D2.5 Report on requirements of next generation TCMS framework

Project number:	730830
Project acronym:	Safe4RAIL
Project title:	Safe4RAIL: SAFE architecture for Robust distributed Application Integration in roLling stock
Start date of the project:	1st of October, 2016
Duration:	27 months
Programme:	H2020-S2RJU-OC-2016-01-2

Deliverable type:	Report
Deliverable reference number:	ICT-730830 / D2.5 / 1.3
Work package	WP 2
Due date:	September 2018 – M24
Actual submission date:	21st of December, 2018

Responsible organisation:	IKL
Editor:	Iñigo Odriozola
Dissemination level:	Public
Revision:	1.3

Abstract:	Elucidates the set of requirements for next generation TCMS frameworks, considering safety and security aspects.
Keywords:	TCMS, Distributed Frameworks

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 730830.

Editor

Iñigo Odriozola (IKL)

Contributors (ordered according to beneficiary numbers)

Arjan Geven, Derya Mete Saatci, Mirko Jakovljevic (TTT)

Iñigo Odriozola, Asier Larrucea, Ekain Azketa, Lorea Belategi (IKL)

Hongjie Fang (SIE)

Mario Münzer (TEC)

Petr Novobilsky, Dobromil Nenutil (UNI)

Bernd Löhr, Gerhard Weiss, Iris Bosse (NEW)

Bernhard Nölte, Alexander Piechullek-Königer (TÜV)

Youlian Kirov (IAV)

Alberto Previti (NIER)

Disclaimer

The information in this document is provided “as is”, and no guarantee or warranty is given that the information is fit for any particular purpose. The content of this document reflects only the author’s view – the Joint Undertaking is not responsible for any use that may be made of the information it contains. The users use the information at their sole risk and liability.

Executive Summary

SAFE4RAIL and WP2 Context

The main task of WP2 of SAFE4RAIL is to provide the "Functional Distribution" architecture concept for a mixed criticality embedded platform, offering an execution environment for multiple Train Control and Monitoring System (TCMS) application functions with a virtual bus inside the end-system.

This concept shall offer:

- host critical (up to SIL4) and non-critical functions based on strict temporal and spatial partitioning,
- enhance modularity and composability of embedded platforms and architectures, thus reducing the complexity of system design, integration, reconfiguration, verification, certification and maintenance,
- lower the costs and effort of integration and certification for different subsystems and functions.

In such a context, this deliverable aims to elucidate the set of requirements for next generation TCMS frameworks. The requirements are contemplated and segregated into functional and non-functional requirements, interface requirements, followed by safety, security and Reliability, Availability, Maintainability and Safety (RAMS) requirements. The requirements are collected using a number of inputs: D2.1 "Report on state-of-the-art of 'functional distribution architecture' frameworks and solutions", D2.2 "Report on analysis of 'functional distribution architecture' frameworks and solutions" and further, inputs from CONNECTA project partners and Safe4RAIL partner's expertise.

These requirements have been the basis for the design of the Functional Distribution Framework (FDF), which is an instantiation of the Functional Distribution architecture concept and the main goal of WP2. Furthermore, they have been updated during the different phases of the software life-cycle, such, design, implementation, verification and validation. As a result, to the best of our knowledge, this deliverable represents a high-value starting point for defining standardization and certification of next generation of TCMS embedded platforms.

Contents

List of Figures

List of Tables

Definitions

Event	Software message indicating that an action occurred
Process	Thread or group of threads with an isolated memory address space
Partition	Logical unit of isolation with exclusive access to predetermined memory space and to the processor in predetermined time slots
Component	A constituent part of the software which has well-defined interfaces and behavior concerning the software architecture and design and fulfils the following criteria: – it is designed according to “Components” (see EN50128 [6] Table A.20); – it covers a specific subset of software requirements; – it is clearly identified and has an independent version inside the configuration management system or is a part of a collection of components (e. g. subsystems) which have an independent version
Leader / Follower	In a redundant execution context the leader is the node which writes the outputs of a given computation while the follower processes the data but does not update the outputs. Only when the leader is out of service the follower takes over the control.

Table 1: List of definitions.

Abbreviations

API	Application Program Interface
BbW	Brake-by-wire
CA	Certification Authority
COM	Communication
COTS	Commercial off-the-shelf
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
DHCP	Dynamic Host Configuration Protocol
EC	European Commission
ECN	Ethernet Consist Network
ECP	Extended Capabilities Port
ECU	Electronic Control Unit
FDF	Functional Distribution Framework
FDS	Functional Distribution Services
FTPS	File Transfer Protocol over Secure socket layers
HIL	Hardware in the Loop
I/O	Input/Output
IMP	Integrated Modular Platform
MAC	Media Access Control
MD	Message Digest
OSEK	Offene Systeme und deren Schnittstellen für die Elektronik in Kraftfahrzeugen
PKI	Public Key Infrastructure
RAMS	Reliability, Availability, Maintainability and Safety
RBCU	Remote Brake Control Unit

RO	Read-Only
RTOS	Real-Time Operating System
RW	Read/Write
S4R	Safe4Rail
SDT	Safe Data Transfer
SFTP	Secure File Transfer Protocol
SIL	Safety Integrity Level
SL	Security Level
SOTA	State Of The Art
SPI	Serial Peripheral Interface
TCMS	Train Control and Management System
TCN	Train Communication Network
TTDB	Train Topology Database
UTC	Universal Time Coordinated
V & V	Verification and Validation
VCU	Vehicle Control Unit
WDT	Watchdog Timer
WP	Work Package

Table 2: List of abbreviations.

1 General description

This deliverable collects the set of requirements that defines the Functional Distribution Framework (FDF). First of all, chapter 1 gives an overview of the structure of the document and a general description of the FDF, by also explaining the requirement distribution and interdependency on the different Work Packages of Safe4RAIL project. Chapter 2, gathers the requirements in the following main sections: Functional requirements, non-functional requirements, interface requirements, safety, security and RAMS. After this, chapter 3 gives a summary of the outcome of the activities involved in this deliverable in form of a conclusion and chapter 4 contains the list of documents that compose the bibliography. Finally, the annexes provide several traceability matrixes to the requirements introduced in chapter 2:

- Annex A – FDF Components
- Annex B – Safety Countermeasures
- Annex C – Security Countermeasures
- Annex D – Brake by Wire electronic control design
- Annex E – Drive-by-Data
- Annex F – Safe4RAIL WP2-CONNECTA T4.4
- Annex G - Integrated Modular Platform

For the sake of improving the clarity and readability, these traceability matrixes are grouped in annexes. These matrixes are tables used to determine the validity and completeness of FDF requirements in correlation with other systems' requirements or properties. For example, Annex A shows how FDF components described in D2.3 "Report on 'TCMS framework concept' design, security concepts and assessment" cover FDF requirements. The terms in the annexes have been taking as they are from the original documents, for instance, FDF component names corresponds to the ones denoted in D2.3. The terms "FDF" and "The Framework" are used interchangeably along the sections, and the concept means an abstraction middleware that allows the integration of real train functionalities. Although the Simulation Framework (SF) is beyond the scope of this deliverable, the interfaces required to be connected to the SF are described as requirements.

1.1 System description and scope

The next generation TCMS follows a functional distribution architecture in which each of the elements is responsible for fulfilling a set of tasks or functions. Each element is called an ECU (Electronic Control Unit) and, connected by an Ethernet network to each other; it is widespread through the different consists and cars of a train or even several trains.

As can be seen in Figure 1, each of these elements consists of a hardware module which is ECU specific and a set of software components which are generic to every ECU. The framework, coloured in green in the figure below, abstracts the different software components from the hardware and networking layers. Besides, a ECU may also perform readings and writings of values to and from Input/Output (I/O) devices through its hardware layer. The network connects every element and the hardware part of every ECU, i.e., the whole element except for the concrete software components, form what is called the IMP (Integrated Modular Platform).

The Work Package 2 (WP2) and this deliverable focus on the Framework. Each of the ECUs in the system must have a different instance of this Framework, which acts as a generic abstraction layer and works as a virtual bus, so that it makes no difference whether a given software component is on the same unit or a different one. The framework is providing middleware services and is therefore not directly providing computation capabilities. These are provided by the underlying hardware which can be accessed through the framework. It provides services, contains drivers and hosts applications.

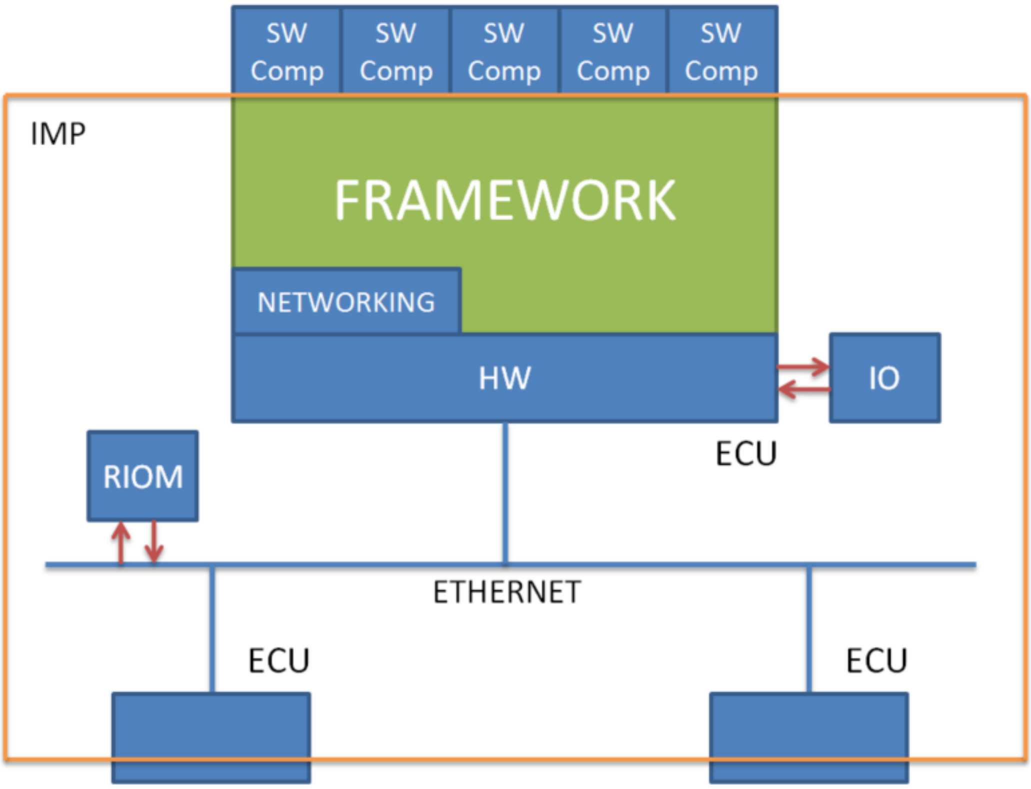

Figure 1: Integrated Modular Platform structure and positioning of the Functional Distribution Framework

1.2 Requirement definition process

A preliminary distinction is made between the Safe4RAIL (S4R) requirements specified for the:

- Generic Safe4RAIL platform (WP1+WP2)
- Specific Brake-by-wire (BbW) application (WP4)
- Simulation environment (WP3)

1.2.1 Generic Safe4RAIL Platform

The generic S4R platform is articulated in three levels:

- **Highest** level, there are requirements provided by CONNECTA for the TCMS and the applicable directives and standards.
- **System** level, there are included:
  - (i) Requirements for the S4R platform which shall satisfy the TCMS User needs.
  - (ii) Requirements for the simulation environment of the S4R platform.
  - (iii) Further sources of requirements coming from results of Safe4RAIL activities, i.e. SOTA or safety analyses on the design concept.

  - (iv) Outputs of the hazard log, which shall be covered by the safety-related requirements of the S4R platform.

- **Sub-System** level, there are requirements for the networking service, I/O interface service and generic functional applications, which shall satisfy the applicable S4R platform requirements.

1.2.2 Specific Brake-by-wire application

The Brake-by-wire application is articulated in three levels:

- **Highest** level: There are the Brake-by-wire user needs to be provided by CONNECTA, the applicable directives and standards and the requirements specified at sub-system level (i.e. for networking service, I/O interface service and generic functional applications) for the generic S4R platform.
- Specific **Application** level, there are included:
  - (i) The Brake-by-Wire requirements, which shall satisfy both the Brake-by-Wire user needs and the generic S4R sub-systems requirements.
  - (ii) Further sources of requirements coming from results of the safety analyses on the Brake-by-Wire preliminary design concept.
  - (iii) Outputs of the hazard log, which shall be covered by the safety-related requirements of the Brake-by-Wire;
- **Lowest** level: there are requirements for the electronic parts of the Brake-by-wire system, which shall satisfy the Brake-by-Wire requirements.

1.2.3 Simulation environment Requirements

The third list of S4R Requirement concerns the Simulation environment, i.e. simulation conditions and instantiation, according to the WP3 objectives. The Simulation Framework's main objective is to validate TCMS subsystems and system by enabling the testing of virtual and/or real equipment at different sites connected via internal LANs or the Internet. For this purpose, the FDF shall support a mechanism so that the FDF can be configured for simulation purposes and a communication interface.

These requirements are not strictly related to the subjects of development (S4R platform and Brake-by-Wire application)

Their collection is recommended, but not mandatory. Indeed, the availability of a set of requirements defining the simulation environment and their relations with the involved requirements specified for the S4R platform and Brake-by-Wire application would contribute to the future V&V activities.

Moreover, it is recommended to consider the activities developed during the WP3 (i.e. the specification of the simulation environment) as a source for derived requirements for Drive-by-Data networking and embedded platform capabilities.

The relationship between the Simulation environment Requirements and the rest of the S4R specifications and activities is shown in Figure 2: Safe4RAIL Global Specification Tree.

1.2.4 Safe4Rail Global Specification Tree

Figure 2 shows the Global Specification Tree of the Safe4RAIL project containing all the set of requirements mentioned before as well as the relationships between the different modules and actions.

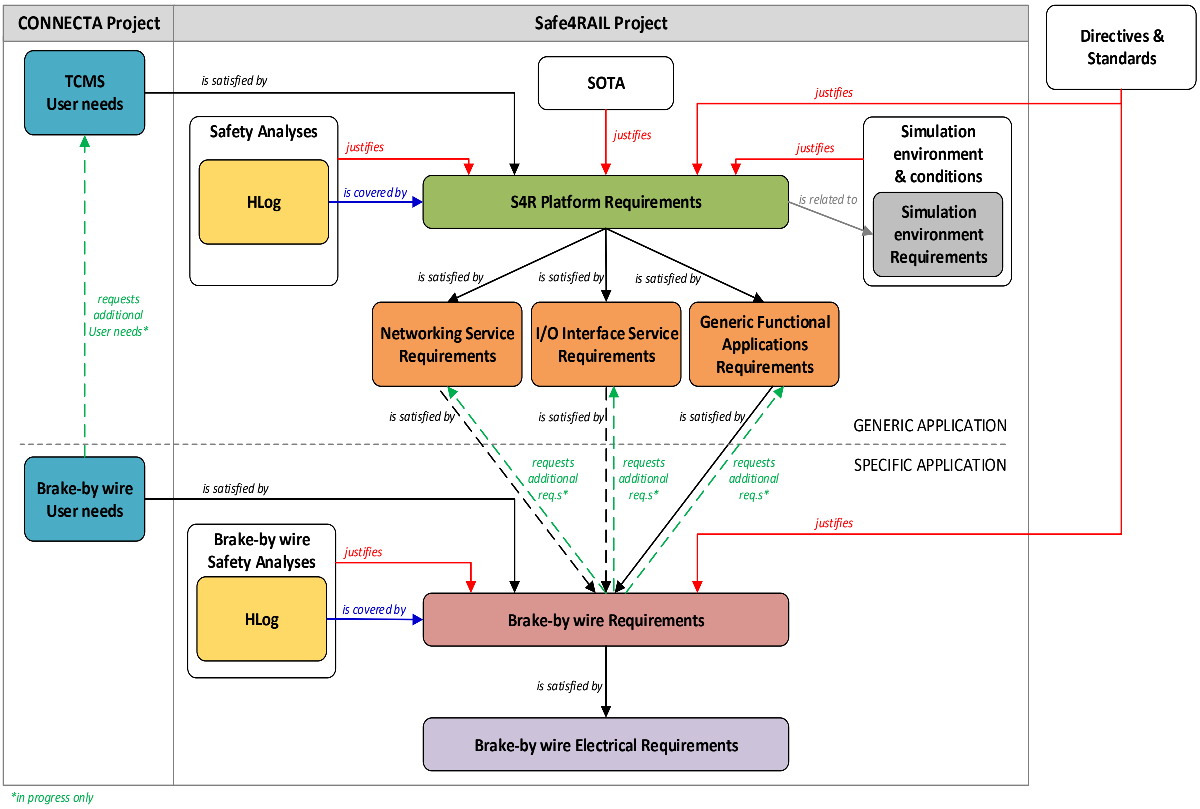
CONNECTA Project
Safe4RAIL Project
Directives & Standards
TCMS User needs
is satisfied by
SOTA
justifies
Safety Analyses
HLog
justifies
is covered by
S4R Platform Requirements
justifies
Simulation environment & conditions
is related to
Simulation environment Requirements
is satisfied by
is satisfied by
is satisfied by
Networking Service Requirements
I/O Interface Service Requirements
Generic Functional Applications Requirements
requests additional User needs*
is satisfied by
is satisfied by
is satisfied by
GENERIC APPLICATION
SPECIFIC APPLICATION
requests additional req.s*
requests additional req.s*
requests additional req.s*
Brake-by wire User needs
is satisfied by
justifies
Brake-by wire Safety Analyses
HLog
justifies
is covered by
Brake-by wire Requirements
is satisfied by
Brake-by wire Electrical Requirements
*in progress only

Figure 2: Safe4RAIL Global Specification Tree

This deliverable focuses on a subset of the S4R Platform requirements, which also hold the Drive-by-Data requirements. The FDF requirements cover both the Generic Functional Applications requirements and I/O Interface Service requirements, which can be found in Chapter 2.3.2.

2 Requirements

On the following table all next generation TCMS requirements for the FDF are gathered. These requirements are managed in DOORS from which they are exported to this document. DOORS is an IBM requirement management tool [10].

In order to ensure the consistency over the deliverable, any change to the set of requirements during and after Safe4RAIL project is handled through DOORS in a centralised way, since only Ikerlan, the editor of this deliverable, is allowed to apply the necessary modifications. The consistent traceability between different modules, kept by permanent links, ensures that no change will be lost in the process.

Id	Text	Safety related	Security related
S4R_FDF_165	**2.1 Functional requirements**		
S4R_FDF_166	***2.1.1 Application execution***		
S4R_FDF_584	**2.1.1.1 Execution management** This subsection defines the execution management which is in charge of handling the execution of application functions and executable instances.		
S4R_FDF_585	The Framework shall support authentication and authorisation of executables at start-up.	n/a	n/a
S4R_FDF_586	The Framework shall check the integrity of executables at start-up.	n/a	n/a
S4R_FDF_737	The Framework shall inhibit the execution of the application function in the case of negative code integrity check.	Yes	n/a
S4R_FDF_766	The Framework shall avoid forcing outputs when application function is operative (nominal and degraded).	Yes	n/a
S4R_FDF_767	The Framework shall prevent the access of off-line services at power-up, during initialization and operation (nominal and degraded).	Yes	n/a
S4R_FDF_768	The Framework shall guarantee the retention of a safe-state after a fatal fault.	Yes	n/a
S4R_FDF_782	The Framework shall be able to generate partitions and allocate resources for application functions requiring multiple-instances for the implementation of reliable and safe architecture.	Yes	n/a
S4R_FDF_588	The Framework shall support multiple executable instances.	Yes	n/a
S4R_FDF_589	The Framework shall consider unambiguous identification of executable instances (i.e., processes) provided by the configuration.	n/a	n/a
S4R_FDF_594	The Framework shall support ordered execution of processes, partitions and FDF components.	n/a	n/a

Id	Text	Safety related	Security related
S4R_FDF_684	The Framework shall guarantee a pre-emptive and priority based schedule for concurrent execution.	Yes	n/a
S4R_FDF_686	The Framework shall manage redundant execution of partitions and/or processes on different devices.	Yes	n/a
S4R_FDF_692	The Framework shall provide a mechanism for service discovery and announcement.	Yes	n/a
S4R_FDF_687	The Framework shall support configurable recovery actions in case of partition or process deviations from normal behaviour.	Yes	n/a
S4R_FDF_688	The Framework shall provide internal variables as outputs and the 'leader" shall update those outputs after each redundant execution of partitions or processes. The internal variables are persistent over more than a single execution of the partition or process.	n/a	n/a
S4R_FDF_693	The Framework shall provide internal variables as the input to synchronise the internal variables of a "follower" with the variables provided by the "leader" before each execution of partitions or processes. The internal variables are persistent over more than a single execution of the partition or process.	n/a	n/a
S4R_FDF_694	The Framework shall interface with the Monitoring Manager in a secure way, by offering authentication measures for instance, to provide the availability of forcing variables.	n/a	Yes
S4R_FDF_695	The Framework shall be able to suspend the execution of processes and/or partitions during a given time.	n/a	n/a
S4R_FDF_741	The Framework shall load the configuration file during inauguration.	Yes	n/a
S4R_FDF_755	The Framework shall guarantee calls to service functions with the same SIL assigned to the application functions using services.	Yes	n/a
S4R_FDF_783	The Framework shall guarantee spatial separation between memory spaces containing read-only and read-write variables, variables with different SIL, variables used by multiple independent instances, software code and parameters of the application function.	Yes	n/a
S4R_FDF_185	**2.1.1.2 Process management** This subsection defines a process and describes how the IMP (Integrated Modular Platform) interacts with each of these. The ECP (Extended Capabilities Port) shall offer services to create and manage timers for sequential execution and semaphores for sequential and concurrent execution.		
S4R_FDF_506	A process shall be in the state: • Suspended: The process is not permitted to be activated. • Waiting: The process is waiting for its activation, which depending on the triggering paradigm will be when the corresponding event is launched or it is a certain instant of time. • Ready: The process is ready to execute and will do it if it has the highest priority among the ready processes. • Running: The process is executing in the processor.	n/a	n/a
S4R_FDF_193	The Framework shall activate a time-triggered process in waiting state if:	n/a	n/a

Id	Text	Safety related	Security related
	• The current time is inside its partition time slot • The current time is a multiple of its period		
S4R_FDF_498	The Framework shall grant spatial separation among processes.	n/a	n/a
S4R_FDF_194	The Framework shall execute the process in ready state with the highest priority.	n/a	Yes
S4R_FDF_195	The Framework shall set the state of a process to waiting when its execution finishes.	n/a	n/a
S4R_FDF_196	The Framework shall launch the finishing event of a process when its execution finishes.	n/a	n/a
S4R_FDF_197	The Framework shall set a process in ready state to waiting if it waits for an event.	n/a	n/a
S4R_FDF_610	The processes shall be configured according to a configuration file.	n/a	n/a
S4R_FDF_497	The Framework shall execute processes sequentially or concurrently.	n/a	n/a
S4R_FDF_698	The Framework shall limit the execution time for each process.	n/a	n/a
S4R_FDF_519	The Framework shall control the execution of processes with the same SIL assigned to the involved application functions.	Yes	n/a
S4R_FDF_587	The Framework shall set-up separate process to execute each instance.	n/a	n/a
S4R_FDF_520	The Framework shall implement service functions whose response times allow the real-time execution of processes and implement mechanisms to ensure that execution.	Yes	n/a
S4R_FDF_521	The Framework shall monitor execution of processes concerning defined time-bounds for communication and processing.	Yes	n/a
S4R_FDF_604	The Framework shall support configurable recovery actions in case of a process deviates from normal behaviour.	n/a	n/a
S4R_FDF_522	The Framework shall notify a fault condition in case of error in the process execution.	Yes	n/a
S4R_FDF_523	A process can belong to different process schedules.	n/a	n/a
S4R_FDF_700	The Framework shall allow to processes to set and get the current FDF's operation mode.	n/a	n/a
S4R_FDF_199	**2.1.1.3 Partition management** Partitions give means to guarantee memory space separation, which might contain all the information of processes. Besides, a cyclic executive scheduler must give and take away access to the processor when it corresponds.		
S4R_FDF_205	A partition shall be active or inactive.	n/a	n/a
S4R_FDF_206	Only active partitions shall be executed.	n/a	n/a
S4R_FDF_208	The Framework shall guarantee temporal separation among partitions by ensuring that a process within a given time budget cannot be affected by the actions of any other tasks of any other partition.	Yes	n/a

Id	Text	Safety related	Security related
S4R_FDF_592	The Framework shall bind the period, and execution time for each partition.	Yes	n/a
S4R_FDF_685	The Framework shall ensure the independence (time, space) of services and to support partitions' independence.	Yes	n/a
S4R_FDF_759	The Framework shall manage interrupts through the O.S, to avoid any disturbance to time partitioning.	Yes	n/a
S4R_FDF_606	The Framework shall support synchronised execution of partitions among different processor cores and devices.	Yes	n/a
S4R_FDF_607	The Framework shall write/update the inputs of each partition before executing them.	n/a	n/a
S4R_FDF_609	The Framework shall write/update the outputs of each partition after executing them.	n/a	n/a
S4R_FDF_689	The Framework shall execute and write/update the outputs, when the partition has the redundancy role "leader".	n/a	n/a
S4R_FDF_690	The Framework shall execute the partition, but shall not write/update its outputs, when the partition has the redundancy role "follower".	n/a	n/a
S4R_FDF_691	The Framework shall activate one of the "follower" partitions in the case that the "leader" partition fails. The "follower" shall write the outputs of "leader" partition.	n/a	n/a
S4R_FDF_524	Partitions shall guarantee spatial separation to ensure that no process in one partition can modify without authorisation software code or application data of another partition. E.g., by means of memory protection mechanisms.	Yes	n/a
S4R_FDF_525	Partitions are configured according to the configuration file of the application functions to be executed.	Yes	n/a
S4R_FDF_527	A partition can belong to different partition schedules.	n/a	n/a
S4R_FDF_526	Partitions have assigned computational resources defined in configuration file. No resource is shared by partitions hosting application functions with different SIL.	Yes	n/a
S4R_FDF_210	Partitions shall contain one or more processes.	n/a	n/a
S4R_FDF_507	A partition shall be in the state: • Suspended: The partition is not permitted to be activated. • Waiting: The partition is waiting for its activation, which depending on the triggering paradigm will be when the corresponding event is launched or it is a certain instant of time. • Ready: The partition is ready to execute and will do it if it has the highest priority among the ready partitions. • Running: The partition is executing in the processor. • Isolated: The partition is isolated and it is not permitted to be activated.	n/a	n/a
S4R_FDF_602	The Framework shall not execute partitions in state suspended or isolated.	n/a	n/a
S4R_FDF_603	The Framework shall support configurable recovery actions in case of a partition deviates from normal behaviour.	n/a	n/a
S4R_FDF_528	Partitions shall notify fault conditions in case of invalid operation in the partition attempt (fatal fault).	Yes	n/a

Id	Text	Safety related	Security related
S4R_FDF_784	The Framework shall assign privileges for read-write access to a memory space only to independent application functions with at least the same SIL. Read-only access could be assigned to remaining application functions, if data alteration during reading can be excluded.	Yes	n/a
S4R_FDF_215	**2.1.1.4 Concurrency management** This subsection gives details regarding concurrency control and synchronisation techniques.		
S4R_FDF_216	An event shall be active or inactive.	n/a	n/a
S4R_FDF_217	The Framework shall activate an event when it is commanded to launch.	n/a	n/a
S4R_FDF_590	The Framework shall support concurrent execution of more than one partition in different processor cores and/or devices.	n/a	n/a
S4R_FDF_529	Concurrent accesses to shared resources shall be synchronised using semaphores and/or mutexes.	n/a	n/a
S4R_FDF_530	Concurrent executions shall be synchronised using semaphores and/or mutexes.	Yes	n/a
S4R_FDF_612	***2.1.2 Configuration management*** This subsection defines the requirements regarding the configuration the Functional Distribution Framework including settings for partitions and variables.		
S4R_FDF_613	The Framework shall statically identify an ECU instance at boot time (e.g., by local digital inputs)	n/a	n/a
S4R_FDF_614	The Framework shall dynamically acquire the identification of ECUs instances at boot time (e.g., by DHCP).	n/a	n/a
S4R_FDF_625	The Framework shall obtain the identifier of an ECU instance.	n/a	n/a
S4R_FDF_742	The configuration and re-configuration of the Framework shall involve all the application functions.	Yes	n/a
S4R_FDF_615	The Framework shall acquire the following configuration parameters of the FDF and make them available to the FDF's components with the same SIL assigned to related application functions. • Version information • User identification and privileges • Contained devices • Contained partitions • Scheduling parameter of contained partitions • Contained communication networks	Yes	n/a
S4R_FDF_616	The Framework shall acquire the following configuration parameters of a device and make them available according to the SIL assigned to related application functions. • Contained I/O units.	Yes	n/a

Id	Text	Safety related	Security related
S4R_FDF_618	The Framework shall acquire the consist network configuration of a given SIL and make it available to the FDF components with the same SIL.	Yes	n/a
S4R_FDF_619	The Framework shall acquire the configuration parameters of partitions and make them available to the FDF components. • Unique identifier • Version information • Execution period • Maximum execution time • Redundancy role • In- and Output variables • Contained processes • Scheduling policy and dependencies of the contained processes • Error handling including recovery actions	Yes	n/a
S4R_FDF_620	The Framework shall acquire the following configuration parameter set for a process FDF and make them available to the FDF components. • Unique identifier • Executable that is executed in the process • Mapping of input/output variables to variables provided by or send to other processes, network or I/O • Assigning rights for publishing/reading variables to the SW components. • Execution period • Maximum execution time • Redundancy role • Scheduling priority • Error handling including recovery actions • Access to FDF services (e.g. set global time)	Yes	n/a
S4R_FDF_621	The Framework shall acquire the following configuration parameter set for an executable and make them available to the FDF components. • Unique identifier • Version information • In- and Output variables • Variables available for external monitoring	n/a	n/a

