Operational Interoperability of S4RIS (SAFETY4RAILS Information System) and Logistics

Deliverable 6.1

Lead Author: Fraunhofer

Contributors: NCSRD, *ERARGE, UNEW*

Dissemination level: PU - Public

Security Assessment Control: passed

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 883532.

D 6.1 Operational interoperability of S4RIS and logistics

Deliverable number:	6.1	
Version:	1.2	
Delivery date:	29/03/2022	
Dissemination level:	PU - Public	
Nature:	Report	
Main author(s)	Andreas Frorath, Katharina Ross, Christoph Brockt, Andreas Weber	Fraunhofer
Contributor(s) to main deliverable production	Stephen Crabbe (through multiple reviews)	Fraunhofer
	Konstantinos Panou	NCSRD
	Alper Kanak, Salih Ergün, Niyazi Uğur, Sercan Tanrıseven, Ünal Ergün, S.Halit Ergün	ERARGE
	Raphael David	UNEW
Internal reviewer(s)	Stephen Crabbe, Malte von Ramin, Katharina Ross	Fraunhofer
	Atta Badii	UREAD
	Antonio Santiago de Laporte	MDM
	Emmanuel Matsika, Raphael David	UNEW
	Alper Kanak	ERARGE
	Andreas Georgakopoulos	WINGS
	Uli Siebold	IC
	Emiliano Costa, Fabio Bolletta	RINA
External reviewer(s)	Thomas Chatelet	Member of EAB

Document control

Version	Date	Author(s)	Change(s)
0.1	30/06/2021	Fraunhofer	Table of contents
0.2	15/07/2021	Fraunhofer	1st draft version
0.3	28/072021	Fraunhofer	Internal review for consortium
0.4	04/08/2021	Fraunhofer	Feedback from consortium integrated regarding the open questions
0.5	21/12/2021	Fraunhofer	Check for missing contributions of partners
0.6	18/01/2022	NCSRD, ERARGE	Contributions inserted
0.7	19/02/2022	Fraunhofer	First internal review by Fraunhofer
0.8	25/02/2022	Fraunhofer, IC, UNEW	Second internal review by Fraunhofer, IC and UNEW
0.9	01/03/2022	ERARGE	Third internal review by ERARGE
0.10	09/03/2022	Fraunhofer, UREAD, MDM, UNEW, WINGS	Internal review by UREAD, MDM, UNEW, WINGS, Fraunhofer
0.11	14/03/2022	Fraunhofer, IC, RINA	Last feedback from IC, Fraunhofer, RINA
0.12 / V1.0	22/03/2022	Fraunhofer	Response to latest comments by Fraunhofer-SC
V1.1	28/03/2022	Fraunhofer	Updates to cover some remaining comments and creation of V1.1
V1.2	29/03/2022	Fraunhofer, MDM	Last updates from Fraunhofer and MDM and creation of final Version V1.2

DISCLAIMER AND COPYRIGHT

The information appearing in this document has been prepared in good faith and represents the views of the authoring organisation(s). Every effort has been made to ensure that all statements and information contained herein are accurate; however, the authoring organisation(s) accept no statutory, contractual or other legal liability for any error or omission to the fullest extent that liability can be limited in law.

This document reflects only the view of its authoring organisation(s). Neither the Research Executive Agency, nor the European Commission are responsible for any use that may be made of the information it contains. The use of the content provided is at the sole risk of the user. The reader is encouraged to investigate whether professional advice is necessary in all situations.

ABOUT SAFETY4RAILS

SAFETY4RAILS is the acronym for the innovation project: **Data-based analysis for SAFETY and security protection FOR detection, prevention, mitigation and response in trans-modal metro and RAILway networkS**. Railways and Metros are safe, efficient, reliable and environmentally friendly mass carriers, and they are becoming even more important means of transportation given the need to address climate change. However, being such critical infrastructures turns metro and railway operators as well as related intermodal transport operators into attractive targets for cyber and/or physical attacks. **The SAFETY4RAILS project delivers methods and systems to increase the safety and recovery of track-based inter-city railway and intra-city metro transportation.** It addresses both cyber-only attacks (such as impact from WannaCry infections), physical-only attacks (such as the Madrid commuter trains bombing in 2004) and combined cyber-physical attacks, which are important emerging scenarios given increasing IoT infrastructure integration.

SAFETY4RAILS concentrates on rush hour rail transport scenarios where many passengers are using metros and railways to commute to work or attend mass events (e.g. large multi-venue sporting events such as the Olympics). When an incident occurs during heavy usage, metro and railway operators have to consider many aspects to ensure passenger safety and security, for example carry out a threat analysis, maintain situation awareness, establish crisis communication and response, and they must ensure that mitigation steps are taken and communicated to travellers and other users. **SAFETY4RAILS will improve the handling of such events through a holistic approach.** It will analyse the cyber-physical resilience of metro and railway systems and deliver mitigation strategies for an efficient response, and, in order to remain secure given everchanging novel emerging risks, it will facilitate continuous adaptation of the SAFETY4RAILS solution; this will be validated by two rail transport operators and the results will support the re-design of the final prototype.

TABLE OF CONTENT

List of tables

List of figures

Executive summary

This document is Deliverable D6.1 - Operational Interoperability of S4RIS (SAFETY4RAILS Information System) and Logistics. This document aims to provide a technical concept of the interoperability and security for the SAFETY4RAILS Information System (S4RIS). The S4RIS integrates the developed software systems of each tool provider into a comprehensive platform. This deliverable presents three main aspects of the S4RIS and also takes into account the outcome from previous deliverables like D 1.4[1], D 2.1[2], D 2.3[3] or D 2.4[4].

