




ERJU SYSTEM PILLAR

# **System Interface Description SDI-DS information model (SERA Version)**



# System Interface Description SDI-DS information model (SERA Version)

Author(s)	Karl-Albrecht Klinge
Abstract	This document describes the as required per SPPRAMSS-349 - <a href="#">EN 50126-1:2017</a> phase 5 (Architecture and apportionment of system requirements) between the and the .
Config Item	System Interface Description
Document ID	TCCS Service Function Diagnostics _SFD_ L5/TCCS System Interface SDI-DS information model_SERA Version#723864  System Interface Description SDI-DS information model (SERA Version)
Classification	Public
Status	In Review by System Pillar
Version	1.0
Revision	723864
Last Change Date	02.10.2025
Copyright	Brussels: Europe's Rail Joint Undertaking, 2025

© Europe's Rail Joint Undertaking, 2025

This document is drafted by and belongs to EU Rail.

EU Rail encourages the distribution and re-use of this document, the technical specifications and the information it contains. EU Rail holds several intellectual property rights, such as copyright and trade mark rights, which need to be considered when this document is used.

EU Rail authorises you to re-publish, re-use, copy and store this document without changing it, provided that you indicate its source and include the following: EU Rail trade mark, title of the document, year of publication, version of document.

EU Rail makes no representation or warranty as to the accuracy or completeness of the information contained within these documents. EU Rail shall have no liability to any party as a result of the use of the information contained herein. EU Rail will have no liability whatsoever for any indirect or consequential loss or damage, and any such liability is expressly excluded.

You may study, research, implement, adapt, improve and otherwise use the information, the content and the models in the this document for your own purposes. If you decide to publish or disclose any adapted, modified or improved version of this document, any amended implementation or derivative work, then you must indicate that you have modified this document, with a reference to the document name and the terms of use of this document. You may not use EU Rail's trade marks or name in any way that may state or suggest, directly or indirectly, that EU Rail is the author of your adaptations.

EU Rail cannot be held responsible for your product, even if you have used this document and its content. It is your responsibility to verify the quality, completeness and the accuracy of the information you use, for your own purposes.


**This work is currently a work in progress. The content presented is subject to change as it undergoes further review, refinement, and development. Please do not consider this version as final or authoritative.**

INFO: History table is not displayed, because this document is in status **doc\_contentApproval**.


RULE: History table is not displayed, in statuses: { draft doc\_open doc\_inprogress doc\_contentApproval doc\_contentDecision }

CONTACT: For more information contact Administrator

## Review description

Approvals	Kilian Jens : Waiting , LOSTUN Virgil : Waiting , SANGO Marc (SNCF / DIR TECHNOLOGIES INNOVATION ET PROJETS GROUPE / IR DIR RECHERCHE - PSF) : Waiting , DE NICOLA, Giuseppe : Waiting , KEFALAS, Georgios : Waiting , Julien Bois : Waiting , Oliver Knapp : Waiting , Wischy, Markus Alexander (SMO RI R&D F IL) : Waiting , HENON Frédéric : Waiting , TEKE, Emre : Waiting , Renato Rodrigues : Waiting , IOVINO, Salvatore : Waiting , Davinder Bhatia : Waiting , BITSCH Friedemann : Waiting , Roman R Treydel : Waiting , Golebniak, Udo (SMO RI ML ADC I&C) : Waiting , Mirko Blazic : Waiting , Benameur, Malik (SMO NEE RC-CH RI PLM SYS) : Waiting , MOTTOLA, Giuseppe Diodato : Waiting , Jack Schneider : Waiting , Zeeshan Z Ansar : Waiting , Patrick Konix : Waiting , NANNI Marco : Waiting , DE MARCO TELESE Giancarlo : Waiting , Tione, Roberto : Waiting , Andreeva-Moschen Emilia (HOLDING) : Waiting
Type of Approval	 Document Review
Attachments	<a href="#">REMINDER_[ERJU SP] Request to review SC2.4 List of deliverables - Task 2_ Transversal Systems .pdf</a> , <a href="#">Review and Approval Jens Kilian.pdf</a> , <a href="#">Review and Approval Virgil Lostun.pdf</a>