Id	Text	Safety related	Security related
	• Variables stored persistently • Provided and required services		
S4R_FDF_622	The Framework shall acquire the following configuration parameter set for an IO unit and make them available to the FDF components. • Unique identifier • In- and Output variables • Decoder configuration for encoder signals • Update cycle of in- and output variables	Yes	n/a
S4R_FDF_623	The Framework shall acquire the following configuration parameter set of a variable and make them available to the FDF components. • Unique identifier • Value interpretation • Default value • Data type	Yes	n/a
S4R_FDF_624	The Framework shall acquire the configuration parameter set of a service and make them available to the FDF components. • Unique identifier	n/a	n/a
S4R_FDF_626	The Framework shall acquire the following configuration parameter set for the event log and make them available to the FDF components. • maximum size • time period for storage of reoccurring events	n/a	n/a
S4R_FDF_167	***2.1.3 Communication management*** This subsection contains requirements related to communication management.		
S4R_FDF_220	**2.1.3.1 Data and event distribution** This chapter contains requirements on event and ECU and application data distribution which is done by the use of distribution variables between processes.		
S4R_FDF_221	The Framework shall provide services to create exchange variables, which are data structured consisting of a set of parameter and value pairs and should be SIL independent.	Yes	n/a
S4R_FDF_222	The variables shall be exchanged between software components using the publish-subscribe pattern.	Yes	n/a

Id	Text	Safety related	Security related
	a) Communication is black channel (including the publish-subscribe pattern) b) Safety relevant process data must be encrypted c) Non-Safe process data may be encrypted d) Encryption credentials must be configured e) Public/private key encryption is not sufficient - there must be certificates exchanged to prevent 3rd party access to safety critical functions handled in 2.5 Security requirements Note: The publish–subscribe is a messaging pattern where senders of messages (publishers) do not directly send messages to specific receivers (subscribers) but instead characterise published messages into classes (e.g. certain variables) without knowledge of which subscribers, if any, there may be. Similarly, subscribers express interest in one or more classes and only receive messages that are of interest, without knowledge of which publishers, if any, there are.		
S4R_FDF_223	The Framework shall give software components read and write access only to the variables they are allowed to publish.	Yes	n/a
S4R_FDF_224	The Framework shall give software components read access only to the variables they are subscribed to (without altering their value).	Yes	n/a
S4R_FDF_753	The Framework shall give software components write access according to specification set during configuration.	Yes	n/a
S4R_FDF_225	The Framework shall guarantee that the software component publishing a variable is able to update its value.	Yes	n/a
S4R_FDF_226	The Framework shall guarantee that an updated value is accessible for every software component that is subscribed to it within the defined timely bound.	Yes	n/a
S4R_FDF_735	The Framework shall guarantee the updating of input variables according to the values of input channels and SIL level.	Yes	n/a
S4R_FDF_227	The Framework shall guarantee that the communicating software components may exchange messages in the same way, regardless of the location of the software components, be it: • in the same process • in different processes of the same partition • in different partitions of the same ECU • or in different ECUs of the same network Especially in the case of different ECUs on the same network, security aspects shall be considered. (handled in 2.5 Security requirements)	n/a	n/a
S4R_FDF_493	The Framework shall provide services to exchange Message data (non-cyclic/best-effort) using a "notification", "call/reply" or "call/reply/confirm" pattern.	n/a	n/a
S4R_FDF_494	The Framework shall provide services to request data of variables non-cyclic/non-deterministic.	n/a	n/a
S4R_FDF_495	The Framework shall provide services to read out the TTDB (Train Topology Database) which is the result of inauguration.	n/a	n/a
S4R_FDF_541	The Framework shall provide the ability to set default values to variables:	n/a	n/a

Id	Text	Safety related	Security related
	- of the train and consist network and - shared between partitions of the same device - Shared between processes of the same partition according to configuration.		
S4R_FDF_496	The Framework shall provide services to supervise the validity of the inauguration result.	Yes	n/a
S4R_FDF_508	The Framework shall provide services to support different redundancy concepts.	Yes	n/a
S4R_FDF_709	The Framework shall be able to replicate the value of local input variables on the consist network according to configuration.	n/a	n/a
S4R_FDF_509	The Framework shall provide services to define a variable which can be then updated from different redundant devices.	Yes	n/a
S4R_FDF_510	The Framework shall provide services to define a set of redundant variables which are each updated by the corresponding redundant device.	Yes	n/a
S4R_FDF_511	The Framework shall mark the variables as valid or invalid according to the chosen redundancy concept. (E.g. one out of two, two out of three...)	Yes	n/a
S4R_FDF_543	The Framework shall provide the ability to create and manage access to shared memories to facilitate communication between processes of the same partition.	Yes	n/a
S4R_FDF_780	The Framework shall guarantee the validity of safety-related data exchange between remote functions through messages composing and decomposing into variables out with the same SIL assigned to the application functions.	Yes	n/a
S4R_FDF_781	The Framework shall allow message function to access to memory spaces containing messages and variables with the same SIL.	Yes	n/a
S4R_FDF_785	The Framework shall guarantee the read-write access to memory spaces (according to the assigned privileges) with the same SIL assigned to the Application function(s) and variables stored.	Yes	n/a
S4R_FDF_786	The Framework shall execute an Application function, giving access to memory resources, only when required by its scheduling plan (and take away access otherwise).	Yes	n/a
S4R_FDF_228	**2.1.3.2 Networking** Networking comprises requirements on location transparency, whether a publish-subscribe pattern is used and the number of participants or the support of deterministic real-time and best-effort messages.	n/a	n/a
S4R_FDF_229	The Framework shall provide communication mechanisms that are abstracted of the physical realisation of the communication hardware.	n/a	n/a
S4R_FDF_230	The Framework shall provide a standardised software interface for communication between software components ensuring their communication independent whether they are located - on the same ECU on the same core	n/a	n/a

Id	Text	Safety related	Security related
	- on the same ECU on another core - on the same ECU on another microcontroller on another ECU		
S4R_FDF_711	The Framework shall provide an IEC 61375-2-3 compliant safety layer for the consist network communication	Yes	n/a
S4R_FDF_231	The Framework shall provide a communication service that allows it to send messages (containing variables) to other components on the network within defined timely bounds from the point in time where the application sends the message to the point in time it is sent on the network (deterministic sending).	Yes	n/a
S4R_FDF_756	The Framework shall instantiate messages according to the configuration file, including: - Unique identifier (ID) - Messages to be received or send - List of variables linked to messages - Messages schedule - Deadline	Yes	n/a
S4R_FDF_750	The Framework shall periodically send messages within defined time bounds and receive them within defined maximum delay (deterministic sending).	Yes	n/a
S4R_FDF_512	The Framework shall provide a communication service which provides a deterministic way for an application to announce/prepare a message/data value for deterministic sending.	Yes	n/a
S4R_FDF_232	The Framework shall provide a communication service that makes received messages from other components on the network available to the application within defined timely bounds (deterministic receiving).	Yes	n/a
S4R_FDF_513	The Framework shall provide a communication service which provides a deterministic way to fetch a message/data value after deterministic reception.	n/a	n/a
S4R_FDF_751	The Framework shall implement communication service without any operation on the messages' safety layer content.	Yes	n/a
S4R_FDF_233	**2.1.3.3 System integration** This chapter contains requirements regarding the COM layer, the inauguration process or transport layer protocols among others. System Integration Requirements have been covered in detail in WP1.	n/a	n/a
S4R_FDF_168	***2.1.4 Time management*** Different ECUs share a unique global time that is synchronised with UTC. These requirements contain details regarding interfaces used, protocols and ways of synchronisation, i.e., automatic or manual.	n/a	n/a
S4R_FDF_235	The Framework shall provide a service for starting application processes based on the progression of time.	Yes	n/a

Id	Text	Safety related	Security related
S4R_FDF_236	The Framework shall synchronise the local computer clock with the external global clock source and keep it synchronised with a maximum deviation of the global clock source of 1 microsecond.	Yes	n/a
S4R_FDF_762	The Framework shall synchronize the local clock independently from the execution of different partition's processes.	Yes	n/a
S4R_FDF_237	The Framework shall allow process and partition execution to be scheduled at a configured time instant within a configured rate-monotonic execution cycle period.	n/a	n/a
S4R_FDF_238	The Framework shall check and inform about successful synchronisation, synchronisation state and synchronisation errors.	Yes	n/a
S4R_FDF_544	The Framework shall allow processes to set the global time if allowed by configuration to do so.	n/a	n/a
S4R_FDF_545	The Framework shall provide the ability to processes to create, configure and delete timers.	n/a	n/a
S4R_FDF_239	The global time shall be made available to all ECUs through the network layer.	n/a	n/a
S4R_FDF_240	Global time dissemination shall be fault tolerant. Note: In case no time synchronisation is available, there is no scheduled (critical) communication possible. In case of erroneous time synchronisation, messages may arrive early or late and can lead to catastrophic events. This erroneous time synchronization must be detected by the SDT layer.	Yes	n/a
S4R_FDF_736	The Framework shall not finalize the inauguration without a valid global-time.	Yes	n/a
S4R_FDF_169	***2.1.5 Input/output management***		
S4R_FDF_261	**2.1.5.1 Input management** This subsection contains requirements specifying which Input devices the ECU must be able to work with and how the data of these devices should be read and interpreted.		
S4R_FDF_263	The Framework shall provide a service to create the controller access to an analog input.	n/a	n/a
S4R_FDF_264	The Framework shall provide a service to create the controller access to a digital input.	n/a	n/a
S4R_FDF_265	The inputs shall be accessible over configurable symbolic names.	Yes	n/a
S4R_FDF_764	The Framework shall allow input functions to access only to memory spaces with the same SIL.	Yes	n/a
S4R_FDF_266	The Framework shall create an exchange variable associated with each input channel.	Yes	n/a
S4R_FDF_546	The Framework shall set default values to digital and analog input variables according to configuration, with the same SIL assigned to related application functions.	Yes	n/a
S4R_FDF_267	The exchange variable associated with an input channel shall contain the acquired input channel value.	Yes	n/a
S4R_FDF_268	The Framework shall store the current value of every used input at the end of each acquisition cycle in the associated exchange variable.	Yes	n/a

Id	Text	Safety related	Security related
S4R_FDF_269	The Framework shall provide a service for reading the last valid value of every used input, stored in the associated exchange variable.	Yes	n/a
S4R_FDF_270	The service for reading the value of every used input stored in the associated exchange variable shall not be interruptible to ensure data consistency.	Yes	n/a
S4R_FDF_716	The Framework shall decode encoder signals and transfer the value into a variable, including validity information.	n/a	n/a
S4R_FDF_262	**2.1.5.2 Output management** Analogously, this other subsection contains requirements specifying which Output devices the ECU must be able to work with and how the data of these devices should be written and interpreted.		
S4R_FDF_271	The Framework shall provide a service to create the controller access to an analog output.	n/a	n/a
S4R_FDF_272	The Framework shall provide a service to create the controller access to a digital output.	n/a	n/a
S4R_FDF_273	The outputs shall be accessible over configurable symbolic names.	Yes	n/a
S4R_FDF_765	The Framework shall allow output functions to access only to memory spaces with the same SIL.	Yes	n/a
S4R_FDF_274	The Framework shall create an exchange variable associated with each output channel.	Yes	n/a
S4R_FDF_547	The Framework shall set digital and analog outputs to default values according to configuration, with the same SIL assigned to related application functions.	Yes	n/a
S4R_FDF_275	The exchange variable associated with an output channel shall contain the output channel set value.	Yes	n/a
S4R_FDF_276	The Framework shall provide a service for writing a new value and update it in the associated exchange variable of every used output.	Yes	n/a
S4R_FDF_277	The service for writing a new value in the associated exchange variable of every used output shall not be interruptible to assure data consistency.	Yes	n/a
S4R_FDF_548	***2.1.6 Health management***		
S4R_FDF_549	The Framework shall support CPU, board and/or rack temperature monitoring, if supported by the HW monitoring.	n/a	n/a
S4R_FDF_551	The Framework shall support checking if partitions are executed within their maximum execution time.	n/a	n/a
S4R_FDF_552	The Framework shall support a HW watchdog timer (WDT).	n/a	n/a
S4R_FDF_553	The Framework shall refresh the WDT.	n/a	n/a
S4R_FDF_554	The Framework shall support integrity checks of the HW.	n/a	n/a
S4R_FDF_555	The Framework shall support check if partitions and processes update their outputs according to the value of variables and	Yes	n/a

Id	Text	Safety related	Security related
	SIL level.		
S4R_FDF_557	The Framework shall log the errors detected in a log file.	n/a	n/a
S4R_FDF_728	The Framework shall check the timeliness and sequence of messages exchanged between remote functions.	Yes	n/a
S4R_FDF_556	The Framework shall provide reaction to errors when a partition or process: • does not write the output • does not terminate execution in time • CPU, board and/or rack temperature exceeds the allowed range • CPU, board and/or rack load too high	n/a	n/a
S4R_FDF_558	The Framework shall consider the following reaction to error mechanisms with the highest SIL assigned to the application functions, without disturbing to other framework's services: • restart the ECU of the affected partition/process (without affecting other ECUs) • restart the affected partition/process (without affecting other partitions/processes) • isolate/terminate the affected partition/process (without affecting other partitions/processes) • inform the application function and continue with normal operation	Yes	n/a
S4R_FDF_746	The Framework shall provide reaction to errors when a communication error is identified: • message authenticity • message integrity • message timeliness • message sequence	Yes	n/a
S4R_FDF_745	The Framework notifies to application function and reacts against safety-related communication errors, for example, discarding erroneous messages.	Yes	n/a
S4R_FDF_754	The Framework shall detect and notify the application SW in case of unavailability of scheduled services or in case of incorrect calls (different schedules).	Yes	n/a
S4R_FDF_758	The Framework shall notify fault conditions to all the application function(s) involved (with SIL) without disturbing to other framework's services and no later than the maximum time for safe state.	Yes	n/a
S4R_FDF_769	The Framework shall notify a fault condition to the related application function in case of inconsistencies between the values stored into an exchange variable and the status of the platform's input/output.	Yes	n/a
S4R_FDF_179	***2.1.7 Monitoring management***		
S4R_FDF_562	The Framework shall allow remotely requesting the list of available variables.	n/a	n/a

Id	Text	Safety related	Security related
S4R_FDF_563	The Framework shall allow remotely registering variables that can be monitored.	n/a	n/a
S4R_FDF_564	The Framework shall send the list of variable that can be monitored to external device.	n/a	n/a
S4R_FDF_355	The Framework shall allow remotely reading the variables of a component.	n/a	n/a
S4R_FDF_356	The Framework shall allow remotely writing the variables of a component.	n/a	n/a
S4R_FDF_357	The Framework shall allow remotely reading the events of a component.	n/a	n/a
S4R_FDF_358	The Framework shall allow remotely writing the events of a component.	n/a	n/a
S4R_FDF_359	The Framework shall allow remotely forcing the variables of a component.	n/a	n/a
S4R_FDF_361	The Framework shall allow remotely unforcing the variables of a component.	n/a	n/a
S4R_FDF_362	The Framework shall allow remotely forcing the events of a component.	n/a	n/a
S4R_FDF_363	The Framework shall allow remotely unforcing the events of a component.	n/a	n/a
S4R_FDF_364	The Framework shall check the state of all existing processes.	n/a	n/a
S4R_FDF_365	The Framework shall check the value of all framework variables, comparing them with the I/O values.	Yes	n/a
S4R_FDF_704	The Framework shall guarantee a secure communication with external devices.	n/a	Yes
S4R_FDF_733	The Framework shall provide services to monitor variables (e.g., remotely (out of FDF)).	Yes	n/a
S4R_FDF_761	The Framework shall detect faults with the highest SIL assigned to the application functions to be executed, without disturbing to other framework's services.	Yes	n/a
S4R_FDF_738	The Framework shall detect resource-related faults at power-up and periodically.	Yes	n/a
S4R_FDF_743	The Framework shall detect incoherence of configuration file.	Yes	n/a
S4R_FDF_744	The Framework shall detect the lack of configuration file's integrity.	Yes	n/a
S4R_FDF_752	The Framework shall assign to the monitoring-function RO privileges to variables stored into memory spaces with lowest integrity level or to all the memory spaces with different integrity levels (SIL) without altering the execution of other services.	Yes	n/a
S4R_FDF_760	The Framework shall monitor the alignment with the external global clock with the highest SIL assigned to the application functions to be executed.	Yes	n/a
S4R_FDF_771	The Framework shall monitor that non-safety data uses different structures than ones used for safety-related data.	n/a	n/a
S4R_FDF_788	The Framework shall provide fault detection during run-time execution.	Yes	n/a
S4R_FDF_789	The Framework shall provide further measures and detection techniques, in addition to the techniques/measures provided, for run-time fault detection.	Yes	n/a

Id	Text	Safety related	Security related
S4R_FDF_377	***2.1.8 Log management*** This subsection describes which information the system log should include. This could be sensitive activity, errors or the state of the different processes.		
S4R_FDF_378	The Framework shall create a log file per day (if applicable persistent log file).	n/a	n/a
S4R_FDF_574	The Framework shall configure the maximum size of the event log.	n/a	n/a
S4R_FDF_575	The Framework shall overwrite previously recorded event if the maximum of the log file size is reached.	n/a	n/a
S4R_FDF_576	The Framework shall only record one error every certain period of time, in case of recurrent errors. The logging period of time shall be configurable.	n/a	n/a
S4R_FDF_380	The Framework shall log the minimum execution time of the processes per hour.	n/a	n/a
S4R_FDF_381	The Framework shall log the maximum execution time of the processes per hour.	n/a	n/a
S4R_FDF_382	The Framework shall log the average execution time of the processes per hour.	n/a	n/a
S4R_FDF_383	The Framework shall log if any of its processes does not meet its deadline.	n/a	n/a
S4R_FDF_384	The Framework shall log if the integrity of the memory space of a partition has an error.	Yes	n/a
S4R_FDF_385	The Framework shall log if the integrity of the configuration file of the Framework has an error.	n/a	n/a
S4R_FDF_386	The Framework shall log if the coherency of the configuration file of the Framework has an error.	n/a	n/a
S4R_FDF_387	The Framework shall log if any unexpected external access is detected.	n/a	n/a
S4R_FDF_388	The Framework shall log if any not allowed external access is detected.	n/a	n/a
S4R_FDF_379	The log file shall follow the “report_yyyymmdd_xxx.log” naming convention, where yyyy, mm and dd stand for the system year, month and day and the xxx represents an incremental value in case more than one file with the same date exists.	n/a	n/a
S4R_FDF_389	The Framework must make a back up of the log files every day.	n/a	n/a
S4R_FDF_390	The Framework shall include a timestamp for each entry of the log file.	n/a	n/a
S4R_FDF_580	The Framework shall provide the application with the ability to add an entry in the event log.	n/a	n/a
S4R_FDF_581	The Framework shall allow the application to use the following logging levels for an entry: a) Debug b) Info c) Warning d) Error e) Fatal	Yes	n/a

Id	Text	Safety related	Security related
S4R_FDF_582	The Framework shall provide the ability to export the current event log as a file with the following information per event log entry: • Identification of triggering entity • Type (logging level) • Event ID • Event message • Raw data	n/a	n/a
S4R_FDF_565	***2.1.9 Deployment management*** This subsection describes the requirements of the deployment management that enables to install and update configuration files and application executables of FDF partitions.		
S4R_FDF_571	The Framework shall implement a secure file transfer such as FTPS or SFTP transfer protocols.	n/a	Yes
S4R_FDF_666	The Framework shall support debug operation and maintenance operation modes.	n/a	n/a
S4R_FDF_770	The Framework shall support maintenance of non-safety data using different structures than ones used for safety-related data.	Yes	n/a
S4R_FDF_567	The Framework shall provide maintenance staff with the ability to install executables on partitions train network, remote and direct connections.	n/a	n/a
S4R_FDF_566	The Framework shall provide maintenance staff with the ability to update executables on partitions train network, remote and direct connections.	n/a	n/a
S4R_FDF_573	The Framework shall provide maintenance staff with the ability to uninstall executables on partitions through train network, remote and direct connections.	n/a	n/a
S4R_FDF_568	The Framework shall provide maintenance staff with the ability to install configuration files through train network, remote and direct connections.	n/a	n/a
S4R_FDF_569	The Framework shall provide maintenance staff with the ability to update configuration files train network, remote and direct connections.	n/a	n/a
S4R_FDF_570	The Framework shall provide maintenance staff with the ability to uninstall configuration files train network, remote and direct connections.	n/a	n/a
S4R_FDF_635	The Framework shall provide the maintenance staff with a secure way to install executables on a partition.	n/a	Yes
S4R_FDF_639	The Framework shall provide the maintenance staff with a secure way to update executables on a partition.	n/a	Yes
S4R_FDF_640	The Framework shall provide the maintenance staff with a secure way to uninstall executables on a partition.	n/a	Yes
S4R_FDF_660	The Framework shall provide the maintenance staff with a secure way to install configuration files.	n/a	Yes

Id	Text	Safety related	Security related
S4R_FDF_661	The Framework shall provide the maintenance staff with a secure way to update configuration files.	n/a	Yes
S4R_FDF_662	The Framework shall provide the maintenance staff with a secure way to uninstall configuration files	n/a	Yes
S4R_FDF_636	The Framework shall allow deleting persistently stored data and files with uninstalled executables.	n/a	n/a
S4R_FDF_658	The Framework shall provide detailed version information of FDF to maintenance staff.	n/a	n/a
S4R_FDF_663	The Framework shall provide detailed version information of each process (installed executable) to the maintenance staff.	n/a	n/a
S4R_FDF_665	The Framework shall provide detailed version information of each configuration file to the maintenance staff.	n/a	n/a
S4R_FDF_659	The Framework shall validate the executable code, schedule and the resource availability before the installation, during the installation and during updating it.	Yes	n/a
S4R_FDF_664	The Framework shall validate the configuration file before processing it or updating it to ensure that there is not conflict in the communication, schedule or resource availability of partitions and processes.	Yes	n/a
S4R_FDF_787	The Framework shall support concurrent re-configuration of partitions, guaranteeing that the re-configuration does not affect the remaining partitions. Those partitions may execute different and independent application functions with the same SIL level and to be hosted by one partition.	Yes	n/a
S4R_FDF_641	***2.1.10 File management*** This subsection writes and reads files and variables that persist over device switch on and switch off cycles.		
S4R_FDF_644	The Framework shall enable to create new files in memory.	n/a	n/a
S4R_FDF_645	The Framework shall allow opening existing files.	n/a	n/a
S4R_FDF_648	The Framework shall allow opening files in read-only (RO) or read/write (RW) modes.	n/a	n/a
S4R_FDF_649	The Framework shall allow writing data into a file.	n/a	n/a
S4R_FDF_650	The Framework shall allow reading data from a file.	n/a	n/a
S4R_FDF_651	The Framework shall allow storing files persist over device switch-on and switch-off cycles.	n/a	n/a
S4R_FDF_652	The Framework shall enable to remove files.	n/a	n/a
S4R_FDF_653	The Framework shall enable to persistently store variables over device switch-on and switch-off cycles.	n/a	n/a
S4R_FDF_654	The Framework shall allow loading variables which are persistently stored.	n/a	n/a
S4R_FDF_655	The Framework shall store variables in way that they can be accessed by a partition using a unique identifier. E.g., identify a value by a key.	n/a	n/a
S4R_FDF_656	The Framework shall guarantee that no variable or file corruption occurs if the device switches off while writing data to a	n/a	n/a

Id	Text	Safety related	Security related
	variable or a file.		
S4R_FDF_657	The Framework shall allow closing files.	n/a	n/a
S4R_FDF_171	**2.2 Non-functional requirements**		
S4R_FDF_172	***2.2.1 Performance requirements***		
S4R_FDF_299	The Framework shall guarantee methodology for performance analysis for considered system configurations.	n/a	n/a
S4R_FDF_300	The Framework shall guarantee methodology for system performance analysis in case of accidental situations.	n/a	n/a
S4R_FDF_301	The Framework shall define, configure, and assess performance of each node of system.	n/a	n/a
S4R_FDF_302	The Framework shall define, configure, and assess node performance for specified (cyber) security level.	n/a	n/a
S4R_FDF_303	The Framework shall define, configure, and assess node performance for I/O interface.	n/a	n/a
S4R_FDF_304	The Framework shall define, configure, and assess node performance for control algorithms and inter-partition communication.	n/a	n/a
S4R_FDF_305	The Framework shall define, configure, and assess node performance for logging and diagnostic subsystem.	n/a	n/a
S4R_FDF_306	The Framework shall define, configure, and assess node performance for communication interface.	n/a	n/a
S4R_FDF_307	The Framework shall define, configure, and assess performance of communication channels • channel priority • channel throughput	Yes	n/a
S4R_FDF_308	The Framework shall define, configure, and assess performance of communication channels for predefined parameters as: • jitter • latency • response time	Yes	n/a
S4R_FDF_309	The Framework shall define, configure, and assess performance for protection communication channels against cyber attack.	n/a	n/a
S4R_FDF_310	The Framework shall define, configure, and assess “performance for future use”: • data communication – capacity, throughput, security • control algorithms	n/a	n/a

Id	Text	Safety related	Security related
	• fault tolerance		
S4R_FDF_173	***2.2.2 Validation and verification support*** The requirements in this subsection include all information regarding techniques used for testing purpose.		
S4R_FDF_630	The Framework shall validate the installation or update of executable code before processing it. The scheduling and resources attached to other partitions shall not be affected.	Yes	n/a
S4R_FDF_631	The Framework shall validate the installation or update of a configuration file before processing it. The communication, scheduling and resources of partitions and processes shall not be affected.	Yes	n/a
S4R_FDF_314	The Framework shall provide services to control and monitor the application execution (start, stop, synchronising to external trigger). I.e., using program flow monitoring techniques.	Yes	n/a
S4R_FDF_316	The Framework shall prevent the access to any validation and verification support service (fault injection and monitoring, forcing of outputs, monitoring of inputs and outputs, application control and monitoring, logging/tracing) on power up. The framework shall enable the validation and verification support services only on explicit request.	n/a	n/a
S4R_FDF_315	The Framework shall provide logging/tracing services for a selectable set of events related to • Fault injection and monitoring • Communication and shared network memory change • Output change • Input change • Application execution and monitoring	n/a	n/a
S4R_FDF_311	The Framework shall provide services to inject faults and monitor the fault reaction related to • non-critical (SIL0) • platform partitioning and isolation mechanism • communication (transmission, reception) and shared network memory • output control • input monitoring • application execution (timing, memory access, start, stop, throttling, …)	n/a	n/a
S4R_FDF_312	The Framework shall provide services to force the outputs to all states (valid and invalid) independent of the current control by the associated application.	n/a	n/a
S4R_FDF_313	The Framework shall provide services to monitor the state of all outputs and inputs independently from the application that is associated to the respective inputs/outputs.	n/a	n/a
S4R_FDF_174	**2.3 Interface requirements**		
S4R_FDF_701	The Framework shall offer an interface to allow registering a variable that can be monitored externally.	n/a	n/a

Id	Text	Safety related	Security related
S4R_FDF_702	The Framework shall offer an interface to allow external devices to request the list of variables which can be monitored.	n/a	n/a
S4R_FDF_703	The Framework shall offer an interface to allow external devices to request monitoring a number of variables with a given frequency.	n/a	n/a
S4R_FDF_706	The Framework shall provide an interface between input and output variables of processes executed in partitions - on the same device - on different devices in the same consist or - on devices in different consists of the same train according to their defined inputs and outputs.	n/a	n/a
S4R_FDF_707	The Framework shall provide an interface between variables provided by I/O devices to inputs of processes executed in partitions - on the same device - on another device in the same consist or - in another consist of the same train according to the input definition of the partitions.	n/a	n/a
S4R_FDF_708	The Framework shall provide an interface between variables provided by a process executed on a partition to variables controlling outputs of I/O devices located - on the same device - on another device in the same consist or - in another consist of the same train according to the interface definition between the partition and the I/O device.	n/a	n/a
S4R_FDF_712	The Framework shall offer an interface to external devices to force variables.	n/a	n/a
S4R_FDF_713	The Framework shall offer an interface to register variable that can be forced.	n/a	n/a
S4R_FDF_734	The Framework shall guarantee the independence of I/O interfaces that can be requested by the application function.	Yes	n/a
S4R_FDF_175	***2.3.1 Application*** The requirements in this section describe the interface requirements between applications and the framework.		
S4R_FDF_318	The Framework shall offer an interface to create time-triggered processes.	n/a	n/a
S4R_FDF_320	The Framework shall offer an interface to set the priority of a process.	n/a	n/a
S4R_FDF_321	The Framework shall offer an interface to set the deadline of a process.	n/a	n/a
S4R_FDF_322	The Framework shall offer an interface to set the period of a time-triggered process.	n/a	n/a
S4R_FDF_323	The Framework shall offer an interface to set the offset of a time-triggered process.	n/a	n/a
S4R_FDF_324	The Framework shall offer an interface to set the activation events of an event-triggered process.	n/a	n/a

