open-source project shall define and maintain a roadmap to identify release planning and contents.
Priority	Should
Rationale	Roadmap enable to plan future version/correction of BB's and gives visibility on liveness of the project and potential reuse of BB's.

ID	FEDERATE_AGOSSCP_HLReq_Coll_9
Topic	Governance
Name	Open-Source Development - Governance
Description	The governance of an open-source project shall be defined with different responsibility (duty and right) in regard to member involvement.
Priority	Should
Rationale	The governance with core team member, and member enable to retribute contribution according to the involvement in the project.

ID	FEDERATE_AGOSSCP_HLReq_Coll_10
Topic	Governance
Name	Open-Source Development - Contributor
Description	The contributor of open-source project shall be identified according to role of member defined in the governance.
Priority	Should
Rationale	It is mandatory to identify contributors in assets capable to be addressed directly and being able to measure their activities.

ID	FEDERATE_AGOSSCP_HLReq_Coll_111
Topic	Exploitation
Name	Open-Source Exploitation - License model
Description	An open-source project shall be defined with an open-source license enabling compatible business model for BBs free exploitation out of the project (access to source, distribution, modification, liability endorsement).
Priority	Should
Rationale	Open-Source license shall be clearly defined to enable free reuse in production.

ID	FEDERATE_AGOSSCP_HLReq_Coll_12
Topic	Exploitation
Name	Open-Source Exploitation - Maintenance
Description	The open-source project shall define a business model that secure BBs maintenance either by contractual engagement of contributors and find an operational mode or business model that enable retribution of contributors.
Priority	Should
Rationale	It is essential to secure the maintenance of BBs and community fee associated to governance can a way to cover it (fees like chipJU retribution or AUTOSAR members fee).

ID	FEDERATE_AGOSSCP_HLReq_Coll_13
Topic	Exploitation
Name	Open-Source Exploitation - Dissemination
Description	The open-source shall organize event to provide metrics on liveness of the project, advice roadmap and promote the community.
Priority	Should
Rationale	It is essential to promote the results of the open-source project in order to show liveness and promote attractivity to new members.

ID	FEDERATE_AGOSSCP_HLReq_Coll_14
Topic	Quality - Technical Development
Name	Open-Source Development - Development process
Description	The open-source project shall document and ensure that BBs follow the development process and identify deviations if relevant.
Priority	Should
Rationale	The quality of BBs shall be documented in order to identify the quality status of BBs.

ID	FEDERATE_AGOSSCP_HLReq_Coll_15
Topic	Quality - Technical Development
Name	Open-Source Development - Change control board
Description	The open-source project shall be managed with a change control board to manage the status of the contributions and the roadmap.
Priority	Should
Rationale	The quality of BB shall be verified by an independent board (of the contributor).

ID	FEDERATE_AGOSSCP_HLReq_Coll_16
Topic	Communication
Name	Open-Source Collaboration - Communication
Description	Open-source collaboration projects must implement clear and open communication channels.
Priority	Should
Rationale	All discussions and decisions should be open and transparent for all parties involved.

ID	FEDERATE_AGOSSCP_HLReq_Coll_17
Topic	Technical Development
Name	Open-Source Development process - Artefacts
Description	The BBs of the open-source project shall be delivered with Requirement, Design document, Code review report, Unitary test case and Report, Review Integration test case, optional Safety case if necessary (in addition to source code).
Priority	Could
Rationale	All quality process artefacts are aimed to be delivered with the code at least the one defined in the open-source project and identify gaps according to artefact required to comply with ASIL level quoted/expected.

ID	FEDERATE_AGOSSCP_HLReq_Coll_18
Topic	Communication
Name	Open-Source Collaboration - Promotion
Description	The open-source project shall organize open technical days / Hackathon to promote project results and give more visibility.
Priority	Could
Rationale	Project must be visible to show that it is alive and attract new contributors.

ID	FEDERATE_AGOSSCP_HLReq_Coll_19
Topic	Technical Development
Name	Open-Source Development process - Compatibility
Description	The BBs of open-source project shall ensure at least backward compatibility of the feature, the interface implemented, the Operating systems services, etc.
Priority	Should
Rationale	Backward compatibility of BB must be guarantee.