## Approval description

Approvals	LOSTUN Virgil : Waiting
Type of Approval	 Document Approval
Attachments	<a href="#">REMINDER_[ERJU SP] Request to review SC2.4 List of deliverables - Task 2_ Transversal Systems .pdf</a> , <a href="#">Review and Approval Jens Kilian.pdf</a> , <a href="#">Review and Approval Virgil Lostun.pdf</a>

1 Preamble	4
1.1 Scope and intended audience	4
1.2 Purpose	4
1.3 Glossary	5
2 Overview	5
2.1 Overall description	5
2.2 Functional requirements	5
2.3 Non-functional characteristics / non-functional requirements	5
3 SDI-DS interface requirements	5
3.1 Data description	6
3.2 Behaviour description	6
3.3 Interdependencies to other interface layers	6
4 Appendix	6
4.1 Input documents	6
4.2 Standards and References	6
5 Workspace for discussions, actions and issues	6
6 Scope of interface constraints	7
6.1 Role of interface	7

## 1 Preamble

### 1.1 Scope and intended audience

This document defines the requirements for Standard Data Interface (SDI) to the aggregated namespace of the Service Function Diagnostics based on the OPC UA protocol, as outlined in Part 1 of the SDI Companion Specification.

This specification is derived from and aligned with the following key documents:

- The System Definition, which describes the overall architecture and objectives of the SDI-based integration approach.
- The System Requirements Specification, which outlines functional, operational, and non-functional expectations for SDI implementations.
- The System Architecture Definition, which provides the structural blueprint for how system components interact, communicate, and are organized.

Intended Audience:

- System Architects and Integrators, responsible for the design and alignment of interoperable interfaces and data models across systems.
- Product Manufacturers and Suppliers, especially those contributing equipment or software components that expose OPC UA-based interfaces according to SDI standards.
- OPC UA Modelers and Developers, implementing companion models in NodeSet2 XML, transforming UML diagrams into OPC UA Information Models, and maintaining semantic consistency.
- Operators and Maintainers, who rely on standardized and interpretable data interfaces for diagnostics, monitoring, and operational decision-making.
- Standards Bodies and Certification Authorities, reviewing and validating SDI-based models for compliance, compatibility, and long-term maintainability.

The scope of this document is limited to general modeling and communication requirements, including semantic standardization, reference modeling, versioning strategies, and structural adaptation rules for mapping domain-specific models into OPC UA. It does not cover the detailed structure of the NodeSet2 files or product-specific information models, which are provided in separate parts of the companion specification (e.g., SDI-GEN and SDI-XX documents).

### 1.2 Purpose

This document describes the Standard Data Interface (SDI) DS in accordance with the requirements of – EN 50126-1:2017, Phase 5: Architecture and Apportionment of System Requirements. The described interface defines the communication and data exchange between and , forming a fundamental part of the system architecture.

The purpose of this document is to provide:

- A general but sufficiently detailed description of the system interface;
- Justification for the architectural and design decisions taken;
- A clear explanation of how the interface contributes to meeting the overall system requirements;
- A basis for consistent implementation, integration, and verification activities across different suppliers or subsystems.

This document aims to ensure a shared understanding of the architectural principles and interface structure, focusing on the rationale that led to the selected approach. It supports traceability from high-level system requirements to concrete technical solutions, particularly those realized using OPC UA.

**Note:** This Interface Definition may cover specific layer(s) of the overall system interface stack where appropriate, allowing reuse across multiple subsystems or interface definitions.

Depending on the context:

- For external interfaces, it serves as an extension of the System Definition, describing interactions from the perspective of the System Under Consideration (SuC).
- For internal interfaces, it complements the System Architecture Document, detailing the internal communication structure within the SuC.

This document is a key input for the development of OPC UA companion specifications and information models (e.g., SDI-GEN and SDI-XX), and provides the foundational guidance for implementing semantically consistent OPC UA-based interfaces across the system.

### 1.3 Glossary

This section provides definitions for key terms and abbreviations used throughout this document. All definitions are based on the official System Pillar Glossary. To ensure consistency and traceability, terms referenced here are aligned with existing definition work items and maintained according to the Glossary Usage Guidelines.