Id	Text	Safety related	Security related
S4R_FDF_325	The Framework shall offer an interface to create periodic timers.	n/a	n/a
S4R_FDF_326	The Framework shall offer an interface to create sporadic timers.	n/a	n/a
S4R_FDF_327	The Framework shall offer an interface to set the deadline of a timer.	n/a	n/a
S4R_FDF_328	The Framework shall offer an interface to start a timer.	n/a	n/a
S4R_FDF_329	The Framework shall offer an interface to stop a timer.	n/a	n/a
S4R_FDF_330	The Framework shall offer an interface to create partitions.	n/a	n/a
S4R_FDF_331	The Framework shall offer an interface to set the offset of a partition.	n/a	n/a
S4R_FDF_332	The Framework shall offer an interface to set the period of a partition.	n/a	n/a
S4R_FDF_333	The Framework shall offer an interface to set the budget of a partition.	n/a	n/a
S4R_FDF_334	The Framework shall offer an interface to set the processes of a partition.	n/a	n/a
S4R_FDF_335	The Framework shall offer an interface to create events.	n/a	n/a
S4R_FDF_336	The Framework shall offer an interface to launch an event.	n/a	n/a
S4R_FDF_337	The Framework shall offer an interface to discover, monitor and control the applications it executes.	n/a	n/a
S4R_FDF_501	The Framework shall offer an interface to read static configuration from a file.	n/a	n/a
S4R_FDF_176	***2.3.2 I/O*** The requirements in this section describe the inputs and outputs of the Framework.		
S4R_FDF_338	The Framework shall offer an interface to read the type and number of input and output ports.	n/a	n/a
S4R_FDF_339	The Framework shall offer an interface to read analog inputs.	n/a	n/a
S4R_FDF_340	The Framework shall offer an interface to read digital inputs.	n/a	n/a
S4R_FDF_341	The Framework shall offer an interface to write analog outputs.	n/a	n/a
S4R_FDF_342	The Framework shall offer an interface to write digital outputs.	n/a	n/a
S4R_FDF_343	The Framework shall offer an interface to map a variable to each analog or digital input or output.	n/a	n/a
S4R_FDF_344	The Framework shall offer an interface to determine the type, size and optional scaling/units of variables mapped to analog inputs and outputs.	n/a	n/a
S4R_FDF_345	The Framework shall offer an interface to determine the type, size and bit usage of variables mapped to digital inputs and	n/a	n/a

Id	Text	Safety related	Security related
	outputs.		
S4R_FDF_346	The Framework shall offer an interface to set the update cycle (multiple of basic cycle) for each mapped variable.	n/a	n/a
S4R_FDF_347	The Framework shall be able to map digital or analog input or output ports to data types complying with IEC 61375-2-1 [7] and IEC 61375-2-3 [2].	n/a	n/a
S4R_FDF_779	The Framework shall support at least 14 analog inputs with 12 bit resolution, 1 digital output and 7 digital outputs. If the controller does not support such capabilities, alternative peripherals shall be provided (e.g., SPI).	n/a	n/a
S4R_FDF_177	***2.3.3 Network*** Network interfacing to COM/Middleware		
S4R_FDF_348	For outgoing messages to the network, the network interface device shall read the message data from the partition message memory.	n/a	n/a
S4R_FDF_489	Application shall place message data into the partition message memory which is per configuration aligned with queuing or sampling ports.	n/a	n/a
S4R_FDF_349	For incoming messages from the network, the network interface device shall write the message data to the partition message memory.	n/a	n/a
S4R_FDF_490	Application shall read message data from the partition message memory which is per configuration aligned with queuing or sampling ports.	n/a	n/a
S4R_FDF_350	The configuration of the Framework and the Network shall specify for each port whether it is operated as a queuing or sampling port.	n/a	n/a
S4R_FDF_351	The configuration of the Framework (software abstraction / COM / middleware layer) shall define which data is stored into the message and at what point in time the message is published to the network.	n/a	n/a
S4R_FDF_352	The configuration of the Framework and the Network shall be consistent with regards to which frames are sent and received, at which times.	n/a	n/a
S4R_FDF_353	The Framework shall be able to receive status and errors related to message transmission in the network interface.	n/a	n/a
S4R_FDF_178	**2.4 Safety requirements**		

Id	Text	Safety related	Security related
S4R_FDF_180	***2.4.1 EC directive***		
S4R_FDF_391	**EC Train Directive** Annex III of DIRECTIVE (EU) 2016/797 [3] on the interoperability of the rail system within the European Union. Relevant chapters of Annex III of the directive: • 1.1.1 General requirements/Safety • 1.5 General requirements/Technical compatibility • 2.3.1 Control-command and signalling/Safety • 2.4.1 Rolling stock/Safety • 2.4.2 Rolling stock/Reliability and availability • 2.4.3 Rolling stock/Technical compatibility		
S4R_FDF_392	**2.4.1.2 TSI LOC&PAS** 1302/2014/CE - COMMISSION REGULATION (EU) No 1302/2014 of 18 November 2014 [4]. Relevant chapters: • 4.2.4.2.1. (3), (4) Functional requirements • 4.2.4.2.1. (11) Functional requirements • 4.2.4.3 (1)/(2) Type of brake system • 4.2.4.10. (3) Brake requirements for rescue purposes • 4.2.5.2. (2), (3) Audible communication system • 4.2.5.3.1 (2) Passenger alarm/General		
S4R_FDF_643	**2.5 Security requirements** This subsection defines the security-related requirements of FDF.		
S4R_FDF_414	The framework shall secure the incoming/outgoing communication (channel) to the ECUs (Electronic Control Units) against security threats with regards to confidentiality, authenticity, integrity and availability whilst respecting real-time constraints (i.e. predictable latency and low jitter).	Yes	Yes
S4R_FDF_416	The framework shall protect stored data against adversaries (with regards to confidentiality, authenticity and data integrity).	n/a	Yes
S4R_FDF_417	The framework shall include a mechanism in order to prevent unknown/unexpected traffic (i.e. admission and access control).	Yes	Yes

Id	Text	Safety related	Security related
S4R_FDF_420	The framework shall accomplish the need of protecting the data and state of the functions during execution on an ECU.	n/a	Yes
S4R_FDF_667	The Framework shall support cryptography algorithms, key sizes and mechanisms to key establishment and management according to common security industry practises and recommendations.	n/a	Yes
S4R_FDF_412	The framework shall provide cryptographic mechanisms and handle cryptographic objects • Ensure framework's security as well as framework's communication channel (receiving and transmitting role) by means of secure cryptographic algorithms • Management of cryptographic keys (creation, deletion and retention) • Calculation of cryptographic functions (digital signatures, MACs, encryption/decryption)	n/a	Yes
S4R_FDF_646	The Framework shall support data encryption.	n/a	Yes
S4R_FDF_647	The Framework shall support data decryption.	n/a	Yes
S4R_FDF_409	The framework shall operate accordingly/with regards to confidentiality • Ensure that data inside the framework cannot be read by an unauthorised entity: ensure non-disclosure of information/data towards entities (i.e. users, processes, and device) unless a successful access authorisation.	n/a	Yes
S4R_FDF_410	The framework shall operate accordingly/with regards to authenticity • Assurance of entities' identity • Ensure/verify data source: information/data comes from a verified and trusted entity (sender) • Information collected by the framework should be authentic with respect to origin and time if the framework performs actions based on that information • The author of the message, respectively the origin sending entity of the information/data, shall be evident and traceable at any time (with regards to non-repudiation)	n/a	Yes
S4R_FDF_415	The Framework shall support availability of access control in the network to ensure robustness to DoS attacks as well as side-channel attacks.	n/a	Yes
S4R_FDF_429	The framework shall ensure that security policy enforcement functions and the data that configures them cannot be modified without authorisation.	n/a	Yes
S4R_FDF_418	The framework shall support secure storage for key(s) and trust anchor(s) for secure authentication and communication (with regards to security services and authenticity).	n/a	Yes
S4R_FDF_419	The framework shall operate with authenticated entities (ECUs, SW/HW components) only (with regards to authenticity) • The framework shall enforce authenticity and integrity of the ECUs in order to meet/fulfil framework's security requirements. • The framework shall enforce authenticity and integrity of the software components in order to meet/fulfil framework's security requirements.	n/a	Yes

Id	Text	Safety related	Security related
S4R_FDF_669	The Framework shall allow to assign privileges to authenticated users (access rights).	n/a	Yes
S4R_FDF_670	The Framework shall support executable identification and authentication.	n/a	Yes
S4R_FDF_671	The Framework shall allow to assign privileges to authenticated executables (access rights).	n/a	Yes
S4R_FDF_672	The Framework shall: • initialise authenticator content • change all default authenticators upon control system installation • change/refresh all authenticators • protect all authenticators from unauthorised disclosure and modification when stored and transmitted.	n/a	Yes
S4R_FDF_673	The Framework shall support the management of identifiers by users, groups, roles or control system interfaces.	n/a	Yes
S4R_FDF_749	The component "Security Management" shall be able to support the management of all accounts by authorized users, including adding, activating, modifying, disabling and removing accounts.	n/a	Yes
S4R_FDF_674	The Framework shall enforce configurable password strength based on minimum length and variety of character types.	n/a	Yes
S4R_FDF_413	The framework shall provide a Public Key Infrastructure (PKI) • Support/ensure the authentication process of entities (with regards to authenticity) • Management of certificates (retention and update)	Yes	Yes
S4R_FDF_676	The Framework shall validate certificates by: • checking the signature of given certificates • constructing a certification path to an accepted CA • deploying leaf certificates to all hosts which communicate with the subject to which the certificate is issued (in the case of self signed certificates) • checking the certificate's revocation.	n/a	Yes
S4R_FDF_677	The Framework shall: • establish user (human, SW process, device) control of the private keys • map the authenticated identity to a user (human, SW process, device).	n/a	Yes
S4R_FDF_678	The Framework shall be able to obscure feedback authentication information during authentication process.	n/a	Yes
S4R_FDF_679	The Framework shall enforce a limit of configurable number of consecutive invalid access attempts by any user (human, SW, device) during a configurable time period.	n/a	Yes
S4R_FDF_680	The Framework shall deny access for specified period of time or until unlocked by an administrator when the access attempts number is exceeded.	n/a	Yes

Id	Text	Safety related	Security related
S4R_FDF_681	The Framework shall display a system notification message before authenticating. This message shall only be configurable by authorised users.	n/a	Yes
S4R_FDF_430	The Framework shall provide the capability to detect, generate and export audit records for security relevant auditable events.	n/a	Yes
S4R_FDF_730	The Framework shall periodically verify the correct operation of security protection functions and notify system administrator when anomalies are discovered.	n/a	Yes
S4R_FDF_411	The Framework shall operate accordingly/with regards to data integrity • Support/offer mechanism(s) in order to ensure data integrity for information collected within the framework. • Ensure that the information has/have not been modified either in transit or in storage on the route from the sender's entity to the receiver's entity.	n/a	Yes
S4R_FDF_421	The framework shall accomplish the need of protecting the data and state of the functions during execution within software components.	n/a	Yes
S4R_FDF_422	The framework shall ensure the data isolation between different partitions created and maintained by the framework so that the data in a partition is accessible only by code running in that partition (SIL).	Yes	Yes
S4R_FDF_423	The framework shall ensure the isolation of the resource between different partitions created and maintained by the framework so that the resources exported by the framework into a partition are accessible only by code running in that partition (with SIL).	Yes	Yes
S4R_FDF_424	The framework shall provide information flow control that enforces strict partition isolation so that only explicitly configured interaction are allowed.	n/a	Yes
S4R_FDF_425	The framework shall ensure that a failure in one partition is not propagated to other partitions.	Yes	Yes
S4R_FDF_426	The framework shall ensure that an attack affecting one partition is not propagated to other partitions.	Yes	Yes
S4R_FDF_427	The framework shall ensure that security policy enforcement functions cannot be bypassed.	n/a	Yes
S4R_FDF_428	The framework shall ensure that security policy enforcement functions are always invoked.	n/a	Yes
S4R_FDF_731	The Framework or its support utilities shall provide user functionality to facilitate creation of backups of user-level and system-level information (including system security state information).	n/a	Yes
S4R_FDF_732	The Framework shall provide user functionality to allow be recovering and reconstituting to previously saved Backup after a disruption or failure.	n/a	Yes
S4R_FDF_182	**2.6 RAMS requirements**		
S4R_FDF_478	The Framework shall provide a safe communication path for transmission/reception of datasets using a safety layer.	n/a	n/a

Id	Text	Safety related	Security related
S4R_FDF_479	The Framework shall offer application interfaces according to the safety layer needed: •non-critical (SIL0) •SIL2 •SIL4 where the ability to provide SIL2 and SIL4 APIs depends on the specific implementation of the framework (on HW/SW).	n/a	n/a
S4R_FDF_480	The Framework shall guarantee the integrity and validity of the received data to meet the requirements for SIL2 (according to IEC61508-1 [5]). SDTv2, as defined in IEC61375-2-3 Annexe B [2], provides this safety level for PFH ≥ 10E-7 < 10E-6 (1% for black channel communication).	n/a	n/a
S4R_FDF_481	The Framework shall guarantee the integrity and validity of the received data to meet the requirements for SIL4 (according to IEC61508-1). A PFH ≥ 10E-9 < 10E-8 (1% for black channel communication) is needed.	n/a	n/a
S4R_FDF_482	The Framework shall inform the application of communication losses, which enable the application to decide whether to set the system into the ‘safe state’.	n/a	n/a
S4R_FDF_483	The Framework shall monitor the operational state of the ECU (and its function(s)) by appropriate means and report in case of failure. I.e., implementing error detection and correction (EDC) technique.	Yes	n/a
S4R_FDF_484	The Framework shall share its operational state with all other ECUs in its functional group(s).	n/a	n/a
S4R_FDF_485	The Framework shall detect and verify the operational status of other redundant ECUs.	n/a	n/a
S4R_FDF_486	The Framework shall inform the application of the operational status of all other ECUs in its functional group(s).	n/a	n/a
S4R_FDF_487	The Framework shall be operational within 60 seconds from power-up.	n/a	n/a
S4R_FDF_488	The Framework shall perform a self-test of the ECU on power-up.	n/a	n/a
S4R_FDF_467	***2.6.1 Configuration management***		
S4R_FDF_431	The Framework shall be configurable on ECU reset or start-up by a local configuration.	n/a	n/a
S4R_FDF_432	The Framework shall be able to receive an additional remote configuration via network.	n/a	n/a
S4R_FDF_433	The Framework shall check the validity and integrity of any configuration. This could be a CRC, MD or signature created by tooling.	Yes	Yes
S4R_FDF_434	The Framework shall check the origin of remote configurations and ignore false configurations. Remote configurations must be certified.	Yes	Yes
S4R_FDF_435	The remote configuration's properties shall take precedence over the same properties of the local configuration.	n/a	n/a

Id	Text	Safety related	Security related
	This relates to dynamic vs. static configuration, e.g. direction dependent addressing and default parameters.		
S4R_FDF_436	The Framework shall provide a local interface to retrieve static and dynamic configuration properties by a host application.	n/a	n/a
S4R_FDF_437	The Framework shall provide a remote (network) interface to retrieve static and dynamic configuration properties of an ECU.	n/a	n/a
S4R_FDF_438	The Framework's local configuration shall define the necessary properties for local communication needs. Note: Annex C of IEC 61375-2-3 [2] defines an XML format which covers most properties of a communication framework. Train-wide communication depends on train inauguration and may therefore not be possible with local configurations, only. This depends on the future network layout (defined in WP1).	n/a	n/a

Table 3: FDF requirements.

3 Conclusion

The Train Control and Monitoring System (TCMS) is often colloquially called “Brain of the Train” but existing solutions are not yet as smart and efficient as the brain. Due to historic reasons, technological limitations and certification costs, each individual subsystem in a train has used its own electronic architecture with very limited interoperability.

Removing the need for those custom island solutions and integrating the train functions into one common converged platform for communication and computation will maximize the interoperability while minimizing physical complexity and costs, which is the joint interest of the manufacturers. This is achieved by the Integrated Modular Platform (IMP), which can host any application up to the most critical applications of the train to the highest certification requirements.

The “computation” part of this platform is achieved by the Functional Distribution Framework. It Facilitates modular integration of applications. Multiple applications can be installed and run within a control computer. The system will guarantee the functional safety and freedom from interference as well as the interoperability for applications on different operating systems and platforms as well as an abstraction from underlying hardware and communications.

This set of requirements is refined from the design goals defined in deliverable D2.1 ´Report on state-of-the-art of ‘functional distribution architecture’ frameworks and solutions´ [8] and completed with insights resulting from the activities reflected on deliverables D2.2 ´Report on analysis of ‘functional distribution architecture’ frameworks and solutions´ [9], which performs a comparative analysis between COTS frameworks and solutions for the deployment of next generation TCMS systems, and D2.3 ´Report on ‘TCMS framework concept’ design, security concepts, and assessment´ [1], in which the reference architecture is defined.

With this deliverable, Safe4RAIL provides a complete definition of requirements for a TCMS framework that supports functional distribution, mixed-criticality, hardware abstraction, communication/coupling services abstraction, railway safety standard compliance, railway domain life-cycle and relevant railway domain specific requirements. Besides, recommendations for standardization and certification of next generation TCMS embedded platform by this document in aim to boost competitiveness and preserve the global leadership of the European transport industry.

4 Bibliography

[1] Deliverable D2.3, Report on ‘TCMS framework concept’ design, security concepts, and assessment.

[2] IEC 61375-2-3:2015. Electronic railway equipment - train communication network (TCN) - part 2-3: TCN communication profile.

[3] Directive (EU) 2016/797 of the European Parliament and of the Council of 11 May 2016 on railway safety, OJ L 138, 26.5.2016.

[4] 1302/2014/CE - COMMISSION REGULATION (EU) No 1302/2014 of 18 November 2014 concerning a technical specification for interoperability relating to the ‘rolling stock - locomotives and passenger rolling stock’ subsystem of the rail system in the European Union.

[5] IEC 61508-1:2010. Functional safety of electrical/electronic/programmable electronic safety-related systems - part1: General requirements.

[6] EN 50128:2011 - Railway applications - Communication, signalling and processing systems - Software for railway control and protection systems.

[7] IEC 61375-2-1 Ed.1. Electronic railway equipment - train communication network - part 2-1: WTB - wire train bus.

[8] Deliverable D2.1, Report on state-of-the-art of ‘functional distribution architecture’ frameworks and solutions.

[9] Deliverable D2.2, Report on analysis of ‘functional distribution architecture’ frameworks and solutions.

[10] DOORS:
https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/DOORS

Annex A – FDF Components: Traceability Matrix

This annex contains the traceability between the FDF requirements and FDFcomponents described in D2.3 [1].

Id	**FDF components**	**FDF requirements**
FDS	**A.1. Functional Distribution Services**	
FDS_FWM	A.1.1. Framework Manager	N/A
FDS_FM	A.1.2. Function Manager	S4R_FDF_193, S4R_FDF_194, S4R_FDF_195, S4R_FDF_196, S4R_FDF_197, S4R_FDF_205, S4R_FDF_206, S4R_FDF_208, S4R_FDF_210, S4R_FDF_315, S4R_FDF_497, S4R_FDF_498, S4R_FDF_506, S4R_FDF_507, S4R_FDF_519, S4R_FDF_520, S4R_FDF_521, S4R_FDF_522, S4R_FDF_523, S4R_FDF_524, S4R_FDF_525, S4R_FDF_526, S4R_FDF_527, S4R_FDF_528, S4R_FDF_585, S4R_FDF_586, S4R_FDF_587, S4R_FDF_588, S4R_FDF_589, S4R_FDF_592, S4R_FDF_594, S4R_FDF_602, S4R_FDF_603, S4R_FDF_604, S4R_FDF_606, S4R_FDF_607, S4R_FDF_609, S4R_FDF_610, S4R_FDF_684, S4R_FDF_685, S4R_FDF_686, S4R_FDF_687, S4R_FDF_688, S4R_FDF_689, S4R_FDF_690, S4R_FDF_691, S4R_FDF_693, S4R_FDF_694, S4R_FDF_695, S4R_FDF_698, S4R_FDF_700, S4R_FDF_737, S4R_FDF_741, S4R_FDF_755, S4R_FDF_759, S4R_FDF_766, S4R_FDF_767, S4R_FDF_768, S4R_FDF_782, S4R_FDF_783, S4R_FDF_784, S4R_FDF_785, S4R_FDF_786
FDS_VM	A.1.3. Variable Manager	S4R_FDF_221, S4R_FDF_223, S4R_FDF_224, S4R_FDF_225, S4R_FDF_226, S4R_FDF_355, S4R_FDF_356, S4R_FDF_359, S4R_FDF_361, S4R_FDF_365, S4R_FDF_416, S4R_FDF_494, S4R_FDF_509, S4R_FDF_510, S4R_FDF_511, S4R_FDF_541, S4R_FDF_558, S4R_FDF_562, S4R_FDF_563, S4R_FDF_564, S4R_FDF_636, S4R_FDF_653, S4R_FDF_654, S4R_FDF_655, S4R_FDF_656, S4R_FDF_694, S4R_FDF_709, S4R_FDF_735, S4R_FDF_746, S4R_FDF_753, S4R_FDF_780, S4R_FDF_781, S4R_FDF_783
FDS_MS	A.1.4. Message Manager	S4R_FDF_221, S4R_FDF_222, S4R_FDF_227, S4R_FDF_231, S4R_FDF_232, S4R_FDF_493, S4R_FDF_512, S4R_FDF_513, S4R_FDF_711, S4R_FDF_728, S4R_FDF_750, S4R_FDF_756
FDS_CRYM	A.1.5. Crypto Manager	S4R_FDF_411, S4R_FDF_412, S4R_FDF_414, S4R_FDF_416, S4R_FDF_646, S4R_FDF_647, S4R_FDF_667, S4R_FDF_670
FDS_IOM	A.1.6. IO Manager	S4R_FDF_226, S4R_FDF_261, S4R_FDF_263, S4R_FDF_264, S4R_FDF_265, S4R_FDF_266, S4R_FDF_267, S4R_FDF_268, S4R_FDF_269, S4R_FDF_270, S4R_FDF_271, S4R_FDF_272, S4R_FDF_273, S4R_FDF_274, S4R_FDF_275, S4R_FDF_276, S4R_FDF_277, S4R_FDF_312, S4R_FDF_315, S4R_FDF_546, S4R_FDF_547, S4R_FDF_607, S4R_FDF_609, S4R_FDF_709, S4R_FDF_716, S4R_FDF_735, S4R_FDF_764, S4R_FDF_766
FDS_CONFM	A.1.7. Configuration Manager	S4R_FDF_525, S4R_FDF_610, S4R_FDF_613, S4R_FDF_614, S4R_FDF_615, S4R_FDF_616, S4R_FDF_618, S4R_FDF_619,

Id	FDF components	FDF requirements
		S4R_FDF_620, S4R_FDF_621, S4R_FDF_622, S4R_FDF_623, S4R_FDF_624, S4R_FDF_625, S4R_FDF_626, S4R_FDF_664, S4R_FDF_742
FDS_SM	A.1.8. Synchronization Manager	S4R_FDF_409, S4R_FDF_410, S4R_FDF_413, S4R_FDF_415, S4R_FDF_417, S4R_FDF_418, S4R_FDF_419, S4R_FDF_420, S4R_FDF_421, S4R_FDF_429, S4R_FDF_430, S4R_FDF_606, S4R_FDF_670, S4R_FDF_671, S4R_FDF_672, S4R_FDF_674, S4R_FDF_676, S4R_FDF_677, S4R_FDF_678, S4R_FDF_679, S4R_FDF_680, S4R_FDF_681, S4R_FDF_730, S4R_FDF_731, S4R_FDF_732
FDS_HM	A.1.9. Health Manager	S4R_FDF_311, S4R_FDF_313, S4R_FDF_315, S4R_FDF_522, S4R_FDF_549, S4R_FDF_551, S4R_FDF_552, S4R_FDF_553, S4R_FDF_554, S4R_FDF_555, S4R_FDF_556, S4R_FDF_557, S4R_FDF_558, S4R_FDF_728, S4R_FDF_745, S4R_FDF_746, S4R_FDF_754, S4R_FDF_758, S4R_FDF_769
FDS_NM	A.1.10. Network Manager	S4R_FDF_221, S4R_FDF_222, S4R_FDF_223, S4R_FDF_224, S4R_FDF_225, S4R_FDF_227, S4R_FDF_229, S4R_FDF_230, S4R_FDF_231, S4R_FDF_232, S4R_FDF_315, S4R_FDF_493, S4R_FDF_494, S4R_FDF_508, S4R_FDF_509, S4R_FDF_510, S4R_FDF_511, S4R_FDF_512, S4R_FDF_513, S4R_FDF_541, S4R_FDF_543, S4R_FDF_558, S4R_FDF_709, S4R_FDF_711, S4R_FDF_728, S4R_FDF_745, S4R_FDF_746, S4R_FDF_750, S4R_FDF_751, S4R_FDF_753, S4R_FDF_756, S4R_FDF_780, S4R_FDF_781
FDS_UAM	A.1.11. User Account Manager	S4R_FDF_669, S4R_FDF_673, S4R_FDF_674, S4R_FDF_677, S4R_FDF_749
FDS_MONM	A.1.12. Monitoring Manager	S4R_FDF_311, S4R_FDF_313, S4R_FDF_315, S4R_FDF_355, S4R_FDF_356, S4R_FDF_357, S4R_FDF_358, S4R_FDF_359, S4R_FDF_361, S4R_FDF_362, S4R_FDF_363, S4R_FDF_364, S4R_FDF_365, S4R_FDF_521, S4R_FDF_522, S4R_FDF_562, S4R_FDF_563, S4R_FDF_564, S4R_FDF_694, S4R_FDF_704, S4R_FDF_712, S4R_FDF_713, S4R_FDF_733, S4R_FDF_761
FDS_TM	A.1.13. Topology Manager	S4R_FDF_495, S4R_FDF_496, S4R_FDF_760
FDS_LM	A.1.14. Log Manager	S4R_FDF_378, S4R_FDF_379, S4R_FDF_380, S4R_FDF_381, S4R_FDF_382, S4R_FDF_383, S4R_FDF_384, S4R_FDF_385, S4R_FDF_386, S4R_FDF_387, S4R_FDF_388, S4R_FDF_389, S4R_FDF_390, S4R_FDF_574, S4R_FDF_575, S4R_FDF_576, S4R_FDF_580, S4R_FDF_581, S4R_FDF_582
FDS_DM	A.1.15. Deployment Manager	S4R_FDF_566, S4R_FDF_567, S4R_FDF_568, S4R_FDF_569, S4R_FDF_570, S4R_FDF_571, S4R_FDF_573, S4R_FDF_630, S4R_FDF_631, S4R_FDF_635, S4R_FDF_636, S4R_FDF_639, S4R_FDF_640, S4R_FDF_658, S4R_FDF_659, S4R_FDF_660, S4R_FDF_661, S4R_FDF_662, S4R_FDF_663, S4R_FDF_664, S4R_FDF_665, S4R_FDF_666, S4R_FDF_770, S4R_FDF_787
FDS_RM	A.1.16. Redundancy Manager	S4R_FDF_508, S4R_FDF_510, S4R_FDF_511, S4R_FDF_686, S4R_FDF_688, S4R_FDF_689, S4R_FDF_690, S4R_FDF_691, S4R_FDF_693, S4R_FDF_709
FDS_SMM	A.1.17. Security Monitoring Manager	S4R_FDF_409, S4R_FDF_410, S4R_FDF_413, S4R_FDF_415, S4R_FDF_417, S4R_FDF_418, S4R_FDF_419, S4R_FDF_420, S4R_FDF_421, S4R_FDF_429, S4R_FDF_430, S4R_FDF_670, S4R_FDF_671, S4R_FDF_672, S4R_FDF_674, S4R_FDF_676,

Id	FDF components	FDF requirements
		S4R_FDF_677, S4R_FDF_678, S4R_FDF_679, S4R_FDF_680, S4R_FDF_681, S4R_FDF_730, S4R_FDF_731, S4R_FDF_732
HAS	**A.2. Hardware Access Services**	
HAS_IODM	A.2.1. IO Driver Manager	S4R_FDF_169
HAS_NICDM	A.2.2. NIC Driver Manager	N/A
HAS_WSDM	A.2.3. WD Driver Manager	S4R_FDF_552, S4R_FDF_553
HAS_ECUDM	A.2.4. ECU Driver Manager	S4R_FDF_658, S4R_FDF_663, S4R_FDF_665
OSS	**A.3. Operating System Services**	
OSS_FM	A.3.1. File Manager	S4R_FDF_636, S4R_FDF_644, S4R_FDF_645, S4R_FDF_648, S4R_FDF_649, S4R_FDF_650, S4R_FDF_651, S4R_FDF_652, S4R_FDF_653, S4R_FDF_654, S4R_FDF_655, S4R_FDF_656, S4R_FDF_657
OSS_MM	A.3.2. Memory Manager	S4R_FDF_384, S4R_FDF_543, S4R_FDF_644
OSS_TM	A.3.3. Time Manager	S4R_FDF_168, S4R_FDF_235, S4R_FDF_236, S4R_FDF_237, S4R_FDF_238, S4R_FDF_239, S4R_FDF_240, S4R_FDF_544, S4R_FDF_545, S4R_FDF_736, S4R_FDF_762
OSS_LM	A.3.4. Library Manager	N/A
OSS_SM	A.3.5. Socket Manager	N/A
OSS_CM	A.3.6. Concurrency Manager	S4R_FDF_216, S4R_FDF_217, S4R_FDF_497, S4R_FDF_529, S4R_FDF_530, S4R_FDF_590, S4R_FDF_684
OSS_EM	A.3.7. Execution Manager	S4R_FDF_315, S4R_FDF_659

Table 4: FDF component - Traceability matrix.

Annex B – Safety Countermeasures: Traceability Matrix

This table shows the list of safety countermeasures. These countermeasures were concluded from the Safety Concept completed in deliverable D2.3 [1]. These countermeasures need to be covered by the FDF Safety requirements and, thus, this table shows how they are traced.