ID	FEDERATE_AGOSSCP_HLReq_Coll_20
Topic	Technical Development
Name	Open-Source Development process - Branching
Description	The open-source project must be able to hold several parallel branches for maintenance purpose, while branch liveness is decided by the Governance.
Priority	Should
Rationale	Parallel maintenance is mandatory for different version used in production.

ID	FEDERATE_AGOSSCP_HLReq_Coll_21
Topic	Technical Development
Name	Open-Source Development process - Archiving
Description	The open-source project must be able to provide a safe archiving mechanism and repository enabling archive and recovery for long term maintenance (> 15 years).
Priority	Should
Rationale	Archiving and recovery are mandatory for industrial project.

ID	FEDERATE_AGOSSCP_HLReq_Coll_22
Topic	Business Model
Name	Business Models for Open-Source Contributions
Description	Business models for open-source collaboration projects should be worked out.
Priority	Should
Rationale	Without clarity on ROI or possible business models, it will be difficult to attract new partners.

ID	FEDERATE_AGOSSCP_HLReq_Coll_23
Topic	Compliance
Name	Open-Source Projects with Support for Industrial Standards
Description	Open-source projects with high quality standards should mention their expectations with respect to standards to be observed.
Priority	Should
Rationale	High quality is a requirement for code which goes into safety and security critical systems.

ID	FEDERATE_AGOSSCP_HLReq_Coll_24
Topic	Quality - Resource Management
Name	Open-Source Projects with Support for Industrial Standards should offer trainings
Description	Open-source projects with high quality standards should mention their expectations with respect to standards and offer specialized trainings for new project partners or committers.
Priority	Should
Rationale	This improves the quality of contributions, increases efficiency and reduces costs.

ID	FEDERATE_AGOSSCP_HLReq_Coll_25
Topic	Quality
Name	Open-Source project should provide metrics to measure the quality
Description	Open-Source projects shall define quality metrics on project artefacts.
Priority	Should
Rationale	The quality metrics shall be defined and applied on projects artefacts.

ID	FEDERATE_AGOSSCP_HLReq_Coll_26
Topic	Technical Development - Tooling
Name	Open-Source Projects should provide an open toolchain
Description	Open-source projects should provide a documented standardized open-source toolchain for the whole development, deployment, maintenance, operation and outdated software cycle.
Priority	Should
Rationale	To toolchain shall support the whole life cycle of the software project.

ID	FEDERATE_AGOSSCP_HLReq_Coll_27
Topic	Technical Development
Name	Automation
Description	Open-source projects in an industrial environment should rely on automation as much as possible.
Priority	Should
Rationale	Reproducible Artifacts, Tools or other components are increasing the reliability / trust of the components provided.

ID	FEDERATE_AGOSSCP_HLReq_Coll_28
Topic	Technical Development
Name	AI support
Description	Industrial open-source projects should regularly update the tools and methods they use and incorporate new approaches such as AI-assisted engineering into their workflow/process as early as possible.
Priority	May
Rationale	To be more efficient and speed up in development cycles we should adapt or development tools, methods and processes in continuous cycles with best practices which includes AI-assisted tools and methods.

ID	FEDERATE_AGOSSCP_HLReq_Coll_29
Topic	Technical Development
Name	Process Traceability
Description	Open-source projects should use tools and data models that make every change in the project traceable.
Priority	Should
Rationale	Traceability of software artefacts is required to comply with quality and safety standards.

ID	FEDERATE_AGOSSCP_HLReq_Coll_30
Topic	Quality
Name	Guidelines - for all engineering phases
Description	Industrial open-source projects should provide guidelines for each phase of development, which serve as a kind of checklist for those involved.
Priority	Should
Rationale	The application of quality on the projects shall be guided to guarantee consistent high-level quality.

ID	FEDERATE_AGOSSCP_HLReq_Coll_31
Topic	Quality
Name	Use state of the art and best practices
Description	Industrial open-source projects should regularly publish guidelines on best practices and state-of-the-art methods.
Priority	Should
Rationale	To reach a high level of process maturity, the processes shall be continuously improved and optimized.