The first main topic is refining the previously designed architecture discussed in D 2.3 and D1.4. This refinement regards the Data Source Layer, that includes all data providers regardless of whether the data was persisted in databases or is streamed from the sensor stations. Therefore, this layer contains the databases of each tool and the decryption tool PRIGM provided by ERARGE. For further reading see Chapter 3.2. Comparing the architecture presented in D2.3 and D1.4 this representation of the data source layer combines the presentation of the separate Source and Storage Layers (their separate presentation remains valid depending on the tools and data involved and as presented in D2.3 and D1.4).

The second main topic is the interoperability of the tools. The tools exchange information mainly via Apache Kafka.

The third main topic is the security concept of the S4RIS. This includes security aspects like how the communication can be encrypted and how the IT infrastructure is being secured.

[1] Siebold, U, Crabbe, S. (2021): D1.4 – Specification of the overall technical architecture. SAFETY4RAILS internal document.

[2] Ferrando, F., De Belloy, F. (2021): D 2.1 – Grid analysis of end-user needs and Workshop minutes. SAFETY4RAILS internal document.

[3] Panou, K., Argyriou, L. (2021): D 2.3 – System specifications and concept architecture. SAFETY4RAILS internal document.

[4] Crismer, F., Macchi, L. (2021): D 2.4 – Specific requirements for standardization and interoperability. SAFETY4RAILS internal document.

1. Introduction

1.1 Overview

A major aim of the SAFETY4RAILS project is to integrate the further developed software systems of each tool provider into a comprehensive platform: the SAFETY4RAILS Information System (S4RIS). The S4RIS platform is characterized by two main aspects:

- On the one hand, the platform provides the infrastructure to exchange information between different tools
- On the other hand, it provides end-users with a central information system for decision support and insights for mitigating security threats.

In this context, the deliverable D 6.1 “Operational Interoperability of S4RIS and Logistics” is intended to offer a technical concept of the interoperability and the security for the SAFETY4RAILS Information System (S4RIS). The focus is on the secure integration of tools and sensor stations. This includes security aspects such as how the communication and data can be encrypted and how the IT infrastructure is being secured. The approach described in the deliverable builds on the requirements, specifications and architectural solution described in the earlier deliverable reports D1.4 and D2.3.

1.2 Structure of the deliverable

This deliverable D6.1 starts with an overview of the prerequisites with relevant aspects from previous work and deliverables in Chapter 2. Chapter 3 describes the architecture of the SAFETY4RAILS Information System (S4RIS) including aspects that were not detailed by previous work or deliverables. Chapter 4 describes in detail the interoperability of the S4RIS and how the tools will communicate with each other. Chapter 5 deals with the security concept of S4RIS and Chapter 6 concludes this deliverable.

2. Prerequisites: Relevant Aspects from Previous Work

In this chapter, the most relevant aspects from previous deliverables in SAFETY4RAILS with a focus on the platform architecture and security aspects are presented: The requirements most specifically relevant for this deliverable and the first description of the SAFETY4RAILS platform architecture based on D1.4 and D2.3. These requirements form the basis for the development of the interoperability architecture of the S4RIS and are mentioned here for the sake of completeness to make sure that all relevant aspects were considered.

2.1 Requirements from D1.4 most relevant for this present deliverable

The short names and IDs of all relevant requirements to such an information system from the D1.4 are listed in Table 1, except those listed separately for standards and security in section 2.3. Since D1.4 is confidential, only the main topics are mentioned here to illustrate the main aspects taken into account. (However, the full requirments and connected specifciations in D1.4 must be taken into consideration during the development.)

TABLE 1: SUMMARY OF THE RELEVANT REQUIREMENTS

Requirement ID	**P-01**
Short name	Platform modularity
Deliverable	D1.4

Requirement ID	**P-03**
Short name	End User configuration
Deliverable	D1.4

Requirement ID	**P-04**
Short name	Minimum requirements for S4RIS use
Deliverable	D1.4

Requirement ID	**P-06**
Short name	Data exchange – end-user sources to S4RIS
Deliverable	D1.4

Requirement ID	**P-08**
Short name	Data exchange – Between S4RIS tools
Deliverable	D1.4

Requirement ID	**P-09**
Short name	Synchronisation
Deliverable	D1.4

Requirement ID	**P-12**
Short name	Archive
Deliverable	D1.4

Requirement ID	**P-17**
Short name	Security
Deliverable	D1.4

Requirement ID	**P-18**
Short name	Public accessibility
Deliverable	D1.4

Requirement ID	**P-20**
Short name	Internal communication over a distributed messaging system
Deliverable	D1.4

Requirement ID	**P-21 (IO-3 in D2.3)**
Short name	Data exchange with the end-users' system
Deliverable	D1.4

Requirement ID	**P-22 (IO-3 in D2.3)**
Short name	Data exchange – Upload already existing data in the S4RIS
Deliverable	D1.4

Requirement ID	**P-23 (IO-3 in D2.3)**
Short name	The S4RIS shall provide a possibility to connect to not specified systems
Deliverable	D1.4