<b>SDI</b> – Standard Diagnostic Interface	A standardized OPC UA-based interface framework for the integration of diagnostic, condition monitoring, and asset-related data across different subsystems and suppliers.
<b>OPC UA</b> – OPC Unified Architecture	A machine-to-machine communication protocol for industrial automation, designed for interoperability, scalability, and platform-independence.
<b>NodeSet2</b>	The XML-based format used to describe OPC UA Information Models for exchange between tools and implementations.

## 2 Overview

### 2.1 Overall description

The interface is based on OPC UA defining the application layer and using the lower levels of the TCP/IP stack.

The SDI-XX requirements apply.

### 2.2 Functional requirements

The SDI-DS must be capable of holding and utilizing the union of all SDI-XX NodeSet2 models present in the namespaces on the aggregated building blocks, as well as the vendor-specific extended models used. This also includes multiple simultaneous versions of these SDI-XX NodeSet2 models.

The SDI-DS includes a model for connection management. That allows to track the status of each of the OPC UA connections being aggregated in the Service Function Diagnostics.

*ToDo: The information model for this will be specified in the next remit phase.*

### 2.3 Non-functional characteristics / non-functional requirements

Decisions and rationales

## 3 SDI-DS interface requirements

The  SPT2TS-129409 - [System Structure and Interface Modeling](#) apply [SPT2TS-130217 ]

### 3.1 Data description

### 3.2 Behaviour description

see OPC UA

- 

### 3.3 Interdependencies to other interface layers

Analysis of dependencies between levels like time-out values among OSI layers, disconnection detection and reconnection, ...

DK: Consideration of Service access points (vertical between OSI layers), APIs? This might be too detailed in some cases.

## 4 Appendix

### 4.1 Input documents

### 4.2 Standards and References

## 5 Workspace for discussions, actions and issues

Define and review template for the System Interface Definition deliverable [SPT2TS-129430]

### Discussion topic: interface definition vs. interface specification

Visualisation:

<Image: diagram\_20240710-1712.51421.mxd>

#### Approach 1: interface definition (without requirements)

only static + dynamic definitions and descriptions for System A and B (example: protocol description, overview, protocol stack definition (example TACS SCI) --> statement of facts, no "shall" -> no requirements

#### Approach 2: interface specification (with requirements)

same as approach 1 but with shall statements for each System -> with requirements for System A and System B regarding the interface A <-> B)

#### General

- There should be no redundancy, means for example that behaviours (e.g. data exchanges, handshakes etc.) are not described here and in the requirements specification again.

**Futher evaluation** (in relation with Polaron work item content and outputs in Capella model explorer):

- recommendation use approach 1 (better because requirements are then placed at once and distributed across multiple documents which are referring)
- TACS / EULYNX uses already approach 1 in principle, easier to integrate them at later stage
- show exchange items + all the detailed of their exchange item elements (unit, data type, multiplicity, ranges, ...)

- one Polarion document per component exchange (or physical link?)?
- considering the use PVMT for attributing OSI layers
- focus on application payload, maybe not lower layers in all cases, just refer then to lower layer specs

## Meeting 2024-08-01 (Dennis, Sayfeddine , Gilles)

Let's try first **approach 2.**

Rationale:

- one single point of truth (even if it is splitted into two documents, definition and specification to hold the requirements) At least all subsystems should refer to the same (set of) documents for a given interface.
- Still called it interface definition for now and see how many requirements to be included and where to stop. Important: avoid redundancy of information between interface def. and req. spec. document Requirements for the interface def. would be: general performance values, master and slave roles, sequencing between the two interfacing systems (e.g. handshakes). The req. specs. would be then refer to this, but only for each corresponding side.
- Some requirements may be included in this documents to have a middleway to establish traceability of system requirements and (or) with the system design satisfies these requirements (to be elaborated in detail).

### Glossary definitions to be considered:

 SPT2OD-6831 - FFFIS - Form Fit Functional Interface Specification

 SPT2ARC-1015 - FORM FIT FUNCTIONAL INTERFACE SPECIFICATION

 SPLI-1157 - FUNCTIONAL INTERFACES SPECIFICATION

## 6 Scope of interface constraints

### 6.1 Role of interface

DK: If there is the case that there will be a new protocol developed and defined by a domain for e.g. the application layer, should we give the possibility to move it to a separate document (e.g. protocol definition) which is referenced here like the standards? In case of new and detailed content, maybe one single interface definition gets to big.