Id	Safety Concept Countermeasures defined in D2.3	FDF requirements
HA_COM_01	The Framework shall provide a communication service that makes received messages available to the Application functions within defined timely bounds (deterministic receiving).	S4R_FDF_232
HA_COM_02	The Framework shall provide a communication service that allows sending messages within defined timely bounds and with defined periodicity, and receiving messages within defined maximum delay (deterministic communication).	S4R_FDF_231, S4R_FDF_750
HA_COM_03	The Framework shall define, configure, assess and guarantee performance of communication channels, including priority, throughput, jitter, latency, response time.	S4R_FDF_307, S4R_FDF_308
HA_COM_04	The Framework shall implement Communication service without any operation on the messages' safety layer content.	S4R_FDF_751
HA_COM_05	The Framework shall monitor the communication between remote functions.	S4R_FDF_733
HA_COM_06	The Framework shall inform the Application function(s) in case of loss of valid communication between remote functions.	S4R_FDF_745
HA_MON_01	The Framework shall assign to the Monitoring Function privilege for read-only the variables stored into SIL0 Memory spaces, or to all the Memory spaces if data alteration during reading can be excluded, and execute Monitoring services without any disturb or unintended effects due to other Service and Application functions.	S4R_FDF_752
HA_MSG_01	The Framework shall ensure the integrity of safety-related data exchanged by communication protocol(s) implementing a safety layer (i.e. a safety code) with source and/or destination identifiers, information that the transmitter is operating properly, redundancy field allowing error detection and assuring data integrity.	S4R_FDF_711
HA_MSG_02	The Framework shall ensure the timeliness and sequence of data exchanged and results of safety algorithms, e.g. by sequence number and/or time stamps generated by unique identifier related to the cycle (or equivalent measures).	S4R_FDF_728
HA_MSG_03	The Framework shall protect the communication of safety-related data against cyber-attack, ensuring data authenticity and confidentiality, e.g. by software and/or hardware security mechanisms (e.g. cryptographic mechanisms, control of access to data).	S4R_FDF_643
HA_MSG_04	The Framework shall use protocols for diagnostic, maintenance, configuration and communication of non-yes data with different structures than one(s) used for the communication of safety-related data.	S4R_FDF_221, S4R_FDF_526, S4R_FDF_618, S4R_FDF_711, S4R_FDF_770
HA_MSG_05	The Framework shall guarantee that Message Function read and write the required variables in a safe way, i.e. variables are read	S4R_FDF_223,

Id	Safety Concept Countermeasures defined in D2.3	FDF requirements
	without altering their value and written according to specification (set during configuration).	S4R_FDF_224, S4R_FDF_753
HA_MSG_06	The Framework shall check the integrity (i.e. information is complete and not altered) of incoming messages containing safety.	S4R_FDF_414
HA_MSG_07	The Framework shall check the timeliness and sequence of messages containing safety-data, exchanged between remote functions.	S4R_FDF_728
HA_MSG_08	The Framework shall check the authenticity of incoming message containing safety data, exchanged between remote functions.	S4R_FDF_222, S4R_FDF_413, S4R_FDF_417
HA_MSG_09	The Framework and Application functions shall ignore the content and discharge a message (containing safety-data) when a communication error is identified through the messages authenticity, integrity, timeliness or sequence checks.	S4R_FDF_745, S4R_FDF_746
HA_MSG_10	The Framework shall implement reactions against errors in the communication of safety-related data that are functionally independent by any non-trusted transmission.	S4R_FDF_745, S4R_FDF_746
HA_MSG_11	The Framework shall guarantee the validity of yes data exchanged between remote functions, through messages composing and decomposing into variables carried out by the Message Function, with the same SIL assigned to the Application function(s) using messages and variables involved.	S4R_FDF_780
HA_MSG_12	The Framework shall allow Message Function to access to memory space(s) containing messages and to memory space(s) containing variables with the same SIL.	S4R_FDF_781
HA_FRM_01	The Framework shall generate Partitions according to the Configuration file of the Application functions to be executed (which specify the SIL, address and size of the memory space, and time window inside the global scheduling plan) and protect each partition’s addressing space through specific memory protection mechanisms, e.g. by a hardware memory management unit, and management of access privilege and restrictions.	S4R_FDF_524, S4R_FDF_525
HA_FRM_02	The Framework shall provide to the partition assigned to an Application functions the computational resources (e.g. CPU time, memory) required into the Configuration file in order to meet the (worst-case) timing requirements.	S4R_FDF_526
HA_FRM_03	The Framework shall provide to the Application functions the read-write privilege only to variables (and related input/output, if any) they are allowed to publish and the read-only privilege to software code, parameters and variables (and related input, if any) they are subscribed to.	S4R_FDF_223, S4R_FDF_224
HA_FRM_04	The Framework shall guarantee that Application functions read / write variables, managing consequently the related platform's I/O, only if the required privilege is provided.	S4R_FDF_223, S4R_FDF_224, S4R_FDF_620
HA_FRM_05	The Framework shall call Services required for the scheduled execution of the Application functions.	S4R_FDF_692
HA_FRM_06	The Framework shall be able to generate partitions and allocate resources for Application function(s) requiring multiple instances (for the implementation of reliable-safe architecture).	S4R_FDF_782
HA_FRM_07	The Framework shall detect an invalid operation in the partition attempts by the Application function(s), e.g. access to a Memory	S4R_FDF_384,

Id	Safety Concept Countermeasures defined in D2.3	FDF requirements
	space without the required reading or writing privilege.	S4R_FDF_555
HA_FRM_08	The Framework shall notify a Fault condition, in case of invalid operation in the partition attempt (fatal Fault), to all the Application functions involved.	S4R_FDF_528, S4R_FDF_687
HA_FRM_09	The Framework shall inform the Application functions in case of unavailability of services required for their scheduled execution, or in case of incorrect call (different than scheduled).	S4R_FDF_754
HA_FRM_10	The Framework shall protect and guarantee the independence of multiple instances of an Application function (e.g. implementing reliable-safe architecture), e.g. by data diversity (e.g. different time-stamp guarantying data freshness), timing diversity (instances do not execute simultaneously the same safety-related software modules), independent (hardware) resources.	S4R_FDF_208, S4R_FDF_524, S4R_FDF_588, S4R_FDF_592, S4R_FDF_606, S4R_FDF_686
HA_FRM_11	The Framework shall guarantee the spatial separation among Partition, in order to ensure that no process in one partition can modify (without authorization) software code or application data (i.e.. write to memory data sections, stacks and code) or manage the I/O assigned to another partition, e.g. through the protection of their memory addressing space and the management of privilege and restrictions for variables read / write and for access to I/O.	S4R_FDF_524
HA_FRM_12	The Framework shall guarantee spatial separation between memory spaces containing read-only (including software code and parameters) and read-write variables, variables with different SIL, variables used by multiple independent instances of the Application function.	S4R_FDF_783
HA_FRM_13	The Framework shall prevent any unintended interactions between the Operating system activities and the Application functions, through the definition of formal boundaries and interaction modalities and protecting the Operating System (data sections, stacks, and code) against undue calls from the Application and Services functions (e.g. with an invalid handle, object, address or out of range value; in the wrong context; without the necessary permissions).	S4R_FDF_685
HA_FRM_14	The Framework shall generate partitions and allocate resources with the same SIL assigned to the Application functions to be executed, including memories spaces storing data with the same (unique) SIL.	S4R_FDF_423, S4R_FDF_526
HA_FRM_15	The Framework shall assign privileges for read-write access to a Memory space only to independent Application functions with the same SIL. Read-only access could be assigned to remaining Application functions, if data alteration during reading can be excluded.	S4R_FDF_784
HA_FRM_16	The Framework shall guarantee the read-write access to memory spaces (according to the assigned privileges) with the same SIL assigned to the Application function(s) and variables stored.	S4R_FDF_785
HA_FRM_17	The Framework shall guarantee the effectiveness of call(s) to Service function(s) with the same SIL assigned to the Application functions using Service(s).	S4R_FDF_755
HA_FRM_18	The Framework shall detect the unavailability of Services required for the scheduled executions of the Application functions and their incorrect call (different than scheduled)	S4R_FDF_754
HA_CONF_01	The Framework shall instantiate messages and variable according to the Configuration file, which specifies at least: messages'	S4R_FDF_623,

Id	Safety Concept Countermeasures defined in D2.3	FDF requirements
	identifier, variables, to receive or to send, schedule, deadline; variables' identifier, type, range, default value, deadline.	S4R_FDF_756
HA_CONF_02	The Framework shall accept only certified remote Configuration file (coming from a verified source), protected against data corruption, e.g. by CRC.	S4R_FDF_434
HA_CONF_03	The Framework shall verify the validity and integrity of the Configuration file, before and after the end of the inauguration services, e.g. by CRC, MD or signature created by tooling.	S4R_FDF_433
HA_CONF_04	The Framework shall verify the validity of results coming from the inauguration (Train Topology Database or equivalent data structure) and their coherence with the Configuration file.	S4R_FDF_496
HA_CONF_05	The Framework shall not execute the Application functions in case of any error detected in the Configuration file or non-valid results coming from the inauguration or undue operation on the Configuration data, and notify a (fatal) Fault condition to all the Application function(s) involved.	S4R_FDF_758
HA_CONF_06	The Framework shall assure that re-configuration required for new or modified Application functions is performed involving all the Application functions to be executed, or anyway the existing configuration for the remaining Application functions is not altered.	S4R_FDF_630, S4R_FDF_631, S4R_FDF_664, S4R_FDF_787
HA_CONF_07	The Framework shall read, parse, load and check data in the Configuration file and configure the platform accordingly, with the same SIL assigned to the related Application function.	S4R_FDF_546, S4R_FDF_547, S4R_FDF_615, S4R_FDF_616, S4R_FDF_743, S4R_FDF_744
HA_CONF_08	The Framework shall load the Configuration file during the execution of the inauguration services and assure that any re-configuration (re-loading of the Configuration file or loading of a new Configuration file) is performed involving all the Application functions to be executed.	S4R_FDF_741, S4R_FDF_742
HA_FNM_01	The Framework shall control the execution (start, stop, synchronizing to external trigger) of Application functions assigned to each individual partition, through the deterministic management of timers (for sequential execution) and semaphores (for sequential and concurrent execution), according to their scheduling plans and to processes priority.	S4R_FDF_314, S4R_FDF_530, S4R_FDF_606, S4R_FDF_684
HA_FNM_02	The Framework shall execute an Application function, giving access to memory resources, only when required by its scheduling plan (and take away access otherwise).	S4R_FDF_786
HA_FNM_03	The Framework shall implement Service functions whose response times allow the real-time execution of processes and the fulfilment of the most restrictive response time required by the Application functions to be executed.	S4R_FDF_235, S4R_FDF_520
HA_FNM_04	The Framework shall implement mechanisms to ensure the execution of real-time processes in spite of transient temporal violations, e.g. due to inter-module communications acknowledgements, time-outs, access to memory, interrupts.	S4R_FDF_520, S4R_FDF_521

Id	Safety Concept Countermeasures defined in D2.3	FDF requirements
HA_FNM_05	The Framework shall avoid interrupts or manage them through the Operating system only (even if triggered by the Application functions or by hardware), avoiding any disturb to the time partitioning, i.e. without any change of the time budget allocation.	S4R_FDF_270, S4R_FDF_277, S4R_FDF_685, S4R_FDF_759
HA_FNM_06	The Framework shall monitor the execution (start, stop, synchronizing to external trigger) of processes with respect to defined timing bounds for (intra-partition and inter-partition) communication and processing.	S4R_FDF_314
HA_FNM_07	The Framework shall notify a Fault condition, in case of error in the execution of processes according to the scheduling plans, including the violation of timing bounds (fatal Fault), to all the Application functions involved.	S4R_FDF_522, S4R_FDF_528
HA_FNM_08	The Framework shall implement temporal partitioning, by ensuring that a process within a given time budget cannot be affected by the actions of any other task from other partitions, in terms of rate, latency, jitter and duration of the scheduled access.	S4R_FDF_208
HA_FNM_09	The Framework shall control the execution of processes and the transmission of messages (according to their scheduling plans) with the same SIL assigned to the involved Application functions.	S4R_FDF_519
HA_FLT_01	The Framework shall provide services for the detection of faults of (hardware) resources used by Service and Application functions, at the power up (i.e. during the initialization) and periodically during the operation (nominal and degraded phases), e.g. test memories containing yes data are totally tested at the initialization phase and at any new allocation and cyclically at run-time.	S4R_FDF_738
HA_FLT_02	The Framework shall provide services for the detection of faults during the installation of the Applications software (otherwise, to be required to the Applications).	S4R_FDF_659
HA_FLT_03	The Framework shall provide services for the detection of faults during the run-time execution of the Application function code (otherwise, to be required to the Application function), e.g. by monitoring the process and data flow and comparing their state to configured constraints (Program Flow Monitoring), by checking variables values against predefined range and for plausibility, by detecting and correcting errors in sensitive information (Error Detecting and Correcting Codes).	S4R_FDF_314, S4R_FDF_483, S4R_FDF_733, S4R_FDF_788, S4R_FDF_789
HA_FLT_04	The Framework shall execute services for Fault detection, isolation, notification and reaction processes with the highest priority, without any disturb or unintended effects due to other Service and Application functions.	S4R_FDF_558, S4R_FDF_758, S4R_FDF_761
HA_FLT_05	The Framework shall provide services for Fault detection and isolation without any disturb or unintended effects on the execution and performance (e.g. latency/jitter, sampling rate or resource reservation) of other Service and Application functions.	S4R_FDF_426, S4R_FDF_558
HA_FLT_06	The Framework shall verify the capability to notify a Fault condition under a representative set of failure scenarios.	S4R_FDF_581, S4R_FDF_758
HA_FLT_07	The Framework shall inhibit the execution of the Application function in case of negative results of the initial code integrity check.	S4R_FDF_737
HA_FLT_08	The Framework, after the detection of a condition that blocks or threats the proper execution of Service or Application functions (fatal Fault), shall notify a Fault condition to all the Application functions involved, in a time that is compatible with their timely transition into safe state (i.e. not later than the maximum time for failure detection and negation specified by the Applications).	S4R_FDF_758

Id	Safety Concept Countermeasures defined in D2.3	FDF requirements
HA_FLT_09	The framework shall manage the interaction between Service and Application functions:, _avoiding that Service functions can force the outputs independently from the Application function when active, during operation (normal and degraded phases);, _preventing the access to any off-line service (e.g. validation and verification support) at the power up, and during the initialization and the operating (nominal and degraded) phases;, _guarantying the retention of a safe state after a fatal Fault (i.e. condition that blocks or threats the proper execution of Service or Application functions).	S4R_FDF_766, S4R_FDF_767, S4R_FDF_768
HA_FLT_10	The Framework shall detect, isolate, notify and react to fault with the highest SIL assigned to the safety-related Application functions to be executed.	S4R_FDF_558, S4R_FDF_758, S4R_FDF_761
HA_TM_01	The Framework shall synchronize the local computer clock with the external global clock source and keep it synchronized with a maximum defined deviation fixed.	S4R_FDF_236
HA_TM_02	The Framework shall not finalize the inauguration and allow operation without a global time valid (i.e. aligned with the external global clock) and taken as unique reference by all Service and Application functions, independently from the partitions execution.	S4R_FDF_736
HA_TM_03	The Framework shall monitor the alignment with the external global clock, the effectiveness of the global time dissemination and functions synchronization.	S4R_FDF_240, S4R_FDF_760
HA_TM_04	The Framework shall notify a Fault condition, in case of error in the global time synchronization (fatal Fault), to all the Application functions involved.	S4R_FDF_238
HA_TM_05	The Framework shall synchronize the local computer clock with the external global clock source and keep it synchronized independently from the execution of the different partitions' processes.	S4R_FDF_236, S4R_FDF_762
HA_TM_06	The Framework shall disseminate the global time and/or detect any misalignment against the external reference time, with the highest SIL assigned to the Application functions to be executed.	S4R_FDF_240, S4R_FDF_760
HA_IO_01	The Framework shall provide services that allow the Application function to read the last valid value stored into an exchange variable and to update this value according to the status of the related input (coming from the interfaced object).	S4R_FDF_269, S4R_FDF_735
HA_IO_02	The Framework shall provide services that allow the Application function to write a value into an exchange variable and to update accordingly to the status of the related output (toward the interfaced object).	S4R_FDF_265, S4R_FDF_266, S4R_FDF_267, S4R_FDF_273, S4R_FDF_274, S4R_FDF_275, S4R_FDF_276
HA_IO_03	The Framework shall identify univocally each input / output interfacing external objects, each exchange variable, and each association between them, according to the Configuration file(s) of the Application function(s) using them.	S4R_FDF_622
HA_IO_04	The Framework shall read and write all the I/O related to the executed Application function in one cycle only, guarantying that the current value of every input is stored in the associated exchange variable at the beginning of each cycle and the current value of every output is set according to the value stored in the associated exchange variable at the end of each cycle..	S4R_FDF_268, S4R_FDF_275

Id	Safety Concept Countermeasures defined in D2.3	FDF requirements
HA_IO_05	The Framework shall detect inconsistency between the values stored into the exchange variables and the status of the related platform's input and output.	S4R_FDF_365, S4R_FDF_769
HA_IO_06	The Framework, in case of any inconsistency between the values stored into an exchange variable and the status of the related platform's input / output, shall inform the Application function(s) with read and/or write privilege on this variable.	S4R_FDF_558
HA_IO_07	The Framework shall be able to provide independence between different (set of) input / output interfacing external objects (that can be request by Application function to implement reliable-safe architecture).	S4R_FDF_734
HA_IO_08	The Framework shall guarantee the updating of each exchange variable (according to the status of related input) and its reading with the SIL assigned to the Application function(s) involved and to the specific variable.	S4R_FDF_735
HA_IO_09	The Framework shall guarantee the updating the status of each output (according to value stored into the related exchange variable) and its writing with the SIL assigned to the Application function(s) involved and to the specific variable.	S4R_FDF_555
HA_IO_10	The Framework shall allow I/O Function to access only to memory space with the same SIL.	S4R_FDF_764, S4R_FDF_765

Table 5: FDF safety countermeasures - Traceability matrix.

Annex C – Security Countermeasures: Traceability Matrix

This table shows the list of security countermeasures, as a result of the Security Concept completed in deliverable D2.3 [1]. These countermeasures need to be covered by the FDF Security requirements and, thus, are traced to those. **Id**	**Security Countermeasures defined in D2.3**	**FDF requirements**
SEC_COUNT_1	Hardware-based security solutions: chip or TPM A hardware security chip or Trusted Platform Module (TPM) is a tamper-resistance computing chip that can securely store artefacts used to authenticate, such as, passwords, certificates and cryptographic keys. The countermeasure would be used in combination with a crypto USB or smartcard token in which personnel and applications certificates can be stored to be used for public key authentication, PIN support, user-defined key restriction (i.e. one-time password, a limited number of users) and key audit counter (i.e. counts down with each key usage). FDF can use this technology for identification and authentication ECUs and applications, encryption, secure key storage and integrity verification. The result of authentication process shall be obscured and the number of invalid access shall be configurable.	S4R_FDF_409, S4R_FDF_410, S4R_FDF_413, S4R_FDF_414, S4R_FDF_418, S4R_FDF_667, S4R_FDF_670, S4R_FDF_672, S4R_FDF_676, S4R_FDF_678, S4R_FDF_679
SEC_COUNT_2	Password policy Username and password are required worldwide in order to avoid any user impersonation and to login a system and communicate between software components. Password robustness is also required to avoid any password hacking method. Detection of this attack method, for instance blocking the system when a fixed number of wrong passwords are typed, is also a way of improving security. Instead of username and password, there could also be used certificates as credentials to demonstrate who it is, person or application component.	S4R_FDF_409, S4R_FDF_410, S4R_FDF_670, S4R_FDF_672, S4R_FDF_674, S4R_FDF_679,

This table shows the list of security countermeasures, as a result of the Security Concept completed in deliverable D2.3 [1]. These countermeasures need to be covered by the FDF Security requirements and, thus, are traced to those. **Id**	**Security Countermeasures defined in D2.3**	**FDF requirements**
	Therefore, covering this aspect of security that is, limiting access to trusted users only to the FDF/OS with robust passwords, and as a consequence restricting and tailoring the accessible functions to them, the global security NIST recommendation for digital identity guidelines shall be ensured. It is recommended to enable password expiration and to control the number of invalid access for revoking access if needed.	S4R_FDF_680
SEC_COUNT_3	User and application profile policies Access to different services and data (including file systems) offered by FDF shall be restricted based on user and application profiles. Therefore, rules to determine which actions they are allowed to perform and their restrictions to access resources such as hardware (e.g., memory, network) or software (execution of programs or commands) should be taken into account to define and assign proper permissions to different user or application. The system must implement a security policy that specifies who or what may access a file system, and type of access permitted: for example, R-Read, W-Write, X-Execute and Supervisor/User mode. Moreover, there could be policies to enable: runtime, deployment, and so on. The least privilege shall be applied and an indication that a user profile expires or not shall be given. By means of a smart card or USB token, this renewal can be performed efficiently.	S4R_FDF_409, S4R_FDF_415, S4R_FDF_427, S4R_FDF_428, S4R_FDF_673, S4R_FDF_679, S4R_FDF_681
SEC_COUNT_4	Role-based Access Control (RBAC) A role-based access control shall be used to restricting of FDF access to only authorized users based on roles and permission. User roles can be assigned depending on specific operations, such as FDF admin, operator, application function developer, maintenance person, and so on. Each role will have different permissions/privileges, for example, the FDF administrator will have rights to edit system files, access network, edit user profiles and application profiles, and edit configuration files; whereas the operator will only have access to diagnostics data. Roles such as administrator with full privileges, and other with fewer privileges, such as, application developer, operator and maintenance person shall be considered. Roles have to be assigned to users so upon successful authentication of the user;	S4R_FDF_409, S4R_FDF_415, S4R_FDF_420, S4R_FDF_421, S4R_FDF_423, S4R_FDF_427, S4R_FDF_428, S4R_FDF_669, S4R_FDF_671,

This table shows the list of security countermeasures, as a result of the Security Concept completed in deliverable D2.3 [1]. These countermeasures need to be covered by the FDF Security requirements and, thus, are traced to those. **Id**	**Security Countermeasures defined in D2.3**	**FDF requirements**
	they are authorized as having the privileges associated with the assigned role. Administrator user role shall be able to create other user accounts and manage their privileges, always applying the least privilege philosophy. Applications shall also be configured with different privileges, for example, to restrict network, hardware, an operating system based on application's role. , Users and applications have to be categorised in roles allowing a RBAC security paradigm, and the least privilege shall be applied.	S4R_FDF_673
SEC_COUNT_5	Cryptography Apart from using secure channels to transfer data, the transferred sensitive data itself should be encrypted prior to sending it. In that way, a double security level is achieved in data transfer channels between an FDF and another system or FDF. If the secured channel is compromised, as data is encrypted, it could be almost impossible to interpret the data. In the case of FDF, it needs to be considered whether all data stored and messages shall be encrypted due to performance reasons, or whether only confidential or sensitive data that is susceptible of being compromised shall be encrypted. The FDF shall use established and tested encryption, hash algorithms and key sizes. Key generation shall be carried out using an effective random number generator. In Countermeasure 1, guidelines to choose a cryptographic chip or TPM are described and are still valid to this countermeasure. Generally accepted practices and recommendations can be found in documents such as NIST SP800-57. Implementation requirements can be found for example in ISO/IEC 19790.	S4R_FDF_411, S4R_FDF_412, S4R_FDF_414, S4R_FDF_416, S4R_FDF_646, S4R_FDF_647, S4R_FDF_667
SEC_COUNT_6	Session bindings Once authentication has taken place, it is desirable to continue using application/services over time without requiring authentication. To facilitate this behaviour, a session may be started in response to an authentication event, and continue the session until such time that it is terminated. Session management is preferable over the continual presentation of credentials. There are several mechanisms for managing a session over time; in this case, a session binding seems to be desirable. A	S4R_FDF_409

This table shows the list of security countermeasures, as a result of the Security Concept completed in deliverable D2.3 [1]. These countermeasures need to be covered by the FDF Security requirements and, thus, are traced to those. **Id**	**Security Countermeasures defined in D2.3**	**FDF requirements**
	session secret is shared between application and service being accessed. This secret binds the two ends of the session, allowing the application to continue using the service over time. This secret can be given using the security chip or TPM. • *Session timeout:* On the other hand, once the system has granted one session, it should control if it continues online in the long term, and if not, the session should be closed after the established time-out for inactivity is triggered. • *Concurrent session control:* Limiting the number of concurrent sessions per interface for the user (i.e. human, software process or devices).	
SEC_COUNT_7	<u>Network limited bandwidth</u> Usually, the first barrier used where data transfer is carried out in some kind of network is a firewall. A firewall can help to filter connections from known and unknown sources to reduce the incoming traffic to the system. Nevertheless, due to hardware and/or software restrictions and specifically in embedded devices, it is not possible to install and use a firewall as in a desktop computer. The measure that can be used is to enforce bandwidth limitation at the application or FDF level, together with the corresponding limitation of bandwidth at the network components. The use of internal network ports should be tailored and restricted (closed) as well as a firewall does create specific rules for incoming data. Whitelisting can also be defined to accept communications from different applications, but everything else is denied. If the communication does not appear on the white list, the communication is rejected. It is preferable to deny all traffic and permit only that traffic that is necessary. This security model is known as Deny All Permit Exception. In general, this is a more secure posture than using a blacklist that permits everything and blocks only traffic that someone decides is bad. All allowed traffic shall be logged for audit purposes. Although some comments address the network level, as stated this is beyond the scope of this security concept. Using Ethernet TSN, the monitoring and control of traffic is achieved to secure and protect critical traffic, together with physical network segmentation.	S4R_FDF_417

This table shows the list of security countermeasures, as a result of the Security Concept completed in deliverable D2.3 [1]. These countermeasures need to be covered by the FDF Security requirements and, thus, are traced to those. **Id**	**Security Countermeasures defined in D2.3**	**FDF requirements**
SEC_COUNT_8	Inventory of authorised and unauthorised assets (e.g. ECUs, software, sensors) An inventory of all authorized and unauthorized assets in FDF, including inputs, outputs, network, network devices, network addresses, machine names, purpose of each system, asset owner responsible for each of them. Authentication of all these devices shall be performed, for example to network level to determine authorised versus unauthorised systems. Furthermore, restricting access to memory and memory-mapped hardware shall be used for controlling hardware peripherals by reading from and writing to registers or memory blocks mapped to system memory. Physically disabling or removing connection ports and I/O devices help prevent disclosure of information or the introduction of malicious code into the system.	S4R_FDF_415, S4R_FDF_419, S4R_FDF_670, S4R_FDF_677
SEC_COUNT_9	Software-based memory protection unit The FDF shall prevent read/write access to an application's memory from non-trusted applications. Moreover, FDF may prevent non-trusted applications from executing code.	S4R_FDF_422, S4R_FDF_423, S4R_FDF_424, S4R_FDF_425, S4R_FDF_426
SEC_COUNT_10	Generation, protection and notification of audit and restoration data and system recovery The FDF shall be able to generate audit reports in the following categories: • Access control • Creation and restoration from backups • Changes in configuration files • Generation of audit logs events, such as access, use The FDF shall be able to notify about these events.	S4R_FDF_429, S4R_FDF_430, S4R_FDF_730, S4R_FDF_731, S4R_FDF_732

Table 6: FDF security countermeasures - Traceability matrix.

Annex D – Brake by Wire electronic control design: Traceability Matrix

This Annex collects Brake-by-Wire (BbW) application-specific needs in form of requirements that the FDF must satisfy. These requirements were defined in WP4 and are traced to FDF requirements in the table below.

Id	WP4 BbW requirements	FDF requirements
S4R_Bbw_1	The RBCU Logic Controller shall be capable to manage at least: • 14 analog inputs with 12 bit resolution • 1 digital input • 6 digital outputs to correctly perform its functionality. If the Logic controller has not sufficient I/O capabilities alternative peripheral for external devices must be included (e.g. SPI to interface an external ADC).	S4R_FDF_779
S4R_Bbw_2	An adequate redundant topology of the ETB shall be granted to guarantee availability of the network in order that a single fault does not stop the train in a no-stop area.	S4R_FDF_508
S4R_Bbw_3	An adequate redundant topology of the ECN shall be granted to guarantee availability of the network in order that a single fault does not stop the train in a no-stop area.	S4R_FDF_508
S4R_Bbw_4	An adequate redundancy of the VCU shall be granted to guarantee availability in order that a single fault does not stop the train in a no-stop area. In case of fault of the active VCU, the silent one shall handover in a transparent manner for all the controllers in the network.	S4R_FDF_508
S4R_Bbw_5	IMP shall host the emergency brake application (on both VCU and RBCU) guaranteeing a safety integrity level SIL4. The target THR for: • the VCU shall be fully covered by IMP; • the RBCU shall be covered by IMP just for what concerns internal logic controller diagnostics.	S4R_FDF_165
S4R_Bbw_6	IMP shall provide a service for the configuration management of any controller composing the brake system (VCU, RBCU). The minimum set of the required functionalities are: • receive the configuration by network and/or file • permanently store the configuration and inhibit any modification during service	S4R_FDF_612

Id	WP4 BbW requirements	FDF requirements
	• retrieve the configuration properties • assure validity of the configuration • assure integrity of the configuration • assure coherence between the local configuration and the received one	
S4R_Bbw_7	IMP shall provide a service for the application function execution of any controller composing the brake system (VCU, RBCU). The minimum set of the required functionalities are: • register a new process to be executed • specify the execution period of a process • specify the execution time of a process • specify if the execution is sequential or concurrent • allow real-time execution • assure spatial isolation: no process in an "isolation group" within a given time budget cannot be affected by the actions of a process from another "isolation group" • assure temporal isolation: no process in an "isolation group" can modify software code or application data or manage the I/O assigned to another "isolation group" • add a process to a defined "isolation group"	S4R_FDF_166
S4R_Bbw_8	IMP shall provide a service for the health management of any controller composing the brake system (VCU, RBCU). The minimum set of the required functionalities are: • perform integrity checks on HW (RAM, Flash, ADC, CPU temperature, etc.) • perform checks on function execution (order, period, execution time, temporal/spatial isolation) • notify a fault condition to all the application functions involved • support configurable recovery actions in case of a process deviates from normal behaviour	S4R_FDF_548
S4R_Bbw_9	IMP shall provide a service for the I/O management of any controller composing the brake system that needs to control input/output lines (RBCU). The minimum set of the required functionalities are: • set the value of a output (analog, digital) from a variable • read the value of an input (analog, digital) into a variable	S4R_FDF_169
S4R_Bbw_10	IMP shall provide a service for the redundancy management of any controller composing the brake system that needs to have multiple instances for availability (VCU). The minimum set of the required functionalities are:	S4R_FDF_689, S4R_FDF_690, S4R_FDF_691, S4R_FDF_508,

Id	WP4 BbW requirements	FDF requirements
	• register stand-by instance(s) of an application running on a different controller • cross-monitoring between the instances to detect a fault of the active • hot-swap between instances in case of fault of the active: the stand-by becomes active in a transparent manner (instances continuously have the same process image)	S4R_FDF_511, S4R_FDF_510, S4R_FDF_686, S4R_FDF_688, S4R_FDF_693, S4R_FDF_709
S4R_Bbw_11	IMP shall provide a service for the network management of any controller composing the brake system (VCU, RBCU). The minimum set of the required functionalities are: • implement a safety layer (timeliness, integrity, authenticity and validity of messages) • send messages from a messages storage to the network • send messages within defined timely bounds and with defined periodicity • receive messages from the network into a messages storage • receive messages within defined maximum delay (deterministic and reliable communication) • notify a fault condition to all the application functions involved • be abstracted from underlying network topology	S4R_FDF_167
S4R_Bbw_12	IMP shall provide a service for the network messages management of any controller composing the brake system (VCU, RBCU). The minimum set of the required functionalities are: • extract a variable value from a message in the messages storage • compose a message into the messages storage from a set of variables	S4R_FDF_167
S4R_Bbw_13	IMP shall provide a service for the variable management of any controller composing the brake system (VCU, RBCU). The minimum set of the required functionalities are: • create a variable into the memory specifying its type and size • set the value of a variable • get the value of a variable • handle the variable freshness (e.g. timestamp of the last update) • allow concurrent access to the variable	S4R_FDF_167
S4R_Bbw_14	IMP shall provide a service for the topology management of any controller composing the brake system (VCU, RBCU). The minimum set of the required functionalities are: • support inauguration • univocally identify a device in the network during inauguration • check the validity of the topology after inauguration	S4R_FDF_495

Id	WP4 BbW requirements	FDF requirements
	• check for topology changes at runtime (missing devices, new devices)	
S4R_Bbw_15	IMP shall provide network access with the following performances: • defined maximum delay (deterministic and reliable communication) • 1ms maximum latency on ECN • 10ms maximum latency on ETB	S4R_FDF_167

Table 7: FDF BbW requirements - Traceability matrix.