Requirement ID	**GUI-R01**
Short name	Web-based interface
Deliverable	D1.4

Requirement ID	**GUI-R02**
Short name	Login page
Deliverable	D1.4

Requirement ID	**GUI-R05**
Short name	How to launch tools
Deliverable	D1.4

Requirement ID	**GUI-R14**
Short name	Opening web-based tools
Deliverable	D1.4

Requirement ID	**GUI-R14**
Short name	Opening web-based tools
Deliverable	D1.4

Requirement ID	**GUI-R15**
Short name	Opening desktop tools
Deliverable	D1.4

Requirement ID	**GUI-R16**
Short name	Opening CLI tools
Deliverable	D1.4

Requirement ID	***OSINT_3***
Short name	Storage and representation
Deliverable	D1.4

Requirement ID	***OSINT_5***
Short name	Data access and messaging
Deliverable	D1.4

Requirement ID	***Blockchain_02***
Short name	Data ingestion
Deliverable	D1.4

Requirement ID	***Blockchain_04***
Short name	Data access
Deliverable	D1.4

2.2 Basic Concept of the S4RIS Architecture Based on D 2.3[5]

According to Chapter 3.2 in D2.3, the SAFETY4RAILS platform is envisaged to have a layered architecture with different tools capable of performing different tasks. The layers are modelled according to a semantic grouping of the tools and their interactions. Hence, the tools can simultaneously be present in different layers. The architecture consists of the following five layers:

- Source Layer
- Information Exchange Layer
- Storage Layer
- Data Processing Layer
- Decision Support Layer

Figure 1 shows the layered concept architecture where arrows represent information flow between the tools or layers.

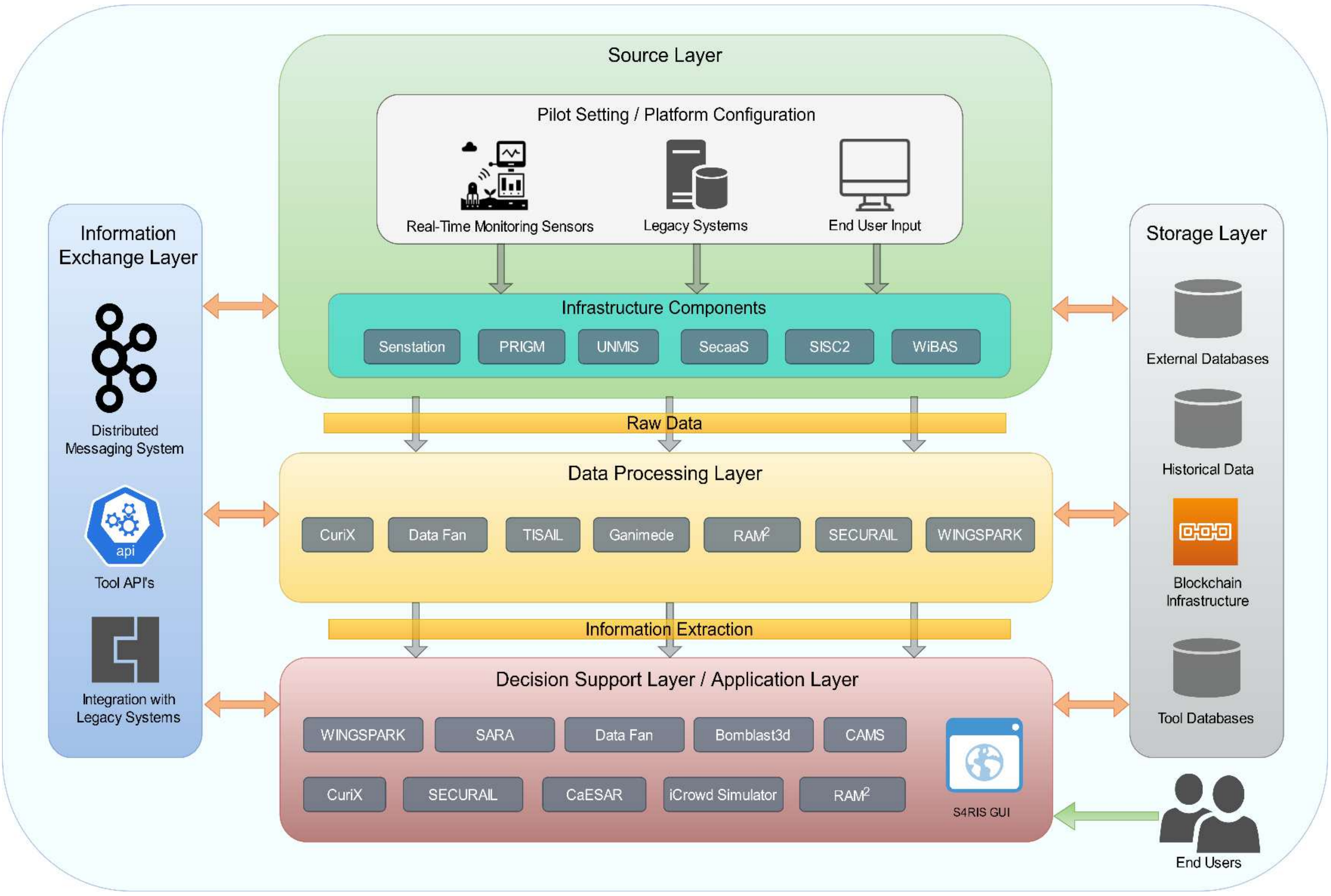

FIGURE 1: S4RIS PLATFORM CONCEPT ARCHITECTURE ACCORDING TO D2.3.SOURCE: D2.3, CHAPTER 3.2, PAGE 39

[5] Panou, K., Argyriou, L. (2021): D 2.3 – System specifications and concept architecture. SAFETY4RAILS internal document.