ID	FEDERATE_AGOSSCP_HLReq_Coll_32
Topic	Community
Name	Regular face-2-face meetings
Description	Industrial open-source projects should hold regular face-to-face meetings and coordination with other projects.
Priority	Should
Rationale	This improves the quality of the synchronization.

ID	FEDERATE_AGOSSCP_HLReq_Coll_33
Topic	Quality
Name	Provenance
Description	The origin of each individual contribution to an open-source project must be apparent.
Priority	Must
Rationale	This improves the quality and is a requirement for a trusted software framework.

ID	FEDERATE_AGOSSCP_HLReq_Coll_34
Topic	Community
Name	Hello World
Description	Open-source projects should provide a well maintained "Hello World" example that makes understanding the usage of building blocks as easy as possible.
Priority	Should
Rationale	This improves the accessibility of the building blocks.

ID	FEDERATE_AGOSSCP_HLReq_Coll_35
Topic	Community
Name	How to contribute
Description	Open-source projects should clearly document in which ways it is possible to contribute to the project. Often there are more ways than just contributing code (e.g. contributing tests or documentation).
Priority	Should
Rationale	Lower the barrier for potential contributors.

ID	FEDERATE_AGOSSCP_HLReq_Coll_36
Topic	Quality
Name	Well-defined Scope
Description	An Open-Source project should address a clearly defined need, and resist feature and scope creep.
Priority	Should
Rationale	OSS is ideal for infrastructure concerns; fully scoped solutions or products are usually too big to succeed in an OSS model, as the problem they solve is too specific (so the target audience is too small).

ID	FEDERATE_AGOSSCP_HLReq_Coll_37
Topic	Collaboration
Name	Actively involved Adopters
Description	An Open-Source project should be a collaboration hotspot between users/adopters of the developed functionality.
Priority	Should
Rationale	If there are no users/adopters actively involved in the project, it is likely irrelevant. Without actively engaged users and adopters, OSS projects lack a clear direction and drive - no users means no reason to exist.

ID	FEDERATE_AGOSSCP_HLReq_Coll_38
Topic	Quality - Resource
Name	Technically competent contributors
Description	An Open-Source project must be driven by competent, technical and dedicated people. The actual development resources allocated to OSS projects are almost always very limited, so there is no place for roles that do not contribute to the actual project codebase.
Priority	Must
Rationale	In an OSS project, nobody cares about PowerPoint, architecture astronauting, theoretical castles made of sand - people need to understand what the code is doing and be able to discuss on that level.

5 Conclusions

The first part, documented in section 3, is a list of high-level requirements collected and 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 understanding about gaps and what is needed or what already exists. 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 is continuously discussed in work package 2 meetings to be able to take appropriate measures.

The second part, starting with section 4, addresses the requirements for collaborative implementation of BBs. To collaboratively develop open-source software that can be considered *automotive grade*, the requirements defined under the ID FEDERATE_**AGOSSCP**_HLReq_Coll_Nr (**A**utomotive **G**rade **O**pen-**S**ource **S**oftware **C**ollaboration **P**roject) should be taken into account and further developed. In this context, *automotive grade* means that software artefacts are produced in such a way that they can be transferred into series production with a minimum of modifications. This poses significant challenges for many open-source projects, as processes and standards such as ASPICE, ISO 26262, ISO 21434, or ISO/PAS 5112 have traditionally not been relevant within the open-source domain.

The requirements that a successful open-source software project must fulfil in an industrial safety and security context are outlined in the second part of the requirement set. However, strengthening open-source development in the automotive sector and accelerating the necessary *mindset change* among decision-makers and stakeholders is mandatory.

6 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

First Report of high level requirement list

[3] <https://federate-sdv.eu/2024/09/30/d2-1-first-report-of-high-level-requirement-list/>

List of Tables

Table 1: Definitions, Acronyms, Abbreviations	8
---	---

List of Figures

Figure 1 Snapshot of the Requirements Template.....	7
---	---