Annex E – Drive-by-Data: Traceability Matrix

This section contains Drive-by-Data requirements, defined in WP1, which need to be addressed to the Functional Distribution Framework. The table shows the coverage of the Drive-by-Data requirements, with identifier “S4R-IMP”, with the FDF requirements, with identifier “S4R_FDF”.

Id	WP1 requirements	FDF requirements
S4R-IMP-600	FDF shall define time budgets (maximum allowable task schedule execution time) for all middleware operations based on FDF configuration data.	S4R_FDF_233
S4R-IMP-601	FDF configuration and services shall organize the execution of different operations into task schedules (task chains) with defined maximum execution time.	S4R_FDF_233
S4R-IMP-602	FDF shall monitor task schedule (task chain) execution and performance.	S4R_FDF_233
S4R-IMP-603	FDF task scheduling service shall support execution of task schedules unaffected by other less critical task-chains.	S4R_FDF_233
S4R-IMP-604	FDF shall report any deviation from configured task chain performance to the FDF health monitoring service.	S4R_FDF_233
S4R-IMP-605	FDF shall provide task schedule start and stop service.	S4R_FDF_233
S4R-IMP-606	FDF task chain within a partition (or application container) will contain one or more applications of the same criticality.	S4R_FDF_233
S4R-IMP-607	FDF shall assess port health status on each read/write access.	S4R_FDF_233
S4R-IMP-608	FDF shall support hard OSEK synchronization in non-partitioning RTOS environments.	S4R_FDF_233
S4R-IMP-609	FDF shall define time budget for time-critical application task execution.	S4R_FDF_233

Id	WP1 requirements	FDF requirements
S4R-IMP-610	FDF shall define time budget for safety layer processing task tied to time-critical application	S4R_FDF_233
S4R-IMP-611	FDF shall define task time budget and allowable starting time window for IO operations.	S4R_FDF_233
S4R-IMP-612	FDF shall define time budget for dataset creation task (assembling/disassembling) using updated application variables (e.g. process data, local sensor/time data, etc.), tied to time-critical application.	S4R_FDF_233
S4R-IMP-613	FDF shall define time budget for communication middleware tasks used in the dataset transfer to the COM layer, for datasets tied to time-critical application.	S4R_FDF_233
S4R-IMP-614	FDF time budget will be either a part of the partition period (including application and IO tasks), or a part of the dedicated partition for dataset or IO handling.	S4R_FDF_233
S4R-IMP-615	FDF shall transfer IO or IPC data to the network COM layer within the time budget determined by the sum of all time budgets for all tasks within the subset of the task chain, configured for a related time-critical application.	S4R_FDF_233
S4R-IMP-616	FDF shall transfer datasets within a time period required for the timely transfer over the COM layer and network card, for time-critical application.	S4R_FDF_233
S4R-IMP-617	FDF will timestamp datasets in transition and measure their freshness (i.e. compliance with configured time budget).	S4R_FDF_233
S4R-IMP-618	Datasets with freshness violation will be reported to FDF health monitoring.	S4R_FDF_233
S4R-IMP-619	FDF shall indicate violation of partial time budgets (e.g. safety layer, dataset creation, communication middleware) as soon as they appear.	S4R_FDF_233
S4R-IMP-620	FDF shall drop datasets with freshness violation, which arrive too late at COM layer.	S4R_FDF_233
S4R-IMP-621	FDF shall include a margin for estimated WCET time budgets to the COM layer.	S4R_FDF_233
S4R-IMP-622	The margin for task chain execution shall cover any planned IRQ processing and background operation by the RTOS or other services.	S4R_FDF_233
S4R-IMP-623	FDF shall implement integrity measures for periodic configuration checking and fault detection.	S4R_FDF_233
S4R-IMP-624	FDF shall check the order and time instant of task execution and pre-emption.	S4R_FDF_233
S4R-IMP-625	FDF shall support option for defining allowable task initiation window (offset), in relation to local partition time.	S4R_FDF_233
S4R-IMP-626	FDF may provide option to log task chain execution and performance.	S4R_FDF_233
S4R-IMP-627	FDF shall support monitoring per task (time budget, offset, task start window) and per total task schedule (task chain).	S4R_FDF_233
S4R-IMP-628	FDF configuration may support definition of several task schedules.	S4R_FDF_233
S4R-IMP-629	FDF shall provide task schedule synchronous initiation function.	S4R_FDF_233
S4R-IMP-630	FDF shall provide task schedule asynchronous initiation function.	S4R_FDF_233
S4R-IMP-631	FDF shall support task schedule initiation relative to the current time instant.	S4R_FDF_233

Id	WP1 requirements	FDF requirements
S4R-IMP-632	FDF shall support task schedule initiation in absolute time.	S4R_FDF_233
S4R-IMP-633	The FDF configuration for a task schedule lists shall contain at least the following information: • Task ScheduleHeader ○ TaskScheduleID ○ InitialOffset (schedule offset to be added to each task offset) ○ FinalDelay (time allowed for execution of the last task) ○ RelativeTimeAllowed (if the task list can be executed relative to the past period, or only in absolute terms) ○ OneShotMode (One shot or continuous mode) ○ TaskIDListPointer ○ TaskScheduleCRC (with implicit PartitionID, DeviceID, NetworkConfigID) *Note: to ensure mutual configuration compliance* • TaskIDList with: ○ Task ID ○ TaskOffset (in AUTOSAR “expiry point” - The offset on a Schedule Table, measured from zero timer, at which the OS activates tasks or events) ○ TaskWindowStart (minimum offset relative to Task activation time) ○ TaskWindowEnd (maximum offset relative to Task activation time) ○ TaskTimeBudget (maximum task duration) ○ NextTaskPointer	S4R_FDF_233
S4R-IMP-634	FDF shall ensure that each task and task list configuration is compliant with device ID, schedule ID and partition ID.	S4R_FDF_233
S4R-IMP-635	FDF shall conduct power-up verification and plausibility check of the task chain in each partition before getting into synchronous state.	S4R_FDF_233
S4R-IMP-636	FDF shall conduct periodic checks of the FDF configuration data.	S4R_FDF_233
S4R-IMP-637	IMP CMS shall be able to operate statically and/or dynamically.	S4R_FDF_233

Table 8: FDF DbD requirements - Traceability matrix.

Annex F – Safe4RAIL WP2-CONNECTA T4.4: Traceability Matrix

Since the definition of the FDF requirements by CONNECTA project was behind Safe4RAIL's schedule (CONNECTA T4.4), Safe4RAIL defined their own FDF requirements. This table shows how CONNECTA's requirements are covered by proposed FDF requirements.

Those requirements which are not linked, denoted by "n/a" are indeed not derived. They were created based on the state of the art and other activities such as: directives and standards and TCMS user needs. Safety related countermeasures are described in Annex B.

Id	FDF requirement description	CONNECTA requirements
S4R_FDF_165	**Functional requirements**	n/a
S4R_FDF_166	**Application execution**	n/a
S4R_FDF_584	Execution management This subsection defines the execution management which is in charge of handling the execution of application functions and executable instances.	n/a
S4R_FDF_585	The Framework shall support authentication and authorisation of executables at start-up.	CTA-D4.4-EM-1
S4R_FDF_586	The Framework shall check the integrity of executables at start-up.	CTA-D4.4-EM-2
S4R_FDF_737	The Framework shall inhibit the execution of the application function in the case of negative code integrity check.	n/a
S4R_FDF_766	The Framework shall avoid forcing outputs when application function is operative (nominal and degraded).	n/a
S4R_FDF_767	The Framework shall prevent the access of off-line services at power-up, during initialization and operation (nominal and degraded).	n/a
S4R_FDF_768	The Framework shall guarantee the retention of a safe-state after a fatal fault.	n/a
S4R_FDF_782	The Framework shall be able to generate partitions and allocate resources for application functions requiring multiple-instances for the implementation of reliable and safe architecture.	n/a
S4R_FDF_588	The Framework shall support multiple executable instances.	CTA-D4.4-EM-4
S4R_FDF_589	The Framework shall consider unambiguous identification of executable instances (i.e., processes) provided by the	CTA-D4.4-EM-5

Id	FDF requirement description	CONNECTA requirements
	configuration.	
S4R_FDF_594	The Framework shall support ordered execution of processes, partitions and FDF components.	CTA-D4.4-EM-12
S4R_FDF_684	The Framework shall guarantee a pre-emptive and priority based schedule for concurrent execution.	CTA-D4.4-EM-32
S4R_FDF_686	The Framework shall manage redundant execution of partitions and/or processes on different devices.	CTA-D4.4-EM-21
S4R_FDF_692	The Framework shall provide a mechanism for service discovery and announcement.	CTA-D4.4-EM-35
S4R_FDF_687	The Framework shall support configurable recovery actions in case of partition or process deviations from normal behaviour.	CTA-D4.4-EM-17
S4R_FDF_688	The Framework shall provide internal variables as outputs and the 'leader" shall update those outputs after each redundant execution of partitions or processes. The internal variables are persistent over more than a single execution of the partition or process.	CTA-D4.4-EM-24
S4R_FDF_693	The Framework shall provide internal variables as the input to synchronise the internal variables of a "follower" with the variables provided by the "leader" before each execution of partitions or processes. The internal variables are persistent over more than a single execution of the partition or process.	CTA-D4.4-EM-25
S4R_FDF_694	The Framework shall interface with the Monitoring Manager in a secure way, by offering authentication measures for instance, to provide the availability of forcing variables.	CTA-D4.4-EM-36
S4R_FDF_695	The Framework shall be able to suspend the execution of processes and/or partitions during a given time.	CTA-D4.4-EM-39
S4R_FDF_741	The Framework shall load the configuration file during inauguration.	n/a
S4R_FDF_755	The Framework shall guarantee calls to service functions with the same SIL assigned to the application functions using services.	n/a
S4R_FDF_783	The Framework shall guarantee spatial separation between memory spaces containing read-only and read-write variables, variables with different SIL, variables used by multiple independent instances, software code and parameters of the application function.	n/a
S4R_FDF_185	**Process management** This subsection defines a process and describes how the IMP (Integrated Modular Platform) interacts with each of these. The ECP (Extended Capabilities Port) shall offer services to create and manage timers for sequential execution and semaphores for sequential and concurrent execution.	n/a
S4R_FDF_506	A process shall be in the state: • Suspended: The process is not permitted to be activated. • Waiting: The process is waiting for its activation, which depending on the triggering paradigm will be when	CTA-D4.4-EM-38

Id	FDF requirement description	CONNECTA requirements
	the corresponding event is launched or it is a certain instant of time. • Ready: The process is ready to execute and will do it if it has the highest priority among the ready processes. • Running: The process is executing in the processor.	
S4R_FDF_193	The Framework shall activate a time-triggered process in waiting state if: • The current time is inside its partition time slot • The current time is a multiple of its period	CTA-D4.4-EM-39 CTA-D4.4-EM-40
S4R_FDF_498	The Framework shall grant spatial separation among processes.	n/a
S4R_FDF_194	The Framework shall execute the process in ready state with the highest priority.	n/a
S4R_FDF_195	The Framework shall set the state of a process to waiting when its execution finishes.	CTA-D4.4-EM-38
S4R_FDF_196	The Framework shall launch the finishing event of a process when its execution finishes.	n/a
S4R_FDF_197	The Framework shall set a process in ready state to waiting if it waits for an event.	n/a
S4R_FDF_610	The processes shall be configured according to a configuration file.	CTA-D4.4-EM-33
S4R_FDF_497	The Framework shall execute processes sequentially or concurrently.	CTA-D4.4-EM-29 CTA-D4.4-EM-31
S4R_FDF_698	The Framework shall limit the execution time for each process.	CTA-D4.4-EM-40
S4R_FDF_519	The Framework shall control the execution of processes with the same SIL assigned to the involved application functions.	n/a
S4R_FDF_587	The Framework shall set-up separate process to execute each instance.	CTA-D4.4-EM-3
S4R_FDF_520	The Framework shall implement service functions whose response times allow the real-time execution of processes and implement mechanisms to ensure that execution.	n/a
S4R_FDF_521	The Framework shall monitor execution of processes concerning defined time-bounds for communication and processing.	CTA-D4.4-EM-41
S4R_FDF_604	The Framework shall support configurable recovery actions in case of a process deviates from normal behaviour.	CTA-D4.4-EM-17
S4R_FDF_522	The Framework shall notify a fault condition in case of error in the process execution.	n/a
S4R_FDF_523	A process can belong to different process schedules.	n/a
S4R_FDF_700	The Framework shall allow to processes to set and get the current FDF's operation mode.	CTA-D4.4-EM-41

Id	FDF requirement description	CONNECTA requirements
S4R_FDF_199	**Partition management** Partitions give means to guarantee memory space separation, which might contain all the information of processes. Besides, a cyclic executive scheduler must give and take away access to the processor when corresponds.	CTA-D4.4-EM-31
S4R_FDF_205	A partition shall be active or inactive.	n/a
S4R_FDF_206	Only active partitions shall be executed.	n/a
S4R_FDF_208	The Framework shall guarantee temporal separation among partitions by ensuring that a process within a given time budget cannot be affected by the actions of any other tasks of any other partition.	CTA-D4.4-EM-7 CTA-D4.4-EM-14
S4R_FDF_592	The Framework shall bind the period, and execution time for each partition.	CTA-D4.4-EM-10 CTA-D4.4-EM-11
S4R_FDF_685	The Framework shall ensure the independence (time, space) of services and to support partitions' independence.	CTA-D4.4-EM-34
S4R_FDF_759	The Framework shall manage interrupts through the O.S, to avoid any disturbance to time partitioning.	n/a
S4R_FDF_606	The Framework shall support synchronised execution of partitions among different processor cores and devices.	CTA-D4.4-EM-18
S4R_FDF_607	The Framework shall write/update the inputs of each partition before executing them.	CTA-D4.4-EM-19
S4R_FDF_609	The Framework shall write/update the outputs of each partition after executing them.	CTA-D4.4-EM-20
S4R_FDF_689	The Framework shall execute and write/update the outputs, when the partition has the redundancy role "leader".	CTA-D4.4-EM-22
S4R_FDF_690	The Framework shall execute the partition, but shall not write/update its outputs, when the partition has the redundancy role "follower".	CTA-D4.4-EM-23
S4R_FDF_691	The Framework shall activate one of the "follower" partitions in the case that the "leader" partition fails. The "follower" shall write the outputs of "leader" partition.	CTA-D4.4-EM-26
S4R_FDF_524	Partitions shall guarantee spatial separation to ensure that no process in one partition can modify without authorisation software code or application data of another partition. E.g., by means of memory protection mechanisms.	CTA-D4.4-EM-28 CTA-D4.4-EM-15 CTA-D4.4-EM-9
S4R_FDF_525	Partitions are configured according to the configuration file of the application functions to be executed.	CTA-D4.4-EM-33
S4R_FDF_527	A partition can belong to different partition schedules.	n/a
S4R_FDF_526	Partitions have assigned computational resources defined in configuration file. No resource is shared by partitions hosting application functions with different SIL.	n/a

Id	**FDF requirement description**	CONNECTA requirements
S4R_FDF_210	Partitions shall contain one or more processes.	CTA-D4.4-EM-28
S4R_FDF_507	A partition shall be in the state: • Suspended: The partition is not permitted to be activated. • Waiting: The partition is waiting for its activation, which depending on the triggering paradigm will be when the corresponding event is launched or it is a certain instant of time. • Ready: The partition is ready to execute and will do it if it has the highest priority among the ready partitions. • Running: The partition is executing in the processor. • Isolated: The partition is isolated and it is not permitted to be activated.	CTA-D4.4-EM-13
S4R_FDF_602	The Framework shall not execute partitions in state suspended or isolated.	CTA-D4.4-EM-16 CTA-D4.4-EM-15
S4R_FDF_603	The Framework shall support configurable recovery actions in case of a partition deviates from normal behaviour.	CTA-D4.4-EM-17
S4R_FDF_528	Partitions shall notify fault conditions in case of invalid operation in the partition attempt (fatal fault).	n/a
S4R_FDF_784	The Framework shall assign privileges for read-write access to a memory space only to independent application functions with at least the same SIL. Read-only access could be assigned to remaining application functions, if data alteration during reading can be excluded.	n/a
S4R_FDF_215	**Concurrency management** This subsection gives details regarding concurrency control and synchronisation techniques.	n/a
S4R_FDF_216	An event shall be active or inactive.	n/a
S4R_FDF_217	The Framework shall activate an event when it is commanded to launch.	CTA-D4.4-EM-30
S4R_FDF_590	The Framework shall support concurrent execution of more than one partition in different processor cores and/or devices.	CTA-D4.4-EM-6
S4R_FDF_529	Concurrent accesses to shared resources shall be synchronised using semaphores and/or mutexes.	CTA-D4.4-EM-32
S4R_FDF_530	Concurrent executions shall be synchronised using semaphores and/or mutexes.	n/a
S4R_FDF_612	**Configuration management** This subsection defines the requirements regarding the configuration the Functional Distribution Framework including settings for partitions and variables.	n/a

Id	FDF requirement description	CONNECTA requirements
S4R_FDF_613	The Framework shall statically identify an ECU instance at boot time (e.g., by local digital inputs)	CTA-D4.4-CFM-1
S4R_FDF_614	The Framework shall dynamically acquire the identification of ECUs instances at boot time (e.g., by DHCP).	CTA-D4.4-CFM-2
S4R_FDF_625	The Framework shall obtain the identifier of an ECU instance.	CTA-D4.4-CFM-13
S4R_FDF_742	The configuration and re-configuration of the Framework shall involve all the application functions.	n/a
S4R_FDF_615	The Framework shall acquire the following configuration parameters of the FDF and make them available to the FDF's components with the same SIL assigned to related application functions. • Version information • User identification and privileges • Contained devices • Contained partitions • Scheduling parameter of contained partitions • Contained communication networks	CTA-D4.4-CFM-3
S4R_FDF_616	The Framework shall acquire the following configuration parameters of a device and make them available according to the SIL assigned to related application functions. • Contained I/O units.	CTA-D4.4-CFM-4
S4R_FDF_618	The Framework shall acquire the consist network configuration of a given SIL and make it available to the FDF components with the same SIL.	CTA-D4.4-CFM-5
S4R_FDF_619	The Framework shall acquire the configuration parameters of partitions and make them available to the FDF components. • Unique identifier • Version information • Execution period • Maximum execution time • Redundancy role • In- and Output variables • Contained processes • Scheduling policy and dependencies of the contained processes • Error handling including recovery actions	CTA-D4.4-CFM-6
S4R_FDF_620	The Framework shall acquire the following configuration parameter set for a process FDF and make them available to	CTA-D4.4-CFM-7

Id	**FDF requirement description**	CONNECTA requirements
	the FDF components. • Unique identifier • Executable that is executed in the process • Mapping of input/output variables to variables provided by or send to other processes, network or I/O • Assigning rights for publishing/reading variables to the SW components. • Execution period • Maximum execution time • Redundancy role • Scheduling priority • Error handling including recovery actions • Access to FDF services (e.g. set global time)	
S4R_FDF_621	The Framework shall acquire the following configuration parameter set for an executable and make them available to the FDF components. • Unique identifier • Version information • In- and Output variables • Variables available for external monitoring • Variables stored persistently • Provided and required services	CTA-D4.4-CFM-8
S4R_FDF_622	The Framework shall acquire the following configuration parameter set for an IO unit and make them available to the FDF components. • Unique identifier • In- and Output variables • Decoder configuration for encoder signals • Update cycle of in- and output variables	CTA-D4.4-CFM-9
S4R_FDF_623	The Framework shall acquire the following configuration parameter set of a variable and make them available to the FDF components. • Unique identifier • Value interpretation • Default value	CTA-D4.4-CFM-10

Id	**FDF requirement description**	CONNECTA requirements
	- Data type	
S4R_FDF_624	The Framework shall acquire the configuration parameter set of a service and make them available to the FDF components. - Unique identifier	CTA-D4.4-CFM-11
S4R_FDF_626	The Framework shall acquire the following configuration parameter set for the event log and make them available to the FDF components. - maximum size - time period for storage of reoccurring events	CTA-D4.4-CFM-12
S4R_FDF_167	**Communication management** This subsection contains requirements related to communication management.	n/a
S4R_FDF_220	**Data and event distribution** This chapter contains requirements on event and ECU and application data distribution which is done by the use of distribution variables between processes.	n/a
S4R_FDF_221	The Framework shall provide services to create exchange variables, which are data structured consisting of a set of parameter and value pairs and should be SIL independent.	n/a
S4R_FDF_222	The variables shall be exchanged between software components using the publish-subscribe pattern. a) Communication is black channel (including the publish-subscribe pattern) b) Safety relevant process data must be encrypted c) Non-Safe process data may be encrypted d) Encryption credentials must be configured e) Public/private key encryption is not sufficient - there must be certificates exchanged to prevent 3rd party access to safety critical functions handled in 2.5 Security requirements Note: The publish–subscribe is a messaging pattern where senders of messages (publishers) do not directly send messages to specific receivers (subscribers) but instead characterise published messages into classes (e.g. certain variables) without knowledge of which subscribers, if any, there may be. Similarly, subscribers express interest in one or more classes and only receive messages that are of interest, without knowledge of which publishers, if any, there are.	n/a
S4R_FDF_223	The Framework shall give software components read and write access only to the variables they are allowed to publish.	CTA-D4.4-CM-10

Id	FDF requirement description	CONNECTA requirements
		CTA-D4.4-NM-8 CTA-D4.4-NM-9
S4R_FDF_224	The Framework shall give software components read access only to the variables they are subscribed to (without altering their value).	CTA-D4.4-CM-13
S4R_FDF_753	The Framework shall give software components write access according to specification set during configuration.	n/a
S4R_FDF_225	The Framework shall guarantee that the software component publishing a variable is able to update its value.	CTA-D4.4-CM-12 CTA-D4.4-CM-14
S4R_FDF_226	The Framework shall guarantee that an updated value is accessible for every software component that is subscribed to it within the defined timely bound.	CTA-D4.4-CM-12 CTA-D4.4-CM-14
S4R_FDF_735	The Framework shall guarantee the updating of input variables according to the values of input channels and SIL level.	n/a
S4R_FDF_227	The Framework shall guarantee that the communicating software components may exchange messages in the same way, regardless of the location of the software components, be it: • in the same process • in different processes of the same partition • in different partitions of the same ECU • or in different ECUs of the same network Especially in the case of different ECUs on the same network, security aspects shall be considered. (handled in 2.5 Security requirements)	n/a
S4R_FDF_493	The Framework shall provide services to exchange Message data (non-cyclic/best-effort) using a "notification", "call/reply" or "call/reply/confirm" pattern.	n/a
S4R_FDF_494	The Framework shall provide services to request data of variables non-cyclic/non-deterministic.	n/a
S4R_FDF_495	The Framework shall provide services to read out the TTDB (Train Topology Database) which is the result of inauguration.	CTA-D4.4-NM-7
S4R_FDF_541	The framework shall provide the ability to set default values to variables: • of the train and consist network and • shared between partitions of the same device • Shared between processes of the same partition according to configuration.	CTA-D4.4-CM-15
S4R_FDF_496	The Framework shall provide services to supervise the validity of the inauguration result.	n/a

Id	**FDF requirement description**	CONNECTA requirements
S4R_FDF_508	The Framework shall provide services to support different redundancy concepts.	n/a
S4R_FDF_709	The Framework shall be able to replicate the value of local input variables on the consist network according to configuration.	CTA-D4.4-CM-11
S4R_FDF_509	The Framework shall provide services to define a variable which can be then updated from different redundant devices.	n/a
S4R_FDF_510	The Framework shall provide services to define a set of redundant variables which are each updated by the corresponding redundant device.	n/a
S4R_FDF_511	The Framework shall mark the variables as valid or invalid according to the chosen redundancy concept. (E.g. one out of two, two out of three...)	n/a
S4R_FDF_543	The Framework shall provide the ability to create and manage access to shared memories to facilitate communication between processes of the same partition.	CTA-D4.4-CM-16
S4R_FDF_780	The Framework shall guarantee the validity of safety-related data exchange between remote functions through messages composing and decomposing into variables out with the same SIL assigned to the application functions.	n/a
S4R_FDF_781	The Framework shall allow message function to access to memory spaces containing messages and variables with the same SIL.	n/a
S4R_FDF_785	The Framework shall guarantee the read-write access to memory spaces (according to the assigned privileges) with the same SIL assigned to the Application function(s) and variables stored.	n/a
S4R_FDF_786	The Framework shall execute an Application function, giving access to memory resources, only when required by its scheduling plan (and take away access otherwise).	
S4R_FDF_228	**Networking** Networking comprises requirements on location transparency, whether a Publish-Subscribe pattern is used and the number of participants or the support of deterministic real-time and best-effort messages.	n/a
S4R_FDF_229	The Framework shall provide communication mechanisms that are abstracted of the physical realisation of the communication hardware.	n/a
S4R_FDF_230	The Framework shall provide a standardised software interface (Ethernet) for communication between software components ensuring their communication independent whether they are located • on the same ECU on the same core • on the same ECU on another core • on the same ECU on another microcontroller on another ECU	CTA-D4.4-NM-1 CTA-D4.4-NM-2
S4R_FDF_711	The Framework shall provide an IEC 61375-2-3 compliant safety layer for the consist network communication.	CTA-D4.4-NM-3

Id	FDF requirement description	CONNECTA requirements
S4R_FDF_231	The Framework shall provide a communication service that allows it to send messages (containing variables) to other components on the network within defined timely bounds from the point in time where the application sends the message to the point in time it is sent on the network (deterministic sending).	n/a
S4R_FDF_756	The Framework shall instantiate messages according to the configuration file, including: • Unique identifier (ID) • Messages to be received or send • List of variables linked to messages • Messages schedule • Deadline	n/a
S4R_FDF_750	The Framework shall periodically send messages within defined time bounds and receive them within defined maximum delay (deterministic sending).	n/a
S4R_FDF_512	The Framework shall provide a communication service which provides a deterministic way for an application to announce/prepare a message/data value for deterministic sending.	n/a
S4R_FDF_232	The Framework shall provide a communication service that makes received messages from other components on the network available to the application within defined timely bounds (deterministic receiving).	n/a
S4R_FDF_513	The Framework shall provide a communication service which provides a deterministic way to fetch a message/data value after deterministic reception.	n/a
S4R_FDF_751	The Framework shall implement Communication service without any operation on the messages' safety layer content.	
S4R_FDF_233	**System integration** This chapter contains requirements regarding the COM layer, the inauguration process or transport layer protocols among others. System Integration Requirements are covered in detail in D1.11.	n/a
S4R_FDF_168	**Time management** Different ECUs share a unique global time that is synchronised with UTC. These requirements contain details regarding interfaces used, protocols and ways of synchronisation, i.e., automatic or manual.	n/a
S4R_FDF_235	The Framework shall provide a service for starting application processes based on the progression of time.	n/a
S4R_FDF_236	The Framework shall synchronise the local computer clock with the external global clock source and keep it synchronised with a maximum deviation of the global clock source of 1 microsecond.	CTA-D4.4-TM-1

Id	FDF requirement description	CONNECTA requirements
S4R_FDF_762	The Framework shall synchronize the local clock independently from the execution of different partition's processes.	n/a
S4R_FDF_237	The Framework shall allow process and partition execution to be scheduled at a configured time instant within a configured rate-monotonic execution cycle period.	n/a
S4R_FDF_238	The Framework shall check and inform about successful synchronisation, synchronisation state and synchronisation errors.	n/a
S4R_FDF_544	The Framework shall provide allow processes to set the global time if allowed by configuration to do so.	CTA-D4.4-TM-2
S4R_FDF_545	The Framework shall provide the ability to processes to create, configure and delete timers.	CTA-D4.4-TM-4
S4R_FDF_239	The global time shall be made available to all ECUs through the network layer.	CTA-D4.4-TM-5
S4R_FDF_240	Global time dissemination shall be fault tolerant. Note: In case no time synchronisation is available, there is no scheduled (critical) communication possible. In case of erroneous time synchronisation, messages may arrive early or late and can lead to catastrophic events. This erroneous time synchronization must be detected by the SDT layer.	n/a
S4R_FDF_736	The Framework shall not finalize the inauguration without a valid global-time.	n/a
S4R_FDF_169	**Input/output management**	n/a
S4R_FDF_261	**Input management** This subsection contains requirements specifying which Input devices the ECU must be able to work with and how the data of these devices should be read and interpreted.	CTA-D4.4-IO-1 CTA-D4.4-IO-2
S4R_FDF_263	The Framework shall provide a service to create the controller access to an analog input.	n/a
S4R_FDF_264	The Framework shall provide a service to create the controller access to a digital input.	n/a
S4R_FDF_265	The inputs shall be accessible over configurable symbolic names.	n/a
S4R_FDF_764	The Framework shall allow input functions to access only to memory spaces with the same SIL.	n/a
S4R_FDF_266	The Framework shall create an exchange variable associated with each input channel.	n/a
S4R_FDF_546	The Framework shall set default values to digital and analog input variables according to configuration, with the same SIL assigned to related application functions.	CTA-D4.4-IO-5
S4R_FDF_267	The exchange variable associated with an input channel shall contain the acquired input channel value.	n/a
S4R_FDF_268	The Framework shall store the current value of every used input at the end of each acquisition cycle in the associated	n/a