2.3 Standards and Security

In this section, the short names and IDs of the most relevant requirements from the perspective of standards and security are listed together with the source deliverable where to find them (Table 2). These requirements for standards and security are listed separately since there is a focus on the security aspects in D6.1 and this is described in detail in Chapter 5 – the security concept of the S4RIS platform. (However, the full requirements and connected specifications in D1.4 must be taken into consideration during the development e.g. STD-R02 multifactor for remote connection and STD-R04 Non-human user identification and authentication.)

TABLE 2: SUMMARY OF THE STANDARDS AND SECURITY ASPECTS

Requirement ID	**STD-R01**
Short name	Human user identification and authentication
Deliverable	D1.4

Requirement ID	**STD-R03**
Short name	Human user identification and authentication - multifactor
Deliverable	D1.4

Requirement ID	**STD-R05**
Short name	Account management
Deliverable	D1.4

Requirement ID	**STD-R07**
Short name	Secure log-on
Deliverable	D1.4

Requirement ID	**STD-R16**
Short name	Password management
Deliverable	D1.4

Requirement ID	**STD-R40**
Short name	Protection of communications
Deliverable	D1.4

Requirement ID	**STD-R42**
Short name	Information backup
Deliverable	D1.4

Requirement ID	**STD-R45**
Short name	Source code protection
Deliverable	D1.4

Requirement ID	**STD-R47**
Short name	Integration of a security incident tracking system form
Deliverable	D1.4

Requirement ID	**STD-R48**
Short name	Overall security event / incident / vulnerability database
Deliverable	D1.4

As identified in D1.4, requirements described above have been derived from the following standards (Table 3).

TABLE 3: STANDARDS FROM WHICH REQUIREMENTS DERIVED IN S4RIS DEVELOPMENT FOR SECURITY

Standard Code	Name of the Standard	Related Requirements
ISO 27001[6]	Information technology — Security techniques — Information security management systems — Requirements	STD-R01, STD-R03, STD-R05, STD-R07, STD-R16, STD-R40, STD-R42, STD-R42
ISO 27002[7]	Information technology — Security techniques — Code of practice for information security controls	STD-R01, STD-R05, STD-R07, STD-R16, STD-R40, STD-R42, STD-R42
IEC 62443-3-3[8]	Industrial communication networks - Network and system security - Part 3-3: System security requirements and security levels	STD-R01, STD-R03, STD-R05, STD-R07, STD-R40, STD-R42
ISO 22301[9]	Security and resilience – Business continuity management systems – Requirements	STD-R42
ISO 27035[10]	Information security incident management	STD-R42, STD-R42

[6] https://www.iso.org/standard/54534.html [11.03.2022]

[7] https://www.iso.org/standard/54533.html [11.03.2022]

[8] https://webstore.iec.ch/publication/7033 [11.03.2022]

[9] https://www.iso.org/standard/75106.html [11.03.2022]

[10] https://www.iso.org/standard/74033.html [11.03.2022]

2.4 Top-Level Structure of S4RIS according to previous deliverables

In this section, the structure of the layers is described and the changes in comparison to D2.3 are highlighted.

2.4.1 Tools and main components of S4RIS

According to D2.3, the 19 tools[11] involved in SAFETY4RAILs are thematically structured as in Figure 2.

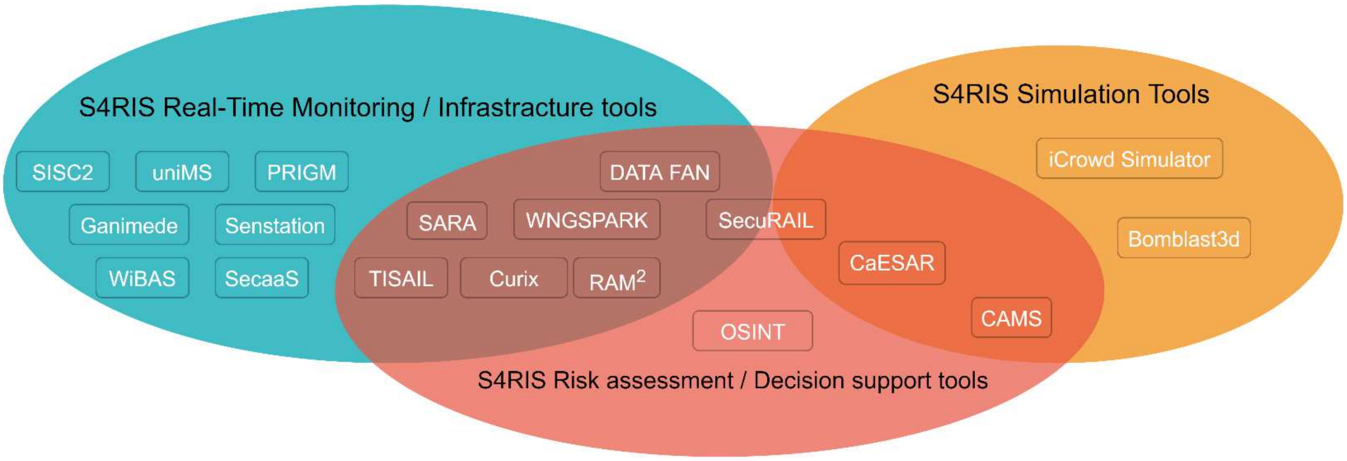

FIGURE 2: DOMAIN INTERSECTION OF TOOLS IN THE S4RIS PLATFORM (SEE D 2.3).

According to D2.3, at a top-level, tools in the S4RIS platform are foreseen to be able to provide functionality under three main domains:

- Real-Time Monitoring / Infrastructure tools
- Simulation tools
- Risk assessment / Decision support tools

As mentioned in D2.3, there are two main enabling components of the S4RIS platform:

- Apache Kafka as the Distributing Messaging System (DMS) provided by NCSRD for the communication between the tools
- The Graphical User Interface (GUI) provided by UNEW

A third component is for the storage of the data identified which is also described in D2.3 and D1.4. The data source layer will be presented in Chapter 3 as one of the main components further described in this deliverable.

For the sake of completeness, these two pre-defined components are summarized in Table 4 and Table 5. For more details, please refer to D2.3[12].

[11] Presently 18 tools are expected.

[12] Panou, K., Argyriou, L. (2021): D 2.3 – System specifications and concept architecture. SAFETY4RAILS internal document.

Table 4: Brief summary of Apache Kafka according to D2.3

Component Name	**Apache Kafka as DMS**
Responsible	NCSRD
Pre-condition / Input	
JSON REST API	
Post-condition / Output	
JSON REST API	
Operational Specification	
Operation/Function Description	**Data Needs**
Provide a secure channel for communication between S4RIS platform tools.	N/A
Provide authentication capabilities to the platform for ensuring that only allowed parties are allowed to use DMS.	N/A
Allow the exchange of messages in JSON format between parties in DMS.	Specification of the Data Model for each topic in the DMS.
Component Dependencies	
All components in S4RIS that will integrate through DMS	

Table 5: Brief summary of SAFETY4RAILS Information System's Graphical User Interface (S4RIS GUI)

Component Name	**S4RIS GUI**
Responsible	UNEW
Pre-condition / Input	
Actions from end-users in the GUI.	
Post-condition / Output	
Depending on user actions this may vary. Results may be displayed or other tools may be opened.	
Operational Specification	
Operation/Function Description	**Data Needs**
Authentication and Authorization capabilities for users.	N/A
Providing links to access other web-based, desktop and CLI S4RIS platform tools.	Information required from each tool that needs to be accessed through S4RIS GUI.
Account management	Specification of the Data Model for each topic in the DMS.
Settings and configuration	N/A
Choice of Interface Languages	Relevant translations for interface elements.
Help and Documentation	N/A
Component Dependencies	
All components in S4RIS will be available for access through S4RIS GUI.	

3. Architecture of S4RIS

In this chapter, the elements for the architecture of the SAFETY4RAILS Information System S4RIS, which are particularly relevant for this deliverable, are considered in more detail, as follows.

3.1 Architecture and Refined Layers

For the envisioned and final product of S4RIS (after the project), the following top-level structure for the integration of the tools will be applied as shown in Figure 3. Furthermore, the different layers result from the previous defined architecture as described in Chapter 2 as well as specific requirements to insert the relevant single tools properly (e.g. PRIGM by ERARGE).

- **Decision Support Layer** includes the graphical user interface (GUI) for the S4RIS provided by UNEW as described in Chapter 2.4.1. This GUI will offer various options to cover the requirements of the various tools that were already described in more detail in D1.4 and D2.3:
  - Tools with a web GUI like CURIX,
  - Tools as a desktop application like the DATA FAN or
  - CLI-Tools like CaESAR.
- **Data Processing Layer** includes the various tools dealing with data.
- **Information Exchange Layer** provides the encryption and authentification via Apache Kafka provided by NCSRD as described in Chapter 2.4.1 to ensure the data exchange of the various tools with the S4RIS platform.
- **Data Source Layer** includes all data providers regardless of whether the data was streamed from the sensor stations or was persisted in databases. Therefore, this layer contains the databases of each tool and the decryption tool PRIGM provided by ERARGE. For further reading see Chapter 3.2. Comparing the architecture represented in D2.3 and D1.4, this representation of the data source layer combines the representation of the separate Source and Storage Layers (which remain valid depending on the tools and data involved and as presented above and in D2.3 and D1.4).
- **Network Infrastructure Layer** includes the tools for setting up a safe network for each tool provider resulting in a secure S4RIS such as the firewall, the intrusion detection system and a (reverse) proxy.
- **Central Network Infrastructure Layer (UNEW primarily responsible):** In comparison to the Network Infrastructure Layer this Central Network Infrastructure Layer provides tools that will cover tasks concerning the security for each tool of the S4RIS platform. Therefore, these tasks do not have to be considered by the tool providers individually. For example, the central layer contains the authentication server which provides the authentication between the end-users and the web applications.