Id	FDF requirement description	CONNECTA requirements
	exchange variable.	
S4R_FDF_269	The Framework shall provide a service for reading the last valid value of every used input, stored in the associated exchange variable.	n/a
S4R_FDF_270	The service for reading the value of every used input stored in the associated exchange variable shall not be interruptible to ensure data consistency.	n/a
S4R_FDF_716	The Framework shall decode encoder signals and transfer the value into a variable, including validity information.	CTA-D4.4-IO-7
S4R_FDF_262	**Output management** Analogously, this other subsection contains requirements specifying which Output devices the ECU must be able to work with and how the data of these devices should be written and interpreted.	CTA-D4.4-IO-3 CTA-D4.4-IO-4
S4R_FDF_271	The Framework shall provide a service to create the controller access to an analog output.	n/a
S4R_FDF_272	The Framework shall provide a service to create the controller access to a digital output.	n/a
S4R_FDF_273	The outputs shall be accessible over configurable symbolic names.	n/a
S4R_FDF_765	The Framework shall allow output functions to access only to memory spaces with the same SIL.	n/a
S4R_FDF_274	The Framework shall create an exchange variable associated with each output channel.	n/a
S4R_FDF_547	The Framework shall set digital and analog outputs to default values according to configuration, with the same SIL assigned to related application functions.	CTA-D4.4-IO-6
S4R_FDF_275	The exchange variable associated with an output channel shall contain the output channel set value.	n/a
S4R_FDF_276	The Framework shall provide a service for writing a new value and update it in the associated exchange variable of every used output.	n/a
S4R_FDF_277	The service for writing a new value in the associated exchange variable of every used output shall not be interruptible to assure data consistency.	n/a
S4R_FDF_548	**Health management**	n/a
S4R_FDF_549	The Framework shall support CPU, board and/or rack temperature monitoring, if supported by the HW monitoring.	CTA-D4.4-HM-1 CTA-D4.4-HM-2
S4R_FDF_551	The Framework shall support checking if partitions are executed within their maximum execution time.	CTA-D4.4-HM-3
S4R_FDF_552	The Framework shall support a HW watchdog timer (WDT).	CTA-D4.4-HM-4

Id	FDF requirement description	CONNECTA requirements
S4R_FDF_553	The Framework shall refresh the WDT.	CTA-D4.4-HM-5
S4R_FDF_554	The Framework shall support integrity checks of the HW.	CTA-D4.4-HM-6
S4R_FDF_555	The Framework shall support check if partitions and processes update their outputs according to the value of variables and SIL level.	CTA-D4.4-HM-7
S4R_FDF_557	The Framework shall log the errors detected in a log file.	CTA-D4.4-HM-9
S4R_FDF_728	The Framework shall check the timeliness and sequence of messages exchanged between remote functions.	n/a
S4R_FDF_556	The Framework shall provide reaction to errors when a partition or process: • does not write the output • does not terminate execution in time • CPU, board and/or rack temperature exceeds the allowed range • CPU, board and/or rack load too high	CTA-D4.4-HM-8
S4R_FDF_558	The Framework shall consider the following reaction to error mechanisms with the highest SIL assigned to the application functions, without disturbing to other framework's services: • restart the ECU of the affected partition/process (without affecting other ECUs) • restart the affected partition/process (without affecting other partitions/processes) • isolate/terminate the affected partition/process (without affecting other partitions/processes) • inform the application function and continue with normal operation	CTA-D4.4-HM-9
S4R_FDF_746	The Framework shall provide reaction to errors when a communication error is identified: • message authenticity • message integrity • message timeliness • message sequence	n/a
S4R_FDF_745	The Framework notifies to application function and reacts against safety-related communication errors, for example, discarding erroneous messages.	n/a
S4R_FDF_754	The Framework shall detect and notify the application SW in case of unavailability of scheduled services or in case of incorrect calls (different schedules).	n/a
S4R_FDF_758	The Framework shall notify fault conditions to all the Application function(s) involved (with SIL) without disturbing to other framework's services and no later than the maximum time for safe state.	n/a
S4R_FDF_769	The Framework shall notify a fault condition to the related application function in case of inconsistencies between the	n/a

Id	**FDF requirement description**	CONNECTA requirements
	values stored into an exchange variable and the status of the platform's input/output.	
S4R_FDF_179	**Monitoring management**	n/a
S4R_FDF_562	The Framework shall allow remotely requesting the list of available variables.	CTA-D4.4-MO-3
S4R_FDF_563	The Framework shall allow remotely registering variables that can be monitored.	CTA-D4.4-MO-2
S4R_FDF_564	The Framework shall send the list of variable that can be monitored to external device.	CTA-D4.4-MO-3 CTA-D4.4-SM-4
S4R_FDF_355	The Framework shall allow remotely reading the variables of a component.	CTA-D4.4-MO-6
S4R_FDF_356	The Framework shall allow remotely writing the variables of a component.	n/a
S4R_FDF_357	The Framework shall allow remotely reading the events of a component.	n/a
S4R_FDF_358	The Framework shall allow remotely writing the events of a component.	n/a
S4R_FDF_359	The Framework shall allow remotely forcing the variables of a component.	CTA-D4.4-SM-5
S4R_FDF_361	The Framework shall allow remotely unforcing the variables of a component.	n/a
S4R_FDF_362	The Framework shall allow remotely forcing the events of a component.	CTA-D4.4-SM-5
S4R_FDF_363	The Framework shall allow remotely unforcing the events of a component.	n/a
S4R_FDF_364	The Framework shall check the state of all existing processes.	n/a
S4R_FDF_365	The Framework shall check the value of all framework variables, comparing them with the I/O values.	n/a
S4R_FDF_704	The Framework shall guarantee a secure communication with external devices.	CTA-D4.4-MO-4 CTA-D4.4-SM-3
S4R_FDF_733	The Framework shall provide services to monitor variables (e.g., remotely (out of FDF)).	n/a
S4R_FDF_761	The Framework shall detect faults with the highest SIL assigned to the application functions to be executed, without disturbing to other framework's services.	n/a
S4R_FDF_738	The Framework shall detect resource-related faults at power-up and periodically.	n/a
S4R_FDF_743	The Framework shall detect incoherence of configuration file.	n/a
S4R_FDF_744	The Framework shall detect the lack of configuration file's integrity.	n/a
S4R_FDF_752	The Framework shall assign to the monitoring-function RO privileges to variables stored into memory spaces with lowest integrity level or to all the memory spaces with different integrity levels (SIL) without altering the execution of other	n/a

Id	FDF requirement description	CONNECTA requirements
	services.	
S4R_FDF_760	The Framework shall monitor the alignment with the external global clock with the highest SIL assigned to the application functions to be executed.	n/a
S4R_FDF_771	The Framework shall monitor that non-safety data uses different structures than ones used for safety-related data.	n/a
S4R_FDF_788	The Framework shall provide fault detection during run-time execution.	n/a
S4R_FDF_789	The Framework shall provide further measures and detection techniques, in addition to the techniques/measures provided, for run-time fault detection.	n/a
S4R_FDF_377	**Log management** This subsection describes which information the system log should include. This could be sensitive activity, errors or the state of the different processes.	n/a
S4R_FDF_378	The Framework shall create a log file per day (if applicable persistent log file).	CTA-D4.4-EL-1
S4R_FDF_574	The Framework shall configure the maximum size of the event log.	CTA-D4.4-EL-2
S4R_FDF_575	The Framework shall overwrite previously recorded event if the maximum of the log file size is reached.	CTA-D4.4-EL-3
S4R_FDF_576	The Framework shall only record one error every certain period of time, in case of recurrent errors. The logging period of time shall be configurable.	CTA-D4.4-EL-4
S4R_FDF_380	The Framework shall log the minimum execution time of the processes per hour.	CTA-D4.4-EL-5
S4R_FDF_381	The Framework shall log the maximum execution time of the processes per hour.	CTA-D4.4-EL-5
S4R_FDF_382	The Framework shall log the average execution time of the processes per hour.	CTA-D4.4-EL-5
S4R_FDF_383	The Framework shall log if any of its processes does not meet its deadline.	CTA-D4.4-EL-5
S4R_FDF_384	The Framework shall log if the integrity of the memory space of a partition has an error.	CTA-D4.4-EL-5
S4R_FDF_385	The Framework shall log if the integrity of the configuration file of the Framework has an error.	CTA-D4.4-EL-5
S4R_FDF_386	The Framework shall log if the coherency of the configuration file of the Framework has an error.	CTA-D4.4-EL-5
S4R_FDF_387	The Framework shall log if any unexpected external access is detected.	CTA-D4.4-EL-4
S4R_FDF_388	The Framework shall log if any not allowed external access is detected.	CTA-D4.4-EL-5
S4R_FDF_379	The log file shall follow the “report_yyyymmdd_xxx.log” naming convention, where yyyy, mm and dd stand for the system year, month and day and the xxx represents an incremental value in case more than one file with the same date exists.	n/a

Id	FDF requirement description	CONNECTA requirements
S4R_FDF_389	The Framework must make a back up of the log files every day.	n/a
S4R_FDF_390	The Framework shall include a timestamp for each entry of the log file.	n/a
S4R_FDF_580	The Framework shall provide the application with the ability to add an entry in the event log.	CTA-D4.4-EL-6
S4R_FDF_581	The Framework shall allow the application to use the following logging levels for an entry: f) Debug g) Info h) Warning i) Error j) Fatal	CTA-D4.4-EL-7
S4R_FDF_582	The Framework shall provide the ability to export the current event log as a file with the following information per event log entry: • Identification of triggering entity • Type (logging level) • Event ID • Event message • Raw data	CTA-D4.4-EL-8
S4R_FDF_565	**Deployment management** This subsection describes the requirements of the deployment management that enables to install and update configuration files and application executables of FDF partitions.	n/a
S4R_FDF_571	The Framework shall implement a secure file transfer such as FTPS or SFTP transfer protocols.	n/a
S4R_FDF_666	The Framework shall support debug operation and maintenance operation modes.	CTA-D4.4-DM-30
S4R_FDF_770	The Framework shall support maintenance of non-safety data using different structures than ones used for safety-related data.	n/a
S4R_FDF_567	The Framework shall provide maintenance staff with the ability to install executables on partitions train network, remote and direct connections.	CTA-D4.4-DM-1 CTA-D4.4-DM-3 CTA-D4.4-DM-2
S4R_FDF_566	The Framework shall provide maintenance staff with the ability to update executables on partitions train network, remote and direct connections.	CTA-D4.4-DM-4 CTA-D4.4-DM-6 CTA-D4.4-DM-5

Id	FDF requirement description	CONNECTA requirements
S4R_FDF_573	The Framework shall provide maintenance staff with the ability to uninstall executables on partitions through train network, remote and direct connections.	CTA-D4.4-DM-7 CTA-D4.4-DM-9
S4R_FDF_568	The Framework shall provide maintenance staff with the ability to install configuration files through train network, remote and direct connections.	CTA-D4.4-DM-18 CTA-D4.4-DM-19 CTA-D4.4-DM-20 CTA-D4.4-DM-31
S4R_FDF_569	The Framework shall provide maintenance staff with the ability to update configuration files train network, remote and direct connections.	CTA-D4.4-DM-21 CTA-D4.4-DM-22 CTA-D4.4-DM-23 CTA-D4.4-DM-31
S4R_FDF_570	The Framework shall provide maintenance staff with the ability to uninstall configuration files train network, remote and direct connections.	CTA-D4.4-DM-8 CTA-D4.4-DM-24 CTA-D4.4-DM-25 CTA-D4.4-DM-26 CTA-D4.4-DM-31
S4R_FDF_635	The Framework shall provide the maintenance staff with a secure way to install executables on a partition.	CTA-D4.4-DM-12
S4R_FDF_639	The Framework shall provide the maintenance staff with a secure way to update executables on a partition.	CTA-D4.4-DM-12
S4R_FDF_640	The Framework shall provide the maintenance staff with a secure way to uninstall executables on a partition.	CTA-D4.4-DM-12
S4R_FDF_660	The Framework shall provide the maintenance staff with a secure way to install configuration files.	CTA-D4.4-DM-28
S4R_FDF_661	The Framework shall provide the maintenance staff with a secure way to update configuration files.	CTA-D4.4-DM-28
S4R_FDF_662	The Framework shall provide the maintenance staff with a secure way to uninstall configuration files	CTA-D4.4-DM-28
S4R_FDF_636	The Framework shall allow deleting persistently stored data and files with uninstalled executables.	CTA-D4.4-DM-16
S4R_FDF_658	The Framework shall provide detailed version information of FDF to maintenance staff.	CTA-D4.4-DM-10
S4R_FDF_663	The Framework shall provide detailed version information of each process (installed executable) to the maintenance staff.	CTA-D4.4-DM-11
S4R_FDF_665	The Framework shall provide detailed version information of each configuration file to the maintenance staff.	CTA-D4.4-DM-27
S4R_FDF_659	The Framework shall validate the executable code, schedule and the resource availability before the installation, during the installation and during updating it.	CTA-D4.4-DM-17

Id	FDF requirement description	CONNECTA requirements
S4R_FDF_664	The Framework shall validate the configuration file before processing it or updating it to ensure that there is not conflict in the communication, schedule or resource availability of partitions and processes.	CTA-D4.4-DM-29
S4R_FDF_787	The Framework shall support concurrent re-configuration of partitions, guaranteeing that the re-configuration does not affect the remaining partitions. Those partitions may execute different and independent application functions with the same SIL level and to be hosted by one partition.	n/a
S4R_FDF_641	**File management** This subsection writes and reads files and variables that persist over device switch on and switch off cycles.	n/a
S4R_FDF_644	The Framework shall enable to create new files in memory.	CTA-D4.4-PS-1
S4R_FDF_645	The Framework shall allow opening existing files.	CTA-D4.4-PS-2
S4R_FDF_648	The Framework shall allow opening files in read-only (RO) or read/write (RW) modes.	CTA-D4.4-PS-3
S4R_FDF_649	The Framework shall allow writing data into a file.	CTA-D4.4-PS-4
S4R_FDF_650	The Framework shall allow reading data from a file.	CTA-D4.4-PS-5
S4R_FDF_651	The Framework shall allow storing files persist over device switch-on and switch-off cycles.	CTA-D4.4-PS-7
S4R_FDF_652	The Framework shall enable to remove files.	CTA-D4.4-PS-8
S4R_FDF_653	The Framework shall enable to persistently store variables over device switch-on and switch-off cycles.	CTA-D4.4-PS-9
S4R_FDF_654	The Framework shall allow loading variables which are persistently stored.	CTA-D4.4-PS-10
S4R_FDF_655	The Framework shall store variables in way that they can be accessed by a partition using a unique identifier. E.g., identify a value by a key.	CTA-D4.4-PS-11
S4R_FDF_656	The Framework shall guarantee that no variable or file corruption occurs if the device switches off while writing data to a variable or a file.	CTA-D4.4-PS-12
S4R_FDF_657	The Framework shall allow closing files.	CTA-D4.4-PS-6
S4R_FDF_171	**Non-functional requirements**	n/a
S4R_FDF_172	**Performance requirements**	n/a
S4R_FDF_299	The Framework shall guarantee methodology for performance analysis for considered system configurations.	n/a
S4R_FDF_300	The Framework shall guarantee methodology for system performance analysis in case of accidental situations.	n/a

Id	**FDF requirement description**	CONNECTA requirements
S4R_FDF_301	The Framework shall define, configure, and assess performance of each node of system.	n/a
S4R_FDF_302	The Framework shall define, configure, and assess node performance for specified (cyber) security level.	n/a
S4R_FDF_303	The Framework shall define, configure, and assess node performance for I/O interface.	n/a
S4R_FDF_304	The Framework shall define, configure, and assess node performance for control algorithms and inter-partition communication.	n/a
S4R_FDF_305	The Framework shall define, configure, and assess node performance for logging and diagnostic subsystem.	n/a
S4R_FDF_306	The Framework shall define, configure, and assess node performance for communication interface.	n/a
S4R_FDF_307	The Framework shall define, configure, and assess performance of communication channels • channel priority • channel throughput	n/a
S4R_FDF_308	The Framework shall define, configure, and assess performance of communication channels for predefined parameters as: • jitter • latency • response time	n/a
S4R_FDF_309	The Framework shall define, configure, and assess performance for protection communication channels against cyber attack.	n/a
S4R_FDF_310	The Framework shall define, configure, and assess “performance for future use”: • data communication – capacity, throughput, security • control algorithms • fault tolerance	n/a
S4R_FDF_173	**Validation and verification support** The requirements in this subsection include all information regarding techniques used for testing purpose.	n/a
S4R_FDF_630	The Framework shall validate the installation or update of executable code before processing it. The scheduling and resources attached to other partitions shall not be affected.	CTA-D4.4-DM-17
S4R_FDF_631	The Framework shall validate the installation or update of a configuration file before processing it. The communication, scheduling and resources of partitions and processes shall not be affected.	CTA-D4.4-DM-29

Id	**FDF requirement description**	CONNECTA requirements
S4R_FDF_314	The Framework shall provide services to control and monitor the application execution (start, stop, synchronising to external trigger). I.e., using program flow monitoring techniques.	n/a
S4R_FDF_316	The Framework shall prevent the access to any validation and verification support service (fault injection and monitoring, forcing of outputs, monitoring of inputs and outputs, application control and monitoring, logging/tracing) on power up. The framework shall enable the validation and verification support services only on explicit request.	n/a
S4R_FDF_315	The Framework shall provide logging/tracing services for a selectable set of events related to • Fault injection and monitoring • Communication and shared network memory change • Output change • Input change • Application execution and monitoring	n/a
S4R_FDF_311	The Framework shall provide services to inject faults and monitor the fault reaction related to • non-critical (SIL0) • platform partitioning and isolation mechanism • communication (transmission, reception) and shared network memory • output control • input monitoring • application execution (timing, memory access, start, stop, throttling, …)	n/a
S4R_FDF_312	The Framework shall provide services to force the outputs to all states (valid and invalid) independent of the current control by the associated application.	n/a
S4R_FDF_313	The Framework shall provide services to monitor the state of all outputs and inputs independently from the application that is associated to the respective inputs/outputs.	n/a
S4R_FDF_174	**Interface requirements**	n/a
S4R_FDF_701	The Framework shall offer an interface to allow registering a variable that can be monitored externally.	CTA-D4.4-MO-1
S4R_FDF_702	The Framework shall offer an interface to allow external devices to request the list of variables which can be monitored.	CTA-D4.4-MO-2
S4R_FDF_703	The Framework shall offer an interface to allow external devices to request monitoring a number of variables with a given frequency.	CTA-D4.4-MO-5
S4R_FDF_706	The Framework shall provide an interface between input and output variables of processes executed in partitions - on the same device - on different devices in the same consist or - on devices in different consists of the same train according to their defined inputs and outputs.	CTA-D4.4-CM-7

Id	FDF requirement description	CONNECTA requirements
S4R_FDF_707	The Framework shall provide an interface between variables provided by I/O devices to inputs of processes executed in partitions - on the same device - on another device in the same consist or - in another consist of the same train according to the input definition of the partitions.	CTA-D4.4-CM-8
S4R_FDF_708	The Framework shall provide an interface between variables provided by a process executed on a partition to variables controlling outputs of I/O devices located - on the same device - on another device in the same consist or - in another consist of the same train according to the interface definition between the partition and the I/O device.	CTA-D4.4-CM-9
S4R_FDF_712	The Framework shall offer an interface to external devices to force variables.	CTA-D4.4-SM-1
S4R_FDF_713	The Framework shall offer an interface to register variable that can be forced.	CTA-D4.4-SM-2
S4R_FDF_734	The Framework shall guarantee the independence of I/O interfaces that can be requested by the application function.	n/a
S4R_FDF_175	**Application** The requirements in this section describe the interface requirements between applications and the framework.	n/a
S4R_FDF_318	The Framework shall offer an interface to create time-triggered processes.	n/a
S4R_FDF_320	The Framework shall offer an interface to set the priority of a process.	n/a
S4R_FDF_321	The Framework shall offer an interface to set the deadline of a process.	n/a
S4R_FDF_322	The Framework shall offer an interface to set the period of a time-triggered process.	n/a
S4R_FDF_323	The Framework shall offer an interface to set the offset of a time-triggered process.	n/a
S4R_FDF_324	The Framework shall offer an interface to set the activation events of an event-triggered process.	n/a
S4R_FDF_325	The Framework shall offer an interface to create periodic timers.	n/a
S4R_FDF_326	The Framework shall offer an interface to create sporadic timers.	n/a
S4R_FDF_327	The Framework shall offer an interface to set the deadline of a timer.	n/a
S4R_FDF_328	The Framework shall offer an interface to start a timer.	n/a
S4R_FDF_329	The Framework shall offer an interface to stop a timer.	n/a

Id	**FDF requirement description**	CONNECTA requirements
S4R_FDF_330	The Framework shall offer an interface to create partitions.	n/a
S4R_FDF_331	The Framework shall offer an interface to set the offset of a partition.	n/a
S4R_FDF_332	The Framework shall offer an interface to set the period of a partition.	n/a
S4R_FDF_333	The Framework shall offer an interface to set the budget of a partition.	n/a
S4R_FDF_334	The Framework shall offer an interface to set the processes of a partition.	n/a
S4R_FDF_335	The Framework shall offer an interface to create events.	n/a
S4R_FDF_336	The Framework shall offer an interface to launch an event.	n/a
S4R_FDF_337	The Framework shall offer an interface to discover, monitor and control the applications it executes.	n/a
S4R_FDF_501	The Framework shall offer an interface to read static configuration from a file.	n/a
S4R_FDF_176	**I/O** The requirements in this section describe the inputs and outputs of the Framework.	n/a
S4R_FDF_338	The Framework shall offer an interface to read the type and number of input and output ports.	n/a
S4R_FDF_339	The Framework shall offer an interface to read analog inputs.	n/a
S4R_FDF_340	The Framework shall offer an interface to read digital inputs.	n/a
S4R_FDF_341	The Framework shall offer an interface to write analog outputs.	n/a
S4R_FDF_342	The Framework shall offer an interface to write digital outputs.	n/a
S4R_FDF_343	The Framework shall offer an interface to map a variable to each analog or digital input or output.	n/a
S4R_FDF_344	The Framework shall offer an interface to determine the type, size and optional scaling/units of variables mapped to analog inputs and outputs.	n/a
S4R_FDF_345	The Framework shall offer an interface to determine the type, size and bit usage of variables mapped to digital inputs and outputs.	n/a
S4R_FDF_346	The Framework shall offer an interface to set the update cycle (multiple of basic cycle) for each mapped variable.	n/a
S4R_FDF_347	The Framework shall be able to map digital or analog input or output ports to data types complying with IEC 61375-2-1 [7] and IEC 61375-2-3 [2].	n/a
S4R_FDF_779	The Framework shall support at least 14 analog inputs with 12 bit resolution, 1 digital output and 7 digital outputs. If the	n/a

Id	FDF requirement description	CONNECTA requirements
	controller does not support such capabilities, alternative peripherals shall be provided (e.g., SPI).	
S4R_FDF_177	**Network** Network interfacing to COM/Middleware	n/a
S4R_FDF_348	For outgoing messages to the network, The network interface device shall read the message data from the partition message memory.	n/a
S4R_FDF_489	Application shall place message data into the partition message memory which is per configuration aligned with queuing or sampling ports.	n/a
S4R_FDF_349	For incoming messages from the network, the network interface device shall write the message data to the partition message memory.	n/a
S4R_FDF_490	Application shall read message data from the partition message memory which is per configuration aligned with queuing or sampling ports.	n/a
S4R_FDF_350	The configuration of the Framework and the Network shall specify for each port whether it is operated as a queuing or sampling port.	n/a
S4R_FDF_351	The configuration of the Framework (software abstraction / COM / middleware layer) shall define which data is stored into the message and at what point in time the message is published to the network.	n/a
S4R_FDF_352	The configuration of the Framework and the Network shall be consistent with regards to which frames are sent and received, at which times.	n/a
S4R_FDF_353	The Framework shall be able to receive status and errors related to message transmission in the network interface.	n/a
S4R_FDF_178	**Safety requirements**	n/a
S4R_FDF_180	**EC directive**	n/a
S4R_FDF_391	**EC Train Directive** DIRECTIVE (EU) 2016/797 [3] on the interoperability of the rail system within the European Union. Relevant chapters of Annex III of the directive: • 1.1.1 General requirements/Safety • 1.5 General requirements/Technical compatibility • 2.3.1 Control-command and signalling/Safety	n/a

Id	FDF requirement description	CONNECTA requirements
	• 2.4.1 Rolling stock/Safety • 2.4.2 Rolling stock/Reliability and availability • 2.4.3 Rolling stock/Technical compatibility	
S4R_FDF_392	**TSI LOC&PAS** 1302/2014/CE - COMMISSION REGULATION (EU) No 1302/2014 of 18 November 2014 [4]. Relevant chapters: • 4.2.4.2.1. (3), (4) Functional requirements • 4.2.4.2.1. (11) Functional requirements • 4.2.4.3 (1)/(2) Type of brake system • 4.2.4.10. (3) Brake requirements for rescue purposes • 4.2.5.2. (2), (3) Audible communication system • 4.2.5.3.1 (2) Passenger alarm/General	n/a
S4R_FDF_643	**Security requirements** This subsection defines the security-related requirements of FDF.	n/a
S4R_FDF_414	The framework shall secure the incoming/outgoing communication (channel) to the ECUs (Electronic Control Units) against security threats with regards to confidentiality, authenticity, integrity and availability whilst respecting real-time constraints (i.e. predictable latency and low jitter).	n/a
S4R_FDF_416	The framework shall protect stored data against adversaries (with regards to confidentiality, authenticity and data integrity).	n/a
S4R_FDF_417	The framework shall include a mechanism in order to prevent unknown/unexpected traffic (i.e. admission and access control).	n/a
S4R_FDF_420	The framework shall accomplish the need of protecting the data and state of the functions during execution on an ECU.	n/a
S4R_FDF_667	The Framework shall support cryptography algorithms, key sizes and mechanisms to key establishment and management according to common security industry practises and recommendations.	CTA-D4.4-SEC-13
S4R_FDF_412	The framework shall provide cryptographic mechanisms and handle cryptographic objects • Ensure framework's security as well as framework's communication channel (receiving and transmitting role) by means of secure cryptographic algorithms • Management of cryptographic keys (creation, deletion and retention)	CTA-D4.4-SEC-16

Id	FDF requirement description	CONNECTA requirements
	• Calculation of cryptographic functions (digital signatures, MACs, encryption/decryption)	
S4R_FDF_646	The Framework shall support data encryption.	CTA-D4.4-SEC-14
S4R_FDF_647	The Framework shall support data decryption.	CTA-D4.4-SEC-15
S4R_FDF_409	The framework shall operate accordingly/with regards to confidentiality • Ensure that data inside the framework cannot be read by an unauthorised entity: ensure non-disclosure of information/data towards entities (i.e. users, processes, and device) unless a successful access authorisation.	n/a
S4R_FDF_410	The framework shall operate accordingly/with regards to authenticity • Assurance of entities' identity • Ensure/verify data source: information/data comes from a verified and trusted entity (sender) • Information collected by the framework should be authentic with respect to origin and time if the framework performs actions based on that information • The author of the message, respectively the origin sending entity of the information/data, shall be evident and traceable at any time (with regards to non-repudiation)	CTA-D4.4-SEC-1
S4R_FDF_415	The Framework shall support availability of access control in the network to ensure robustness to DoS attacks as well as side-channel attacks.	n/a
S4R_FDF_429	The framework shall ensure that security policy enforcement functions and the data that configures them cannot be modified without authorisation.	n/a
S4R_FDF_418	The framework shall support secure storage for key(s) and trust anchor(s) for secure authentication and communication (with regards to security services and authenticity).	n/a
S4R_FDF_419	The framework shall operate with authenticated entities (ECUs, SW/HW components) only (with regards to authenticity) • The framework shall enforce authenticity and integrity of the ECUs in order to meet/fulfil framework's security requirements. • The framework shall enforce authenticity and integrity of the software components in order to meet/fulfil framework's security requirements.	CTA-D4.4-SEC-1
S4R_FDF_669	The Framework shall allow to assign privileges to authenticated users (access rights).	CTA-D4.4-SEC-2
S4R_FDF_670	The Framework shall support executable identification and authentication.	CTA-D4.4-SEC-3
S4R_FDF_671	The Framework shall allow to assign privileges to authenticated executables (access rights).	CTA-D4.4-SEC-4
S4R_FDF_672	The Framework shall: • initialise authenticator content	CTA-D4.4-SEC-5