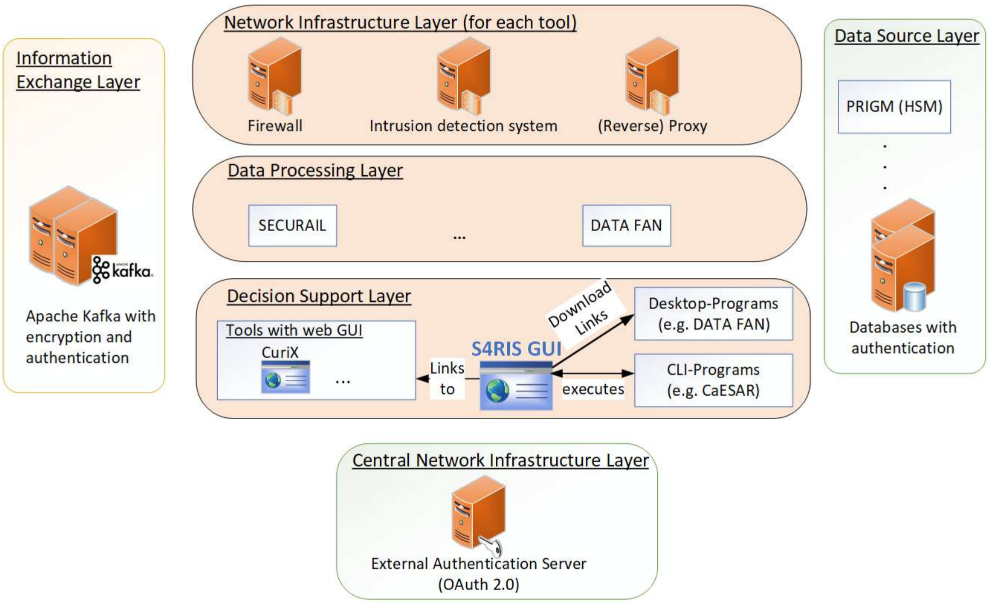
Information Exchange Layer
kafka
Apache Kafka with encryption and authentication
Network Infrastructure Layer (for each tool)
Firewall
Intrusion detection system
(Reverse) Proxy
Data Processing Layer
SECURAIL
...
DATA FAN
Decision Support Layer
Tools with web GUI
CuriX
...
Links to
S4RIS GUI
Download Links
executes
Desktop-Programs (e.g. DATA FAN)
CLI-Programs (e.g. CaESAR)
Data Source Layer
PRIGM (HSM)
Databases with authentication
Central Network Infrastructure Layer
External Authentication Server (OAuth 2.0)

FIGURE 3: TOP-LEVEL STRUCTURE OF THE VARIOUS LAYERS IN S4RIS

For the simulation exercises during the SAFETY4RAILS project, all tools including Apache Kafka can be accessed from the internet. Furthermore, each tool is secured by an individual appropriate infrastructure that is included in the Network Infrastructure Layer. These security aspects are described in more detail in Section 5 and will be evaluated during the exercises and demonstrations within the project SAFETY4RAILS.

3.2 Data Source Layer

As shown in Figure 3, the data source layer will be designed to fulfill the requirements stated in D1.4, e.g. P-06, P-08 as mentioned in Section 2.1, and to control the data according to deliverable D1.6[13]. It also includes tools such as PRIGM offered by ERARGE providing an encryption solution for long-term data. PRIGM does not offer any database except a log registry that can be used by third-party applications (e.g. network intrusion detection tools, etc). PRIGM also presents the possibility of crypto services (API, in the form of Crypto-as-a-service i.e. when not installed as a hardware module on the S4RIS platform) that can be used by other services. According to D2.3, PRIGM is described in Table 6.

TABLE 6: SHORTENED DESCRIPTION OF PRIGM ACCORDING TO D 2.3

Component Name	**PRIGM**
Responsible	ERARGE
Description and Objectives	
PRIGM is a Hardware Security Module (HSM). It is a device capable of doing major cryptography operations such as true random number generation, prime number generation, key generation and management, secure key storage and exchange, symmetric encryption (AES, 3DES), asymmetric encryption (RSA, ECDSA), and hashing (SHA). HSM connects to a host device (server, PC, etc.) using PCIe interface. It is enclosed in a tamper-proof enclosure. PRIGM operates on the server-side.	
Pre-condition / Input	
Sensory or any type of data comes from field to the S4RIS (e.g. Operation Control Centre (OCC)) through IoT channels or service APIs. If the acquired data is sensory data and Senstation (a secure IoT gateway) is set up at end nodes, Senstation and PRIGM work coherently to encrypt data. Any other data can be encrypted (or decrypted before it is used) by using PRIGM only.	
Post-condition / Output	
If the encrypted data is received by PRIGM the output is the decrypted data in its original format. (e.g., json, csv, xml, SQL). If PRIGM was asked to encrypt open data (supposed to be input from a service), the output will be the encrypted version of the open (received) data. Output can be streaming data or output data can be stored in a database	

[13] Matsika, E. (2021): D 1.6 - Data control and management plan. SAFETY4RAILS internal document.

Operational Specification		
S4RIS Domain	**Operation/Function Description**	**Data Needs**
Monitoring	PRIGM gets sensory data from the field and decrypts them. PRIGM is connected to a server via PCIe connector. Sensory data can be stored in the server or sent through the network.	N/A

Component Dependencies	
Component Name	**Needs and data description**
Senstation	PRIGM needs encrypted sensory data that comes from the field. Senstation listens to the sensor connected to Senstation, encrypts the data and sends them to the central server.
Information Exchange Layer	Any published data from the Information Exchange Layer can be securely encapsulated by PRIGM (optional), i.e. if such data is planned to be kept for a long-term.

Additional to PRIGM, the storage layer contains databases to store long-term data as well. Most tools in S4RIS do have their own internal database (SQL or NoSQL) since they are already designed as stand-alone tools. Therefore, these databases will be implemented during the project and tested and validated during the project exercises and demonstrations.

The D1.4 describes also a central database for e.g. the recognition and persistence of messages transported over Kafka and also for the sending of persisted data over Kafka in the same sequence as they arrived to simulate real-time behaviour of pre-recorded data. Minimum functionalities of the central database with regards to Kafka messages are delivered by the database configured with Kafka which works as a temporary store for messages. In principle tools with relevant databases and functionalities could also replay Kafka messages. The relevance of a central database particularly for a future product with an end-user remains valid.

The database for S4RIS user account data for example will be held in the database(s) provided by UNEW and MdM. The communication of the storage layer is described in Chapter 4 in more detail.