Id	FDF requirement description	CONNECTA requirements
	• change all default authenticators upon control system installation • change/refresh all authenticators • protect all authenticators from unauthorised disclosure and modification when stored and transmitted.	
S4R_FDF_673	The Framework shall support the management of identifiers by users, groups, roles or control system interfaces.	CTA-D4.4-SEC-6
S4R_FDF_749	The component "Security Management" shall be able to support the management of all accounts by authorized users, including adding, activating, modifying, disabling and removing accounts.	n/a
S4R_FDF_674	The Framework shall enforce configurable password strength based on minimum length and variety of character types.	CTA-D4.4-SEC-7
S4R_FDF_413	The framework shall provide a Public Key Infrastructure (PKI) • Support/ensure the authentication process of entities (with regards to authenticity) • Management of certificates (retention and update)	CTA-D4.4-SEC-8
S4R_FDF_676	The Framework shall validate certificates by: • checking the signature of given certificates • constructing a certification path to an accepted CA • deploying leaf certificates to all hosts which communicate with the subject to which the certificate is issued (in the case of self signed certificates) • checking the certificate's revocation.	CTA-D4.4-SEC-9
S4R_FDF_677	The Framework shall: • establish user (human, SW process, device) control of the private keys • map the authenticated identity to a user (human, SW process, device).	CTA-D4.4-SEC-9
S4R_FDF_678	The Framework shall be able to obscure feedback authentication information during authentication process.	CTA-D4.4-SEC-10
S4R_FDF_679	The Framework shall enforce a limit of configurable number of consecutive invalid access attempts by any user (human, SW, device) during a configurable time period.	CTA-D4.4-SEC-11
S4R_FDF_680	The Framework shall deny access for specified period of time or until unlocked by an administrator when the access attempts number is exceeded.	CTA-D4.4-SEC-11
S4R_FDF_681	The Framework shall display a system notification message before authenticating. This message shall only be configurable by authorised users.	CTA-D4.4-SEC-12
S4R_FDF_430	The Framework shall provide the capability to detect, generate and export audit records for security relevant auditable events.	n/a
S4R_FDF_730	The Framework shall periodically verify the correct operation of security protection functions and notify system	n/a

Id	FDF requirement description	CONNECTA requirements
	administrator when anomalies are discovered.	
S4R_FDF_411	The Framework shall operate accordingly/with regards to data integrity • Support/offer mechanism(s) in order to ensure data integrity for information collected within the framework. • Ensure that the information has/have not been modified either in transit or in storage on the route from the sender's entity to the receiver's entity.	CTA-D4.4-SEC-17
S4R_FDF_421	The framework shall accomplish the need of protecting the data and state of the functions during execution within software components.	n/a
S4R_FDF_422	The framework shall ensure the data isolation between different partitions created and maintained by the framework so that the data in a partition is accessible only by code running in that partition (SIL).	n/a
S4R_FDF_423	The framework shall ensure the isolation of the resource between different partitions created and maintained by the framework so that the resources exported by the framework into a partition are accessible only by code running in that partition (with SIL).	n/a
S4R_FDF_424	The framework shall provide information flow control that enforces strict partition isolation so that only explicitly configured interaction are allowed.	n/a
S4R_FDF_425	The framework shall ensure that a failure in one partition is not propagated to other partitions.	n/a
S4R_FDF_426	The framework shall ensure that an attack affecting one partition is not propagated to other partitions.	n/a
S4R_FDF_427	The framework shall ensure that security policy enforcement functions cannot be bypassed.	n/a
S4R_FDF_428	The framework shall ensure that security policy enforcement functions are always invoked.	n/a
S4R_FDF_731	The Framework or its support utilities shall provide user functionality to facilitate creation of backups of user-level and system-level information (including system security state information).	n/a
S4R_FDF_732	The Framework shall provide user functionality to allow be recovering and reconstituting to previously saved Backup after a disruption or failure.	n/a
S4R_FDF_182	**RAMS requirements**	n/a
S4R_FDF_478	The Framework shall provide a safe communication path for transmission/reception of datasets using a safety layer.	n/a
S4R_FDF_479	The Framework shall offer application interfaces according to the safety layer needed: •non-critical (SIL0) •SIL2 •SIL4	

Id	FDF requirement description	CONNECTA requirements
	where the ability to provide SIL2 and SIL4 APIs depends on the specific implementation of the framework (on HW/SW).	
S4R_FDF_480	The Framework shall guarantee the integrity and validity of the received data to meet the requirements for SIL2 (according to IEC61508-1 [5]). SDTv2, as defined in IEC61375-2-3 Annexe B [2], provides this safety level for PFH ≥ 10E-7 < 10E-6 (1% for black channel communication).	n/a
S4R_FDF_481	The Framework shall guarantee the integrity and validity of the received data to meet the requirements for SIL4 (according to IEC61508-1). A PFH ≥ 10E-9 < 10E-8 (1% for black channel communication) is needed.	n/a
S4R_FDF_482	The Framework shall inform the application of communication losses, which enable the application to decide whether to set the system into the 'safe state'.	n/a
S4R_FDF_483	The Framework shall monitor the operational state of the ECU (and its function(s)) by appropriate means and report in case of failure. I.e., implementing error detection and correction (EDC) technique.	n/a
S4R_FDF_484	The Framework shall share its operational state with all other ECUs in its functional group(s).	n/a
S4R_FDF_485	The Framework shall detect and verify the operational status of other redundant ECUs.	n/a
S4R_FDF_486	The Framework shall inform the application of the operational status of all other ECUs in its functional group(s).	n/a
S4R_FDF_487	The Framework shall be operational within 60 seconds from power-up.	n/a
S4R_FDF_488	The Framework shall perform a self-test of the ECU on power-up.	n/a
S4R_FDF_467	**Configuration management**	n/a
S4R_FDF_431	The Framework shall be configurable on ECU reset or start-up by a local configuration.	n/a
S4R_FDF_432	The Framework shall be able to receive an additional remote configuration via network.	n/a
S4R_FDF_433	The Framework shall check the validity and integrity of any configuration. This could be a CRC, MD or signature created by tooling.	n/a
S4R_FDF_434	The Framework shall check the origin of remote configurations and ignore false configurations. Remote configurations must be certified.	n/a
S4R_FDF_435	The remote configuration's properties shall take precedence over the same properties of the local configuration. This relates to dynamic vs. static configuration, e.g. direction dependent addressing and default parameters.	n/a
S4R_FDF_436	The Framework shall provide a local interface to retrieve static and dynamic configuration properties by a host	n/a

Id	FDF requirement description	CONNECTA requirements
	application.	
S4R_FDF_437	The Framework shall provide a remote (network) interface to retrieve static and dynamic configuration properties of an ECU.	n/a
S4R_FDF_438	The Framework's local configuration shall define the necessary properties for local communication needs. Note: Annex C of IEC 61375-2-3 [2] defines an XML format which covers most properties of a communication framework. Train-wide communication depends on train inauguration and may therefore not be possible with local configurations, only. This depends on the future network layout (defined in WP1).	n/a

Table 9: Safe4RAIL FDF requirements vs. CONNECTA FDF requirements defined in CONNECTA T4.4 - Traceability matrix.

Annex G – Integrated Modular Platform: Traceability Matrix

The Integrated Modular Platform (IMP) is composed of the Functional Distribution Framework and Drive-by-Data technologies and, thus, its requirements, defined in WP1, are propagated to one, another or both technologies. Thetable below contains the IMP traceability towards the FDF.

Id	**FDF requirement description**	IMP requirements
S4R_FDF_165	**Functional requirements**	n/a
S4R_FDF_166	**Application execution**	n/a
S4R_FDF_584	**Execution management** This subsection defines the execution management which is in charge of handling the execution of application functions and executable instances.	n/a
S4R_FDF_585	The Framework shall support authentication and authorisation of executables at start-up.	n/a
S4R_FDF_586	The Framework shall check the integrity of executables at start-up.	n/a
S4R_FDF_737	The Framework shall inhibit the execution of the application function in the case of negative code integrity check.	n/a
S4R_FDF_766	The Framework shall avoid forcing outputs when application function is operative (nominal and degraded).	n/a
S4R_FDF_767	The Framework shall prevent the access of off-line services at power-up, during initialization and operation (nominal and degraded).	n/a
S4R_FDF_768	The Framework shall guarantee the retention of a safe-state after a fatal fault.	n/a
S4R_FDF_782	The Framework shall be able to generate partitions and allocate resources for application functions requiring multiple-instances for the implementation of reliable and safe architecture.	n/a
S4R_FDF_588	The Framework shall support multiple executable instances.	n/a
S4R_FDF_589	The Framework shall consider unambiguous identification of executable instances (i.e., processes) provided by the configuration.	n/a

Id	FDF requirement description	IMP requirements
S4R_FDF_594	The Framework shall support ordered execution of processes, partitions and FDF components.	n/a
S4R_FDF_684	The Framework shall guarantee a pre-emptive and priority based schedule for concurrent execution.	n/a
S4R_FDF_686	The Framework shall manage redundant execution of partitions and/or processes on different devices.	n/a
S4R_FDF_692	The Framework shall provide a mechanism for service discovery and announcement.	n/a
S4R_FDF_687	The Framework shall support configurable recovery actions in case of partition or process deviations from normal behaviour.	n/a
S4R_FDF_688	The Framework shall provide internal variables as outputs and the 'leader" shall update those outputs after each redundant execution of partitions or processes. The internal variables are persistent over more than a single execution of the partition or process.	n/a
S4R_FDF_693	The Framework shall provide internal variables as the input to synchronise the internal variables of a "follower" with the variables provided by the "leader" before each execution of partitions or processes. The internal variables are persistent over more than a single execution of the partition or process.	n/a
S4R_FDF_694	The Framework shall interface with the Monitoring Manager in a secure way, by offering authentication measures for instance, to provide the availability of forcing variables.	n/a
S4R_FDF_695	The Framework shall be able to suspend the execution of processes and/or partitions during a given time.	n/a
S4R_FDF_741	The Framework shall load the configuration file during inauguration.	n/a
S4R_FDF_755	The Framework shall guarantee calls to service functions with the same SIL assigned to the application functions using services.	n/a
S4R_FDF_783	The Framework shall guarantee spatial separation between memory spaces containing read-only and read-write variables, variables with different SIL, variables used by multiple independent instances, software code and parameters of the application function.	n/a
S4R_FDF_185	**Process management** This subsection defines a process and describes how the IMP (Integrated Modular Platform) interacts with each of these. The ECP (Extended Capabilities Port) shall offer services to create and manage timers for sequential execution and semaphores for sequential and concurrent execution.	n/a
S4R_FDF_506	A process shall be in the state: • Suspended: The process is not permitted to be activated. • Waiting: The process is waiting for its activation, which depending on the triggering paradigm will be when the corresponding event is launched or it is a certain instant of time. • Ready: The process is ready to execute and will do it if it has the highest priority among the ready processes. • Running: The process is executing in the processor.	n/a

Id	FDF requirement description	IMP requirements
S4R_FDF_193	The Framework shall activate a time-triggered process in waiting state if: • The current time is inside its partition time slot • The current time is a multiple of its period	n/a
S4R_FDF_498	The Framework shall grant spatial separation among processes.	S4R-IMP-003
S4R_FDF_194	The Framework shall execute the process in ready state with the highest priority.	n/a
S4R_FDF_195	The Framework shall set the state of a process to waiting when its execution finishes.	n/a
S4R_FDF_196	The Framework shall launch the finishing event of a process when its execution finishes.	n/a
S4R_FDF_197	The Framework shall set a process in ready state to waiting if it waits for an event.	n/a
S4R_FDF_610	The processes shall be configured according to a configuration file.	n/a
S4R_FDF_497	The Framework shall execute processes sequentially or concurrently.	n/a
S4R_FDF_698	The Framework shall limit the execution time for each process.	S4R-IMP-009
S4R_FDF_519	The Framework shall control the execution of processes with the same SIL assigned to the involved application functions.	S4R-IMP-004
S4R_FDF_587	The Framework shall set-up separate process to execute each instance.	n/a
S4R_FDF_520	The Framework shall implement service functions whose response times allow the real-time execution of processes and implement mechanisms to ensure that execution.	n/a
S4R_FDF_521	The Framework shall monitor execution of processes concerning defined time-bounds for communication and processing.	S4R-IMP-012
S4R_FDF_604	The Framework shall support configurable recovery actions in case of a process deviates from normal behaviour.	n/a
S4R_FDF_522	The Framework shall notify a fault condition in case of error in the process execution.	n/a
S4R_FDF_523	A process can belong to different process schedules.	n/a
S4R_FDF_700	The Framework shall allow to processes to set and get the current FDF's operation mode.	n/a
S4R_FDF_199	**Partition management** Partitions give means to guarantee memory space separation, which might contain all the information of processes. Besides, a cyclic executive scheduler must give and take away access to the processor when corresponds.	n/a
S4R_FDF_205	A partition shall be active or inactive.	n/a

Id	**FDF requirement description**	IMP requirements
S4R_FDF_206	Only active partitions shall be executed.	n/a
S4R_FDF_208	The Framework shall guarantee temporal separation among partitions by ensuring that a process within a given time budget cannot be affected by the actions of any other tasks of any other partition.	S4R-IMP-009
S4R_FDF_592	The Framework shall bind the period, and execution time for each partition.	n/a
S4R_FDF_685	The Framework shall ensure the independence (time, space) of services and to support partitions' independence.	n/a
S4R_FDF_759	The Framework shall manage interrupts through the O.S, to avoid any disturbance to time partitioning.	n/a
S4R_FDF_606	The Framework shall support synchronised execution of partitions among different processor cores and devices.	S4R-IMP-003
S4R_FDF_607	The Framework shall write/update the inputs of each partition before executing them.	n/a
S4R_FDF_609	The Framework shall write/update the outputs of each partition after executing them.	n/a
S4R_FDF_689	The Framework shall execute and write/update the outputs, when the partition has the redundancy role "leader".	n/a
S4R_FDF_690	The Framework shall execute the partition, but shall not write/update its outputs, when the partition has the redundancy role "follower".	n/a
S4R_FDF_691	The Framework shall activate one of the "follower" partitions in the case that the "leader" partition fails. The "follower" shall write the outputs of "leader" partition.	n/a
S4R_FDF_524	Partitions shall guarantee spatial separation to ensure that no process in one partition can modify without authorisation software code or application data of another partition. E.g., by means of memory protection mechanisms.	n/a
S4R_FDF_525	Partitions are configured according to the configuration file of the application functions to be executed.	n/a
S4R_FDF_527	A partition can belong to different partition schedules.	n/a
S4R_FDF_526	Partitions have assigned computational resources defined in configuration file. No resource is shared by partitions hosting application functions with different SIL.	S4R-IMP-006, S4R-IMP-011
S4R_FDF_210	Partitions shall contain one or more processes.	n/a
S4R_FDF_507	A partition shall be in the state: • Suspended: The partition is not permitted to be activated. • Waiting: The partition is waiting for its activation, which depending on the triggering paradigm will be when the corresponding event is launched or it is a certain instant of time. • Ready: The partition is ready to execute and will do it if it has the highest priority among the ready partitions. • Running: The partition is executing in the processor. • Isolated: The partition is isolated and it is not permitted to be activated.	n/a

Id	FDF requirement description	IMP requirements
S4R_FDF_602	The Framework shall not execute partitions in state suspended or isolated.	n/a
S4R_FDF_603	The Framework shall support configurable recovery actions in case of a partition deviates from normal behaviour.	n/a
S4R_FDF_528	Partitions shall notify fault conditions in case of invalid operation in the partition attempt (fatal fault).	n/a
S4R_FDF_784	The Framework shall assign privileges for read-write access to a memory space only to independent application functions with at least the same SIL. Read-only access could be assigned to remaining application functions, if data alteration during reading can be excluded.	n/a
S4R_FDF_215	**Concurrency management** This subsection gives details regarding concurrency control and synchronisation techniques.	n/a
S4R_FDF_216	An event shall be active or inactive.	n/a
S4R_FDF_217	The Framework shall activate an event when it is commanded to launch.	n/a
S4R_FDF_590	The Framework shall support concurrent execution of more than one partition in different processor cores and/or devices.	n/a
S4R_FDF_529	Concurrent accesses to shared resources shall be synchronised using semaphores and/or mutexes.	n/a
S4R_FDF_530	Concurrent executions shall be synchronised using semaphores and/or mutexes.	n/a
S4R_FDF_612	**Configuration management** This subsection defines the requirements regarding the configuration the Functional Distribution Framework including settings for partitions and variables.	n/a
S4R_FDF_613	The Framework shall statically identify an ECU instance at boot time (e.g., by local digital inputs)	n/a
S4R_FDF_614	The Framework shall dynamically acquire the identification of ECUs instances at boot time (e.g., by DHCP).	n/a
S4R_FDF_625	The Framework shall obtain the identifier of an ECU instance.	n/a
S4R_FDF_742	The configuration and re-configuration of the Framework shall involve all the application functions.	S4R-IMP-024
S4R_FDF_615	The Framework shall acquire the following configuration parameters of the FDF and make them available to the FDF's components with the same SIL assigned to related application functions. • Version information • User identification and privileges • Contained devices	S4R-IMP-026, S4R-IMP-028

Id	FDF requirement description	IMP requirements
	• Contained partitions • Scheduling parameter of contained partitions • Contained communication networks	
S4R_FDF_616	The Framework shall acquire the following configuration parameters of a device and make them available according to the SIL assigned to related application functions. • Contained I/O units.	n/a
S4R_FDF_618	The Framework shall acquire the consist network configuration of a given SIL and make it available to the FDF components with the same SIL.	n/a
S4R_FDF_619	The Framework shall acquire the configuration parameters of partitions and make them available to the FDF components. • Unique identifier • Version information • Execution period • Maximum execution time • Redundancy role • In- and Output variables • Contained processes • Scheduling policy and dependencies of the contained processes • Error handling including recovery actions	n/a
S4R_FDF_620	The Framework shall acquire the following configuration parameter set for a process FDF and make them available to the FDF components. • Unique identifier • Executable that is executed in the process • Mapping of input/output variables to variables provided by or send to other processes, network or I/O • Assigning rights for publishing/reading variables to the SW components. • Execution period • Maximum execution time • Redundancy role • Scheduling priority • Error handling including recovery actions	n/a

Id	FDF requirement description	IMP requirements
	• Access to FDF services (e.g. set global time)	
S4R_FDF_621	The Framework shall acquire the following configuration parameter set for an executable and make them available to the FDF components. • Unique identifier • Version information • In- and Output variables • Variables available for external monitoring • Variables stored persistently • Provided and required services	n/a
S4R_FDF_622	The Framework shall acquire the following configuration parameter set for an IO unit and make them available to the FDF components. • Unique identifier • In- and Output variables • Decoder configuration for encoder signals • Update cycle of in- and output variables	n/a
S4R_FDF_623	The Framework shall acquire the following configuration parameter set of a variable and make them available to the FDF components. • Unique identifier • Value interpretation • Default value • Data type	n/a
S4R_FDF_624	The Framework shall acquire the configuration parameter set of a service and make them available to the FDF components. • Unique identifier	n/a
S4R_FDF_626	The Framework shall acquire the following configuration parameter set for the event log and make them available to the FDF components. • maximum size • time period for storage of reoccurring events	n/a
S4R_FDF_167	**Communication management**	n/a

Id	FDF requirement description	IMP requirements
	This subsection contains requirements related to communication management.	
S4R_FDF_220	**Data and event distribution** This chapter contains requirements on event and ECU and application data distribution which is done by the use of distribution variables between processes.	n/a
S4R_FDF_221	The Framework shall provide services to create exchange variables, which are data structured consisting of a set of parameter and value pairs and should be SIL independent.	S4R-IMP-005
S4R_FDF_222	The variables shall be exchanged between software components using the publish-subscribe pattern. f) Communication is black channel (including the publish-subscribe pattern) g) Safety relevant process data must be encrypted h) Non-Safe process data may be encrypted i) Encryption credentials must be configured j) Public/private key encryption is not sufficient - there must be certificates exchanged to prevent 3rd party access to safety critical functions handled in 2.5 Security requirements Note: The publish–subscribe is a messaging pattern where senders of messages (publishers) do not directly send messages to specific receivers (subscribers) but instead characterise published messages into classes (e.g. certain variables) without knowledge of which subscribers, if any, there may be. Similarly, subscribers express interest in one or more classes and only receive messages that are of interest, without knowledge of which publishers, if any, there are.	n/a
S4R_FDF_223	The Framework shall give software components read and write access only to the variables they are allowed to publish.	n/a
S4R_FDF_224	The Framework shall give software components read access only to the variables they are subscribed to (without altering their value).	n/a
S4R_FDF_753	The Framework shall give software components write access according to specification set during configuration.	n/a
S4R_FDF_225	The Framework shall guarantee that the software component publishing a variable is able to update its value.	n/a
S4R_FDF_226	The Framework shall guarantee that an updated value is accessible for every software component that is subscribed to it within the defined timely bound.	n/a
S4R_FDF_735	The Framework shall guarantee the updating of input variables according to the values of input channels and SIL level.	n/a
S4R_FDF_227	The Framework shall guarantee that the communicating software components may exchange messages in the same way, regardless of the location of the software components, be it: • in the same process	S4R-IMP-001, S4R-IMP-002

Id	FDF requirement description	IMP requirements
	• in different processes of the same partition • in different partitions of the same ECU • or in different ECUs of the same network Especially in the case of different ECUs on the same network, security aspects shall be considered. (handled in 2.5 Security requirements)	
S4R_FDF_493	The Framework shall provide services to exchange Message data (non-cyclic/best-effort) using a "notification", "call/reply" or "call/reply/confirm" pattern.	n/a
S4R_FDF_494	The Framework shall provide services to request data of variables non-cyclic/non-deterministic.	n/a
S4R_FDF_495	The Framework shall provide services to read out the TTDB (Train Topology Database) which is the result of inauguration.	S4R-IMP-015
S4R_FDF_541	The framework shall provide the ability to set default values to variables: • of the train and consist network and • shared between partitions of the same device • Shared between processes of the same partition according to configuration.	n/a
S4R_FDF_496	The Framework shall provide services to supervise the validity of the inauguration result.	n/a
S4R_FDF_508	The Framework shall provide services to support different redundancy concepts.	n/a
S4R_FDF_709	The Framework shall be able to replicate the value of local input variables on the consist network according to configuration.	n/a
S4R_FDF_509	The Framework shall provide services to define a variable which can be then updated from different redundant devices.	n/a
S4R_FDF_510	The Framework shall provide services to define a set of redundant variables which are each updated by the corresponding redundant device.	n/a
S4R_FDF_511	The Framework shall mark the variables as valid or invalid according to the chosen redundancy concept. (E.g. one out of two, two out of three...)	n/a
S4R_FDF_543	The Framework shall provide the ability to create and manage access to shared memories to facilitate communication between processes of the same partition.	n/a
S4R_FDF_780	The Framework shall guarantee the validity of safety-related data exchange between remote functions through messages composing and decomposing into variables out with the same SIL assigned to the application functions.	n/a
S4R_FDF_781	The Framework shall allow message function to access to memory spaces containing messages and variables with the same SIL.	n/a
S4R_FDF_785	The Framework shall guarantee the read-write access to memory spaces (according to the assigned privileges) with the same SIL	n/a

Id	FDF requirement description	IMP requirements
	assigned to the Application function(s) and variables stored.	
S4R_FDF_786	The Framework shall execute an Application function, giving access to memory resources, only when required by its scheduling plan (and take away access otherwise).	n/a
S4R_FDF_228	**Networking** Networking comprises requirements on location transparency, whether a Publish-Subscribe pattern is used and the number of participants or the support of deterministic real-time and best-effort messages.	n/a
S4R_FDF_229	The Framework shall provide communication mechanisms that are abstracted of the physical realisation of the communication hardware.	S4R-IMP-002
S4R_FDF_230	The Framework shall provide a standardised software interface (Ethernet) for communication between software components ensuring their communication independent whether they are located • on the same ECU on the same core • on the same ECU on another core • on the same ECU on another microcontroller on another ECU	S4R-IMP-001
S4R_FDF_711	The Framework shall provide an IEC 61375-2-3 compliant safety layer for the consist network communication.	n/a
S4R_FDF_231	The Framework shall provide a communication service that allows it to send messages (containing variables) to other components on the network within defined timely bounds from the point in time where the application sends the message to the point in time it is sent on the network (deterministic sending).	n/a
S4R_FDF_756	The Framework shall instantiate messages according to the configuration file, including: • Unique identifier (ID) • Messages to be received or send • List of variables linked to messages • Messages schedule • Deadline	n/a
S4R_FDF_750	The Framework shall periodically send messages within defined time bounds and receive them within defined maximum delay (deterministic sending).	n/a
S4R_FDF_512	The Framework shall provide a communication service which provides a deterministic way for an application to announce/prepare a message/data value for deterministic sending.	n/a
S4R_FDF_232	The Framework shall provide a communication service that makes received messages from other components on the network available to the application within defined timely bounds (deterministic receiving).	n/a

Id	FDF requirement description	IMP requirements
S4R_FDF_513	The Framework shall provide a communication service which provides a deterministic way to fetch a message/data value after deterministic reception.	n/a
S4R_FDF_751	The Framework shall implement Communication service without any operation on the messages' safety layer content.	n/a
S4R_FDF_233	**System integration** This chapter contains requirements regarding the COM layer, the inauguration process or transport layer protocols among others. System Integration Requirements are covered in detail in D1.11.	n/a
S4R_FDF_168	**Time management** Different ECUs share a unique global time that is synchronised with UTC. These requirements contain details regarding interfaces used, protocols and ways of synchronisation, i.e., automatic or manual.	n/a
S4R_FDF_235	The Framework shall provide a service for starting application processes based on the progression of time.	n/a
S4R_FDF_236	The Framework shall synchronise the local computer clock with the external global clock source and keep it synchronised with a maximum deviation of the global clock source of 1 microsecond.	n/a
S4R_FDF_762	The Framework shall synchronize the local clock independently from the execution of different partition's processes.	n/a
S4R_FDF_237	The Framework shall allow process and partition execution to be scheduled at a configured time instant within a configured rate-monotonic execution cycle period.	n/a
S4R_FDF_238	The Framework shall check and inform about successful synchronisation, synchronisation state and synchronisation errors.	n/a
S4R_FDF_544	The Framework shall provide allow processes to set the global time if allowed by configuration to do so.	n/a
S4R_FDF_545	The Framework shall provide the ability to processes to create, configure and delete timers.	n/a
S4R_FDF_239	The global time shall be made available to all ECUs through the network layer.	n/a
S4R_FDF_240	Global time dissemination shall be fault tolerant. Note: In case no time synchronisation is available, there is no scheduled (critical) communication possible. In case of erroneous time synchronisation, messages may arrive early or late and can lead to catastrophic events. This erroneous time synchronization must be detected by the SDT layer.	n/a
S4R_FDF_736	The Framework shall not finalize the inauguration without a valid global-time.	n/a

Id	FDF requirement description	IMP requirements
S4R_FDF_169	**Input/output management**	n/a
S4R_FDF_261	**Input management** This subsection contains requirements specifying which Input devices the ECU must be able to work with and how the data of these devices should be read and interpreted.	n/a
S4R_FDF_263	The Framework shall provide a service to create the controller access to an analog input.	n/a
S4R_FDF_264	The Framework shall provide a service to create the controller access to a digital input.	n/a
S4R_FDF_265	The inputs shall be accessible over configurable symbolic names.	n/a
S4R_FDF_764	The Framework shall allow input functions to access only to memory spaces with the same SIL.	n/a
S4R_FDF_266	The Framework shall create an exchange variable associated with each input channel.	n/a
S4R_FDF_546	The Framework shall set default values to digital and analog input variables according to configuration, with the same SIL assigned to related application functions.	n/a
S4R_FDF_267	The exchange variable associated with an input channel shall contain the acquired input channel value.	n/a
S4R_FDF_268	The Framework shall store the current value of every used input at the end of each acquisition cycle in the associated exchange variable.	n/a
S4R_FDF_269	The Framework shall provide a service for reading the last valid value of every used input, stored in the associated exchange variable.	n/a
S4R_FDF_270	The service for reading the value of every used input stored in the associated exchange variable shall not be interruptible to ensure data consistency.	n/a
S4R_FDF_716	The Framework shall decode encoder signals and transfer the value into a variable, including validity information.	n/a
S4R_FDF_262	**Output management** Analogously, this other subsection contains requirements specifying which Output devices the ECU must be able to work with and how the data of these devices should be written and interpreted.	n/a
S4R_FDF_271	The Framework shall provide a service to create the controller access to an analog output.	n/a
S4R_FDF_272	The Framework shall provide a service to create the controller access to a digital output.	n/a
S4R_FDF_273	The outputs shall be accessible over configurable symbolic names.	n/a
S4R_FDF_765	The Framework shall allow output functions to access only to memory spaces with the same SIL.	n/a

Id	FDF requirement description	IMP requirements
S4R_FDF_274	The Framework shall create an exchange variable associated with each output channel.	n/a
S4R_FDF_547	The Framework shall set digital and analog outputs to default values according to configuration, with the same SIL assigned to related application functions.	n/a
S4R_FDF_275	The exchange variable associated with an output channel shall contain the output channel set value.	n/a
S4R_FDF_276	The Framework shall provide a service for writing a new value and update it in the associated exchange variable of every used output.	n/a
S4R_FDF_277	The service for writing a new value in the associated exchange variable of every used output shall not be interruptible to assure data consistency.	n/a
S4R_FDF_548	**Health management**	S4R-IMP-012
S4R_FDF_549	The Framework shall support CPU, board and/or rack temperature monitoring, if supported by the HW monitoring.	n/a
S4R_FDF_551	The Framework shall support checking if partitions are executed within their maximum execution time.	n/a
S4R_FDF_552	The Framework shall support a HW watchdog timer (WDT).	n/a
S4R_FDF_553	The Framework shall refresh the WDT.	n/a
S4R_FDF_554	The Framework shall support integrity checks of the HW.	n/a
S4R_FDF_555	The Framework shall support check if partitions and processes update their outputs according to the value of variables and SIL level.	n/a
S4R_FDF_557	The Framework shall log the errors detected in a log file.	n/a
S4R_FDF_728	The Framework shall check the timeliness and sequence of messages exchanged between remote functions.	S4R-IMP-018
S4R_FDF_556	The Framework shall provide reaction to errors when a partition or process: • does not write the output • does not terminate execution in time • CPU, board and/or rack temperature exceeds the allowed range • CPU, board and/or rack load too high	n/a
S4R_FDF_558	The Framework shall consider the following reaction to error mechanisms with the highest SIL assigned to the application functions, without disturbing to other framework's services: • restart the ECU of the affected partition/process (without affecting other ECUs) • restart the affected partition/process (without affecting other partitions/processes)	n/a