4. Interoperability

To provide a flexible framework for the interoperability within S4RIS a Distributed Messaging System (DMS) will be used (implementing a REST API "wrapper"). This will also enable the secure integration of external tools. The DMS chosen is based on Apache Kafka technology, which is a widely used and trusted system. Other bilateral interfaces (e.g. also based on REST API) can be used if necessary, but only on a case by case basis and with an agreement, especially in the project. To ensure coherent data handling, all tools are required to be able to read in and provide their output in JSON format.

4.1 Integration of Tools in S4RIS

The integration of a tool in S4RIS comprises two different aspects. First, a tool can be integrated from the end-user-perspective by integrating it within the S4RIS GUI. The second aspect is to integrate a tool by connecting it to the DMS, so that data can be exchanged between tools.

During the project the S4RIS tools are hosted by the respective tool providers for the S4RIS prototype and are connected to the platform via Apache Kafka. Each partner is responsible to provide state-of-the-art security for their tools with access to the DMS. In a product version, all tools are expected to be hosted with the end-user remains valid (unless a "virtual" product version is implemented, should this prove to be a marketable solution).

The GUI of S4RIS shows all the tools and provides the users with the ability to select the ones they want (except for PRIGM, see below). Following the D1.4 specification GUI-05 (page 28):

"For each tool, there should be a clickable link that opens the tool in one of the following possibilities:

- *Navigate to the web GUI of the selected tool (within the S4RIS GUI or open another tab/browser window)*
- *Open the respective tool on the client machine via an executable file*
- *Open a web page that provides access to a remote machine in which the tool can be executed"*

It is identified that a user may not need to access a specific tool's GUI if that tool's input/output is sole via the DMS from/to other tools.

It is planned for the final product, that in case several tools are selected (expected as the standard) the user can insert data in the main GUI that can be used by all selected tools.

Most of the tools can be integrated as previously described. But for integrating PRIGM and Senstation, another approach is needed: In a potential product version of S4RIS, a physical installation of a hardware security module (PRIGM) could be an option for customers. PRIGM does not itself have a user interface as mentioned above. In the project, the hardware installation will be realised only for EGO and TCDD use cases for the sake of proof of concept. The physical integration will be done by installing PRIGM on an experimental server in close coordination with EGO via its PCIe interface. Senstation will also be physically installed to collect IoT data from various sensors such as wind sensors, temperature, humidity, vibration, etc. The collected data is planned to be published to other partners so that they will have the chance to improve their algorithms with realistic data.

In the S4RIS prototype a software replacement for PRIGM (in the form of Crypto-as-a-Service API) has been offered for the demonstrations (e.g. Ankara Simulation Exercise as described in D8.2[14]) by ERARGE with tool provider(s) for exemplary long-term data. This decision was taken for the sake of simplicity and practical reasons as the integration of software-based Crypto-as-a-Service that is built on the hardware-based HSM (PRIGM) is easier to implement and demonstrate. As agreed, this approach will be presented as proof of the use of a hardware-based HSM in an information management system such as S4RIS although the mainstream S4RIS will rely on the TLS (Transport Layer Security). See Figure 4 for the use of PRIGM for short and long-term data security needs.

[14] Villamor, E. (2022) : D 8.2 – First Version – Development of a blueprint exercise handbook. SAFETY4RAILS internal document.

4.2 Data Sources

According to chapter 3.4.1 of D1.4 there are three main data sources for S4RIS:

- **Manual input via GUI**:
  The user manually inserts the data e.g. via forms
- **Uploading data**:
  The end-user uploads data via the S4RIS
- **Streaming real-time data**:
  Relevant data from sensors can be streamed into S4RIS

In addition to these three data sources, there are tool individual data sources, such as databases for example, which were previously mentioned in Chapter 3.

A schedule for integrating the data sources can also be found in Chapter 3.4.1 of D1.4. According to this schedule, a technical realization of the first two options, the manual input via the GUI and uploading data, will be already implemented during the project duration and also be part of the testing and validation phase during the project's exercises and demonstrations. Only the third option, including streaming of real-time data is most likely not to be demonstrated in the project. Instead, data mimicking real-time data will be provided to the necessary tools by preparing files and transferring them to the tools.

4.3 Communication of Tools with the Database of S4RIS

Apache Kafka is used for exchanging messages between tools and to store data temporarily. For persisting and storing data in the long term, databases are needed. As mentioned in chapter 3.2, most of the tools have their own databases. Subject to what is written in D1.4 and chapter 3.2 above, it is up to individual tool providers to decide which data is relevant to store in the project in agreement with the other beneficiaries. Relevant data for persisting for a later use, could be for example alerts or data processing steps, which were previously sent via Apache Kafka. For sharing and publishing achieved data the corresponding tool loads the data from its database, process and publishes it via Apache Kafka. Hence, one tool cannot access a database of another tool directly. They must communicate typically via Apache Kafka. An exception could be when sharing larger files. For larger files, a different solution to Apache Kafka must be chosen e.g. providing a link to the respective location such as network-attached storage (NAS).