Id	FDF requirement description	IMP requirements
	- isolate/terminate the affected partition/process (without affecting other partitions/processes) - inform the application function and continue with normal operation	
S4R_FDF_746	The Framework shall provide reaction to errors when a communication error is identified: - message authenticity - message integrity - message timeliness - message sequence	n/a
S4R_FDF_745	The Framework notifies to application function and reacts against safety-related communication errors, for example, discarding erroneous messages.	n/a
S4R_FDF_754	The Framework shall detect and notify the application SW in case of unavailability of scheduled services or in case of incorrect calls (different schedules).	n/a
S4R_FDF_758	The Framework shall notify fault conditions to all the Application function(s) involved (with SIL) without disturbing to other framework's services and no later than the maximum time for safe state.	n/a
S4R_FDF_769	The Framework shall notify a fault condition to the related application function in case of inconsistencies between the values stored into an exchange variable and the status of the platform's input/output.	n/a
S4R_FDF_179	**Monitoring management**	n/a
S4R_FDF_562	The Framework shall allow remotely requesting the list of available variables.	n/a
S4R_FDF_563	The Framework shall allow remotely registering variables that can be monitored.	n/a
S4R_FDF_564	The Framework shall send the list of variable that can be monitored to external device.	n/a
S4R_FDF_355	The Framework shall allow remotely reading the variables of a component.	n/a
S4R_FDF_356	The Framework shall allow remotely writing the variables of a component.	n/a
S4R_FDF_357	The Framework shall allow remotely reading the events of a component.	n/a
S4R_FDF_358	The Framework shall allow remotely writing the events of a component.	n/a
S4R_FDF_359	The Framework shall allow remotely forcing the variables of a component.	n/a
S4R_FDF_361	The Framework shall allow remotely unforcing the variables of a component.	n/a

Id	FDF requirement description	IMP requirements
S4R_FDF_362	The Framework shall allow remotely forcing the events of a component.	n/a
S4R_FDF_363	The Framework shall allow remotely unforcing the events of a component.	n/a
S4R_FDF_364	The Framework shall check the state of all existing processes.	n/a
S4R_FDF_365	The Framework shall check the value of all framework variables, comparing them with the I/O values.	n/a
S4R_FDF_704	The Framework shall guarantee a secure communication with external devices.	n/a
S4R_FDF_733	The Framework shall provide services to monitor variables (e.g., remotely (out of FDF)).	n/a
S4R_FDF_761	The Framework shall detect faults with the highest SIL assigned to the application functions to be executed, without disturbing to other framework's services.	n/a
S4R_FDF_738	The Framework shall detect resource-related faults at power-up and periodically.	n/a
S4R_FDF_743	The Framework shall detect incoherence of configuration file.	n/a
S4R_FDF_744	The Framework shall detect the lack of configuration file's integrity.	S4R-IMP-025
S4R_FDF_752	The Framework shall assign to the monitoring-function RO privileges to variables stored into memory spaces with lowest integrity level or to all the memory spaces with different integrity levels (SIL) without altering the execution of other services.	n/a
S4R_FDF_760	The Framework shall monitor the alignment with the external global clock with the highest SIL assigned to the application functions to be executed.	n/a
S4R_FDF_771	The Framework shall monitor that non-safety data uses different structures than ones used for safety-related data.	n/a
S4R_FDF_788	The Framework shall provide fault detection during run-time execution.	n/a
S4R_FDF_789	The Framework shall provide further measures and detection techniques, in addition to the techniques/measures provided, for run-time fault detection.	n/a
S4R_FDF_377	**Log management** This subsection describes which information the system log should include. This could be sensitive activity, errors or the state of the different processes.	n/a
S4R_FDF_378	The Framework shall create a log file per day (if applicable persistent log file).	n/a
S4R_FDF_574	The Framework shall configure the maximum size of the event log.	n/a
S4R_FDF_575	The Framework shall overwrite previously recorded event if the maximum of the log file size is reached.	n/a

Id	FDF requirement description	IMP requirements
S4R_FDF_576	The Framework shall only record one error every certain period of time, in case of recurrent errors. The logging period of time shall be configurable.	n/a
S4R_FDF_380	The Framework shall log the minimum execution time of the processes per hour.	n/a
S4R_FDF_381	The Framework shall log the maximum execution time of the processes per hour.	n/a
S4R_FDF_382	The Framework shall log the average execution time of the processes per hour.	n/a
S4R_FDF_383	The Framework shall log if any of its processes does not meet its deadline.	n/a
S4R_FDF_384	The Framework shall log if the integrity of the memory space of a partition has an error.	n/a
S4R_FDF_385	The Framework shall log if the integrity of the configuration file of the Framework has an error.	n/a
S4R_FDF_386	The Framework shall log if the coherency of the configuration file of the Framework has an error.	n/a
S4R_FDF_387	The Framework shall log if any unexpected external access is detected.	n/a
S4R_FDF_388	The Framework shall log if any not allowed external access is detected.	n/a
S4R_FDF_379	The log file shall follow the “report_yyyymmdd_xxx.log” naming convention, where yyyy, mm and dd stand for the system year, month and day and the xxx represents an incremental value in case more than one file with the same date exists.	n/a
S4R_FDF_389	The Framework must make a back up of the log files every day.	n/a
S4R_FDF_390	The Framework shall include a timestamp for each entry of the log file.	n/a
S4R_FDF_580	The Framework shall provide the application with the ability to add an entry in the event log.	n/a
S4R_FDF_581	The Framework shall allow the application to use the following logging levels for an entry: k) Debug l) Info m) Warning n) Error o) Fatal	n/a
S4R_FDF_582	The Framework shall provide the ability to export the current event log as a file with the following information per event log entry: • Identification of triggering entity • Type (logging level) • Event ID • Event message	n/a

Id	FDF requirement description	IMP requirements
	- Raw data	
S4R_FDF_565	**Deployment management** This subsection describes the requirements of the deployment management that enables to install and update configuration files and application executables of FDF partitions.	IMP-DPL-001
S4R_FDF_571	The Framework shall implement a secure file transfer such as FTPS or SFTP transfer protocols.	IMP-DPL-001
S4R_FDF_666	The Framework shall support debug operation and maintenance operation modes.	IMP-DPL-001
S4R_FDF_770	The Framework shall support maintenance of non-safety data using different structures than ones used for safety-related data.	IMP-DPL-001
S4R_FDF_567	The Framework shall provide maintenance staff with the ability to install executables on partitions train network, remote and direct connections.	IMP-DPL-001
S4R_FDF_566	The Framework shall provide maintenance staff with the ability to update executables on partitions train network, remote and direct connections.	IMP-DPL-001
S4R_FDF_573	The Framework shall provide maintenance staff with the ability to uninstall executables on partitions through train network, remote and direct connections.	IMP-DPL-001
S4R_FDF_568	The Framework shall provide maintenance staff with the ability to install configuration files through train network, remote and direct connections.	IMP-DPL-001
S4R_FDF_569	The Framework shall provide maintenance staff with the ability to update configuration files train network, remote and direct connections.	IMP-DPL-001
S4R_FDF_570	The Framework shall provide maintenance staff with the ability to uninstall configuration files train network, remote and direct connections.	IMP-DPL-001
S4R_FDF_635	The Framework shall provide the maintenance staff with a secure way to install executables on a partition.	IMP-DPL-001
S4R_FDF_639	The Framework shall provide the maintenance staff with a secure way to update executables on a partition.	IMP-DPL-001
S4R_FDF_640	The Framework shall provide the maintenance staff with a secure way to uninstall executables on a partition.	IMP-DPL-001
S4R_FDF_660	The Framework shall provide the maintenance staff with a secure way to install configuration files.	IMP-DPL-001
S4R_FDF_661	The Framework shall provide the maintenance staff with a secure way to update configuration files.	IMP-DPL-001
S4R_FDF_662	The Framework shall provide the maintenance staff with a secure way to uninstall configuration files	IMP-DPL-001
S4R_FDF_636	The Framework shall allow deleting persistently stored data and files with uninstalled executables.	IMP-DPL-001
S4R_FDF_658	The Framework shall provide detailed version information of FDF to maintenance staff.	IMP-DPL-001

Id	FDF requirement description	IMP requirements
S4R_FDF_663	The Framework shall provide detailed version information of each process (installed executable) to the maintenance staff.	IMP-DPL-001
S4R_FDF_665	The Framework shall provide detailed version information of each configuration file to the maintenance staff.	IMP-DPL-001
S4R_FDF_659	The Framework shall validate the executable code, schedule and the resource availability before the installation, during the installation and during updating it.	IMP-DPL-001
S4R_FDF_664	The Framework shall validate the configuration file before processing it or updating it to ensure that there is not conflict in the communication, schedule or resource availability of partitions and processes.	IMP-DPL-001
S4R_FDF_787	The Framework shall support concurrent re-configuration of partitions, guaranteeing that the re-configuration does not affect the remaining partitions. Those partitions may execute different and independent application functions with the same SIL level and to be hosted by one partition.	IMP-DPL-001
S4R_FDF_641	**File management** This subsection writes and reads files and variables that persist over device switch on and switch off cycles.	n/a
S4R_FDF_644	The Framework shall enable to create new files in memory.	n/a
S4R_FDF_645	The Framework shall allow opening existing files.	n/a
S4R_FDF_648	The Framework shall allow opening files in read-only (RO) or read/write (RW) modes.	n/a
S4R_FDF_649	The Framework shall allow writing data into a file.	n/a
S4R_FDF_650	The Framework shall allow reading data from a file.	n/a
S4R_FDF_651	The Framework shall allow storing files persist over device switch-on and switch-off cycles.	n/a
S4R_FDF_652	The Framework shall enable to remove files.	n/a
S4R_FDF_653	The Framework shall enable to persistently store variables over device switch-on and switch-off cycles.	n/a
S4R_FDF_654	The Framework shall allow loading variables which are persistently stored.	n/a
S4R_FDF_655	The Framework shall store variables in way that they can be accessed by a partition using a unique identifier. E.g., identify a value by a key.	n/a
S4R_FDF_656	The Framework shall guarantee that no variable or file corruption occurs if the device switches off while writing data to a variable or a file.	n/a
S4R_FDF_657	The Framework shall allow closing files.	n/a
S4R_FDF_171	**Non-functional requirements**	n/a

Id	**FDF requirement description**	IMP requirements
S4R_FDF_172	**Performance requirements**	n/a
S4R_FDF_299	The Framework shall guarantee methodology for performance analysis for considered system configurations.	n/a
S4R_FDF_300	The Framework shall guarantee methodology for system performance analysis in case of accidental situations.	n/a
S4R_FDF_301	The Framework shall define, configure, and assess performance of each node of system.	n/a
S4R_FDF_302	The Framework shall define, configure, and assess node performance for specified (cyber) security level.	n/a
S4R_FDF_303	The Framework shall define, configure, and assess node performance for I/O interface.	n/a
S4R_FDF_304	The Framework shall define, configure, and assess node performance for control algorithms and inter-partition communication.	n/a
S4R_FDF_305	The Framework shall define, configure, and assess node performance for logging and diagnostic subsystem.	n/a
S4R_FDF_306	The Framework shall define, configure, and assess node performance for communication interface.	n/a
S4R_FDF_307	The Framework shall define, configure, and assess performance of communication channels • channel priority • channel throughput	n/a
S4R_FDF_308	The Framework shall define, configure, and assess performance of communication channels for predefined parameters as: • jitter • latency • response time	n/a
S4R_FDF_309	The Framework shall define, configure, and assess performance for protection communication channels against cyber attack.	n/a
S4R_FDF_310	The Framework shall define, configure, and assess “performance for future use”: • data communication – capacity, throughput, security • control algorithms • fault tolerance	n/a
S4R_FDF_173	**Validation and verification support** The requirements in this subsection include all information regarding techniques used for testing purpose.	n/a
S4R_FDF_630	The Framework shall validate the installation or update of executable code before processing it. The scheduling and resources attached	S4R-IMP-020

Id	FDF requirement description	IMP requirements
	to other partitions shall not be affected.	
S4R_FDF_631	The Framework shall validate the installation or update of a configuration file before processing it. The communication, scheduling and resources of partitions and processes shall not be affected.	S4R-IMP-021
S4R_FDF_314	The Framework shall provide services to control and monitor the application execution (start, stop, synchronising to external trigger). I.e., using program flow monitoring techniques.	n/a
S4R_FDF_316	The Framework shall prevent the access to any validation and verification support service (fault injection and monitoring, forcing of outputs, monitoring of inputs and outputs, application control and monitoring, logging/tracing) on power up. The framework shall enable the validation and verification support services only on explicit request.	n/a
S4R_FDF_315	The Framework shall provide logging/tracing services for a selectable set of events related to • Fault injection and monitoring • Communication and shared network memory change • Output change • Input change • Application execution and monitoring	n/a
S4R_FDF_311	The Framework shall provide services to inject faults and monitor the fault reaction related to • non-critical (SIL0) • platform partitioning and isolation mechanism • communication (transmission, reception) and shared network memory • output control • input monitoring • application execution (timing, memory access, start, stop, throttling, …)	n/a
S4R_FDF_312	The Framework shall provide services to force the outputs to all states (valid and invalid) independent of the current control by the associated application.	n/a
S4R_FDF_313	The Framework shall provide services to monitor the state of all outputs and inputs independently from the application that is associated to the respective inputs/outputs.	n/a
S4R_FDF_174	**Interface requirements**	n/a
S4R_FDF_701	The Framework shall offer an interface to allow registering a variable that can be monitored externally.	n/a
S4R_FDF_702	The Framework shall offer an interface to allow external devices to request the list of variables which can be monitored.	n/a
S4R_FDF_703	The Framework shall offer an interface to allow external devices to request monitoring a number of variables with a given frequency.	n/a

Id	FDF requirement description	IMP requirements
S4R_FDF_706	The Framework shall provide an interface between input and output variables of processes executed in partitions - on the same device - on different devices in the same consist or - on devices in different consists of the same train according to their defined inputs and outputs.	n/a
S4R_FDF_707	The Framework shall provide an interface between variables provided by I/O devices to inputs of processes executed in partitions - on the same device - on another device in the same consist or - in another consist of the same train according to the input definition of the partitions.	S4R-IMP-002
S4R_FDF_708	The Framework shall provide an interface between variables provided by a process executed on a partition to variables controlling outputs of I/O devices located - on the same device - on another device in the same consist or - in another consist of the same train according to the interface definition between the partition and the I/O device.	S4R-IMP-002
S4R_FDF_712	The Framework shall offer an interface to external devices to force variables.	n/a
S4R_FDF_713	The Framework shall offer an interface to register variable that can be forced.	n/a
S4R_FDF_734	The Framework shall guarantee the independence of I/O interfaces that can be requested by the application function.	n/a
S4R_FDF_175	**Application** The requirements in this section describe the interface requirements between applications and the framework.	n/a
S4R_FDF_318	The Framework shall offer an interface to create time-triggered processes.	n/a
S4R_FDF_320	The Framework shall offer an interface to set the priority of a process.	n/a
S4R_FDF_321	The Framework shall offer an interface to set the deadline of a process.	n/a
S4R_FDF_322	The Framework shall offer an interface to set the period of a time-triggered process.	n/a
S4R_FDF_323	The Framework shall offer an interface to set the offset of a time-triggered process.	n/a
S4R_FDF_324	The Framework shall offer an interface to set the activation events of an event-triggered process.	n/a
S4R_FDF_325	The Framework shall offer an interface to create periodic timers.	n/a
S4R_FDF_326	The Framework shall offer an interface to create sporadic timers.	n/a

Id	**FDF requirement description**	IMP requirements
S4R_FDF_327	The Framework shall offer an interface to set the deadline of a timer.	n/a
S4R_FDF_328	The Framework shall offer an interface to start a timer.	n/a
S4R_FDF_329	The Framework shall offer an interface to stop a timer.	n/a
S4R_FDF_330	The Framework shall offer an interface to create partitions.	n/a
S4R_FDF_331	The Framework shall offer an interface to set the offset of a partition.	n/a
S4R_FDF_332	The Framework shall offer an interface to set the period of a partition.	n/a
S4R_FDF_333	The Framework shall offer an interface to set the budget of a partition.	n/a
S4R_FDF_334	The Framework shall offer an interface to set the processes of a partition.	n/a
S4R_FDF_335	The Framework shall offer an interface to create events.	n/a
S4R_FDF_336	The Framework shall offer an interface to launch an event.	n/a
S4R_FDF_337	The Framework shall offer an interface to discover, monitor and control the applications it executes.	n/a
S4R_FDF_501	The Framework shall offer an interface to read static configuration from a file.	n/a
S4R_FDF_176	**I/O** The requirements in this section describe the inputs and outputs of the Framework.	n/a
S4R_FDF_338	The Framework shall offer an interface to read the type and number of input and output ports.	n/a
S4R_FDF_339	The Framework shall offer an interface to read analog inputs.	n/a
S4R_FDF_340	The Framework shall offer an interface to read digital inputs.	n/a
S4R_FDF_341	The Framework shall offer an interface to write analog outputs.	n/a
S4R_FDF_342	The Framework shall offer an interface to write digital outputs.	n/a
S4R_FDF_343	The Framework shall offer an interface to map a variable to each analog or digital input or output.	n/a
S4R_FDF_344	The Framework shall offer an interface to determine the type, size and optional scaling/units of variables mapped to analog inputs and outputs.	n/a
S4R_FDF_345	The Framework shall offer an interface to determine the type, size and bit usage of variables mapped to digital inputs and outputs.	n/a
S4R_FDF_346	The Framework shall offer an interface to set the update cycle (multiple of basic cycle) for each mapped variable.	n/a

Id	FDF requirement description	IMP requirements
S4R_FDF_347	The Framework shall be able to map digital or analog input or output ports to data types complying with IEC 61375-2-1 [7] and IEC 61375-2-3 [2].	n/a
S4R_FDF_779	The Framework shall support at least 14 analog inputs with 12 bit resolution, 1 digital output and 7 digital outputs. If the controller does not support such capabilities, alternative peripherals shall be provided (e.g., SPI).	n/a
S4R_FDF_177	**Network** Network interfacing to COM/Middleware	n/a
S4R_FDF_348	For outgoing messages to the network, The network interface device shall read the message data from the partition message memory.	n/a
S4R_FDF_489	Application shall place message data into the partition message memory which is per configuration aligned with queuing or sampling ports.	n/a
S4R_FDF_349	For incoming messages from the network, the network interface device shall write the message data to the partition message memory.	n/a
S4R_FDF_490	Application shall read message data from the partition message memory which is per configuration aligned with queuing or sampling ports.	n/a
S4R_FDF_350	The configuration of the Framework and the Network shall specify for each port whether it is operated as a queuing or sampling port.	n/a
S4R_FDF_351	The configuration of the Framework (software abstraction / COM / middleware layer) shall define which data is stored into the message and at what point in time the message is published to the network.	n/a
S4R_FDF_352	The configuration of the Framework and the Network shall be consistent with regards to which frames are sent and received, at which times.	n/a
S4R_FDF_353	The Framework shall be able to receive status and errors related to message transmission in the network interface.	n/a
S4R_FDF_178	**Safety requirements**	n/a
S4R_FDF_180	**EC directive**	n/a
S4R_FDF_391	**EC Train Directive** DIRECTIVE (EU) 2016/797 [3] on the interoperability of the rail system within the European Union. Relevant chapters of Annex III of the directive: • 1.1.1 General requirements/Safety • 1.5 General requirements/Technical compatibility	n/a

Id	FDF requirement description	IMP requirements
	• 2.3.1 Control-command and signalling/Safety • 2.4.1 Rolling stock/Safety • 2.4.2 Rolling stock/Reliability and availability • 2.4.3 Rolling stock/Technical compatibility	
S4R_FDF_392	**TSI LOC&PAS** 1302/2014/CE - COMMISSION REGULATION (EU) No 1302/2014 of 18 November 2014 [4]. Relevant chapters: • 4.2.4.2.1. (3), (4) Functional requirements • 4.2.4.2.1. (11) Functional requirements • 4.2.4.3 (1)/(2) Type of brake system • 4.2.4.10. (3) Brake requirements for rescue purposes • 4.2.5.2. (2), (3) Audible communication system • 4.2.5.3.1 (2) Passenger alarm/General	n/a
S4R_FDF_643	**Security requirements** This subsection defines the security-related requirements of FDF.	S4R-IMP-023
S4R_FDF_414	The framework shall secure the incoming/outgoing communication (channel) to the ECUs (Electronic Control Units) against security threats with regards to confidentiality, authenticity, integrity and availability whilst respecting real-time constraints (i.e. predictable latency and low jitter).	n/a
S4R_FDF_416	The framework shall protect stored data against adversaries (with regards to confidentiality, authenticity and data integrity).	n/a
S4R_FDF_417	The framework shall include a mechanism in order to prevent unknown/unexpected traffic (i.e. admission and access control).	n/a
S4R_FDF_420	The framework shall accomplish the need of protecting the data and state of the functions during execution on an ECU.	n/a
S4R_FDF_667	The Framework shall support cryptography algorithms, key sizes and mechanisms to key establishment and management according to common security industry practises and recommendations.	n/a
S4R_FDF_412	The framework shall provide cryptographic mechanisms and handle cryptographic objects • Ensure framework's security as well as framework's communication channel (receiving and transmitting role) by means of	n/a

Id	FDF requirement description	IMP requirements
	secure cryptographic algorithms • Management of cryptographic keys (creation, deletion and retention) • Calculation of cryptographic functions (digital signatures, MACs, encryption/decryption)	
S4R_FDF_646	The Framework shall support data encryption.	n/a
S4R_FDF_647	The Framework shall support data decryption.	n/a
S4R_FDF_409	The framework shall operate accordingly/with regards to confidentiality • Ensure that data inside the framework cannot be read by an unauthorised entity: ensure non-disclosure of information/data towards entities (i.e. users, processes, and device) unless a successful access authorisation.	n/a
S4R_FDF_410	The framework shall operate accordingly/with regards to authenticity • Assurance of entities' identity • Ensure/verify data source: information/data comes from a verified and trusted entity (sender) • Information collected by the framework should be authentic with respect to origin and time if the framework performs actions based on that information • The author of the message, respectively the origin sending entity of the information/data, shall be evident and traceable at any time (with regards to non-repudiation)	n/a
S4R_FDF_415	The Framework shall support availability of access control in the network to ensure robustness to DoS attacks as well as side-channel attacks.	n/a
S4R_FDF_429	The framework shall ensure that security policy enforcement functions and the data that configures them cannot be modified without authorisation.	n/a
S4R_FDF_418	The framework shall support secure storage for key(s) and trust anchor(s) for secure authentication and communication (with regards to security services and authenticity).	n/a
S4R_FDF_419	The framework shall operate with authenticated entities (ECUs, SW/HW components) only (with regards to authenticity) • The framework shall enforce authenticity and integrity of the ECUs in order to meet/fulfil framework's security requirements. • The framework shall enforce authenticity and integrity of the software components in order to meet/fulfil framework's security requirements.	n/a
S4R_FDF_669	The Framework shall allow to assign privileges to authenticated users (access rights).	n/a
S4R_FDF_670	The Framework shall support executable identification and authentication.	n/a

Id	**FDF requirement description**	IMP requirements
S4R_FDF_671	The Framework shall allow to assign privileges to authenticated executables (access rights).	n/a
S4R_FDF_672	The Framework shall: • initialise authenticator content • change all default authenticators upon control system installation • change/refresh all authenticators • protect all authenticators from unauthorised disclosure and modification when stored and transmitted.	n/a
S4R_FDF_673	The Framework shall support the management of identifiers by users, groups, roles or control system interfaces.	n/a
S4R_FDF_749	The component "Security Management" shall be able to support the management of all accounts by authorized users, including adding, activating, modifying, disabling and removing accounts.	n/a
S4R_FDF_674	The Framework shall enforce configurable password strength based on minimum length and variety of character types.	n/a
S4R_FDF_413	The framework shall provide a Public Key Infrastructure (PKI) • Support/ensure the authentication process of entities (with regards to authenticity) • Management of certificates (retention and update)	n/a
S4R_FDF_676	The Framework shall validate certificates by: • checking the signature of given certificates • constructing a certification path to an accepted CA • deploying leaf certificates to all hosts which communicate with the subject to which the certificate is issued (in the case of self signed certificates) • checking the certificate's revocation.	n/a
S4R_FDF_677	The Framework shall: • establish user (human, SW process, device) control of the private keys • map the authenticated identity to a user (human, SW process, device).	n/a
S4R_FDF_678	The Framework shall be able to obscure feedback authentication information during authentication process.	n/a
S4R_FDF_679	The Framework shall enforce a limit of configurable number of consecutive invalid access attempts by any user (human, SW, device) during a configurable time period.	n/a
S4R_FDF_680	The Framework shall deny access for specified period of time or until unlocked by an administrator when the access attempts number is exceeded.	n/a
S4R_FDF_681	The Framework shall display a system notification message before authenticating. This message shall only be configurable by authorised	n/a

Id	FDF requirement description	IMP requirements
	users.	
S4R_FDF_430	The Framework shall provide the capability to detect, generate and export audit records for security relevant auditable events.	n/a
S4R_FDF_730	The Framework shall periodically verify the correct operation of security protection functions and notify system administrator when anomalies are discovered.	n/a
S4R_FDF_411	The Framework shall operate accordingly/with regards to data integrity • Support/offer mechanism(s) in order to ensure data integrity for information collected within the framework. • Ensure that the information has/have not been modified either in transit or in storage on the route from the sender's entity to the receiver's entity.	n/a
S4R_FDF_421	The framework shall accomplish the need of protecting the data and state of the functions during execution within software components.	n/a
S4R_FDF_422	The framework shall ensure the data isolation between different partitions created and maintained by the framework so that the data in a partition is accessible only by code running in that partition (SIL).	n/a
S4R_FDF_423	The framework shall ensure the isolation of the resource between different partitions created and maintained by the framework so that the resources exported by the framework into a partition are accessible only by code running in that partition (with SIL).	n/a
S4R_FDF_424	The framework shall provide information flow control that enforces strict partition isolation so that only explicitly configured interaction are allowed.	n/a
S4R_FDF_425	The framework shall ensure that a failure in one partition is not propagated to other partitions.	n/a
S4R_FDF_426	The framework shall ensure that an attack affecting one partition is not propagated to other partitions.	n/a
S4R_FDF_427	The framework shall ensure that security policy enforcement functions cannot be bypassed.	n/a
S4R_FDF_428	The framework shall ensure that security policy enforcement functions are always invoked.	n/a
S4R_FDF_731	The Framework or its support utilities shall provide user functionality to facilitate creation of backups of user-level and system-level information (including system security state information).	n/a
S4R_FDF_732	The Framework shall provide user functionality to allow be recovering and reconstituting to previously saved Backup after a disruption or failure.	n/a
S4R_FDF_182	**RAMS requirements**	n/a
S4R_FDF_478	The Framework shall provide a safe communication path for transmission/reception of datasets using a safety layer.	S4R-IMP-016
S4R_FDF_479	The Framework shall offer application interfaces according to the safety layer needed:	S4R-IMP-017

Id	FDF requirement description	IMP requirements
	•non-critical (SIL0) •SIL2 •SIL4 where the ability to provide SIL2 and SIL4 APIs depends on the specific implementation of the framework (on HW/SW).	
S4R_FDF_480	The Framework shall guarantee the integrity and validity of the received data to meet the requirements for SIL2 (according to IEC61508-1 [5]). SDTv2, as defined in IEC61375-2-3 Annexe B [2], provides this safety level for PFH ≥ 10E-7 < 10E-6 (1% for black channel communication).	S4R-IMP-018
S4R_FDF_481	The Framework shall guarantee the integrity and validity of the received data to meet the requirements for SIL4 (according to IEC61508-1). A PFH ≥ 10E-9 < 10E-8 (1% for black channel communication) is needed.	S4R-IMP-018
S4R_FDF_482	The Framework shall inform the application of communication losses, which enable the application to decide whether to set the system into the 'safe state'.	n/a
S4R_FDF_483	The Framework shall monitor the operational state of the ECU (and its function(s)) by appropriate means and report in case of failure. I.e., implementing error detection and correction (EDC) technique.	n/a
S4R_FDF_484	The Framework shall share its operational state with all other ECUs in its functional group(s).	n/a
S4R_FDF_485	The Framework shall detect and verify the operational status of other redundant ECUs.	n/a
S4R_FDF_486	The Framework shall inform the application of the operational status of all other ECUs in its functional group(s).	n/a
S4R_FDF_487	The Framework shall be operational within 60 seconds from power-up.	n/a
S4R_FDF_488	The Framework shall perform a self-test of the ECU on power-up.	n/a
S4R_FDF_467	**Configuration management**	n/a
S4R_FDF_431	The Framework shall be configurable on ECU reset or start-up by a local configuration.	n/a
S4R_FDF_432	The Framework shall be able to receive an additional remote configuration via network.	n/a
S4R_FDF_433	The Framework shall check the validity and integrity of any configuration. This could be a CRC, MD or signature created by tooling.	n/a
S4R_FDF_434	The Framework shall check the origin of remote configurations and ignore false configurations.	n/a

Id	FDF requirement description	IMP requirements
	Remote configurations must be certified.	
S4R_FDF_435	The remote configuration's properties shall take precedence over the same properties of the local configuration. This relates to dynamic vs. static configuration, e.g. direction dependent addressing and default parameters.	S4R-IMP-029
S4R_FDF_436	The Framework shall provide a local interface to retrieve static and dynamic configuration properties by a host application.	n/a
S4R_FDF_437	The Framework shall provide a remote (network) interface to retrieve static and dynamic configuration properties of an ECU.	n/a
S4R_FDF_438	The Framework's local configuration shall define the necessary properties for local communication needs. Note: Annex C of IEC 61375-2-3 [2] defines an XML format which covers most properties of a communication framework. Train-wide communication depends on train inauguration and may therefore not be possible with local configurations, only. This depends on the future network layout (defined in WP1).	n/a

Table 10: FDF requirements vs. Integrated Modular Platform requirements defined in D1.11- Traceability Matrix