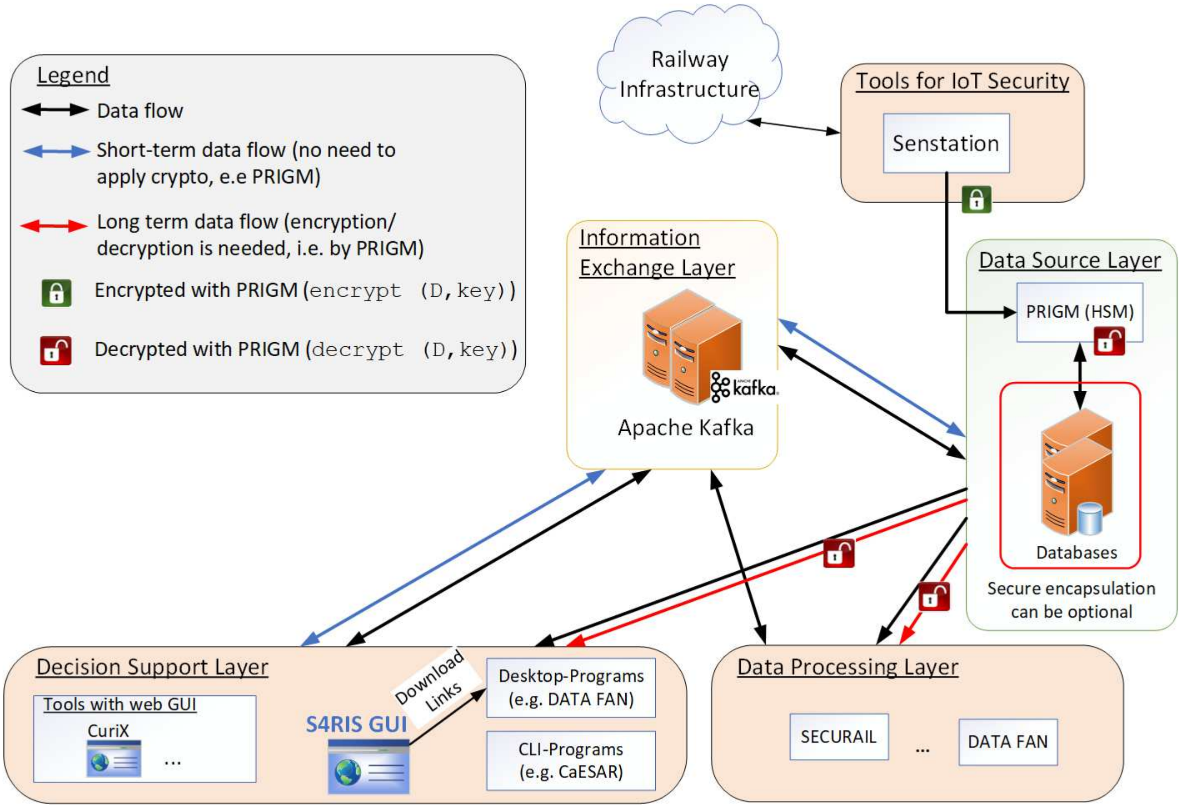
Legend
Data flow
Short-term data flow (no need to apply crypto, e.e PRIGM)
Long term data flow (encryption/ decryption is needed, i.e. by PRIGM)
Encrypted with PRIGM (encrypt (D,key))
Decrypted with PRIGM (decrypt (D,key))
Railway Infrastructure
Tools for IoT Security
Senstation
Information Exchange Layer
kafka
Apache Kafka
Data Source Layer
PRIGM (HSM)
Databases
Secure encapsulation can be optional
Decision Support Layer
Tools with web GUI
CuriX
...
S4RIS GUI
Download Links
Desktop-Programs (e.g. DATA FAN)
CLI-Programs (e.g. CaESAR)
Data Processing Layer
SECURAIL
...
DATA FAN

FIGURE 4: DATA FLOW WITHIN THE S4RIS.

As depicted in Figure 4, using PRIGM for short-term or instant data flows may not be feasible and practical although it presents high throughput and fast cryptographic operations. The blue arrows illustrate such frequent data flows via the Apache Kafka broker where TLS can be seen as a sufficient security countermeasure (see Figure 5).

PRIGM, on the other hand, can be effectively used for secure encapsulation (if needed or can be optional) if the data should be kept for a longer-term. For instance, any data can be kept secure in the database by using the PRIGM's encrypt function (`encrypt(D, key)`). The encrypted data can be decrypted whenever it is requested by using the `decrypt(E, key)` function.

For high-frequency data, e.g. sensory data acquired from end nodes, PRIGM and Senstation jointly used are an option to set up a secure transmission channel. Senstation and PRIGM use the same set of true random number pool (used for private key generation) so that these two devices establish a secure channel protected by unique private keys. Based on such a symmetric cryptography protocol, Senstation encrypts the data that is collected from the field and sends the encrypted data to the Storage Layer where PRIGM is installed. PRIGM can decrypt the data and the database service (e.g. DBMS) can store the sensory data for further use. As described above, this IoT-based approach will be implemented for the needs of end-user partners, TCDD and EGO particularly in this case.

4.4 Communication Between Tools

For exchanging messages between tools Apache Kafka will be used, unless agreed otherwise on a case by case basis. In Apache Kafka the tools push their data to specific topics, the topics are used for direct communication between tools as relevant (depending on the topic). A wrapper was put around Kafka to allow a simple REST connection. Apache Kafka is described in more detail in D6.2[15].

Apache Kafka is the IT infrastructure that enables tools to communicate with each other. Which data is exchanged actually and what topics exist needs to be identified and implemented between tool providers, considering which data is useful for exchanging. The explicit topics and exchanged data will be elaborated and described in other tasks and deliverables. Exemplary for the Metro de Madrid use case the monitoring tool providers committed in task 4.5 on a topic for sending alerts to RAM². The data flow for sending alerts for the Metro de Madrid use case is shown in Figure 5.

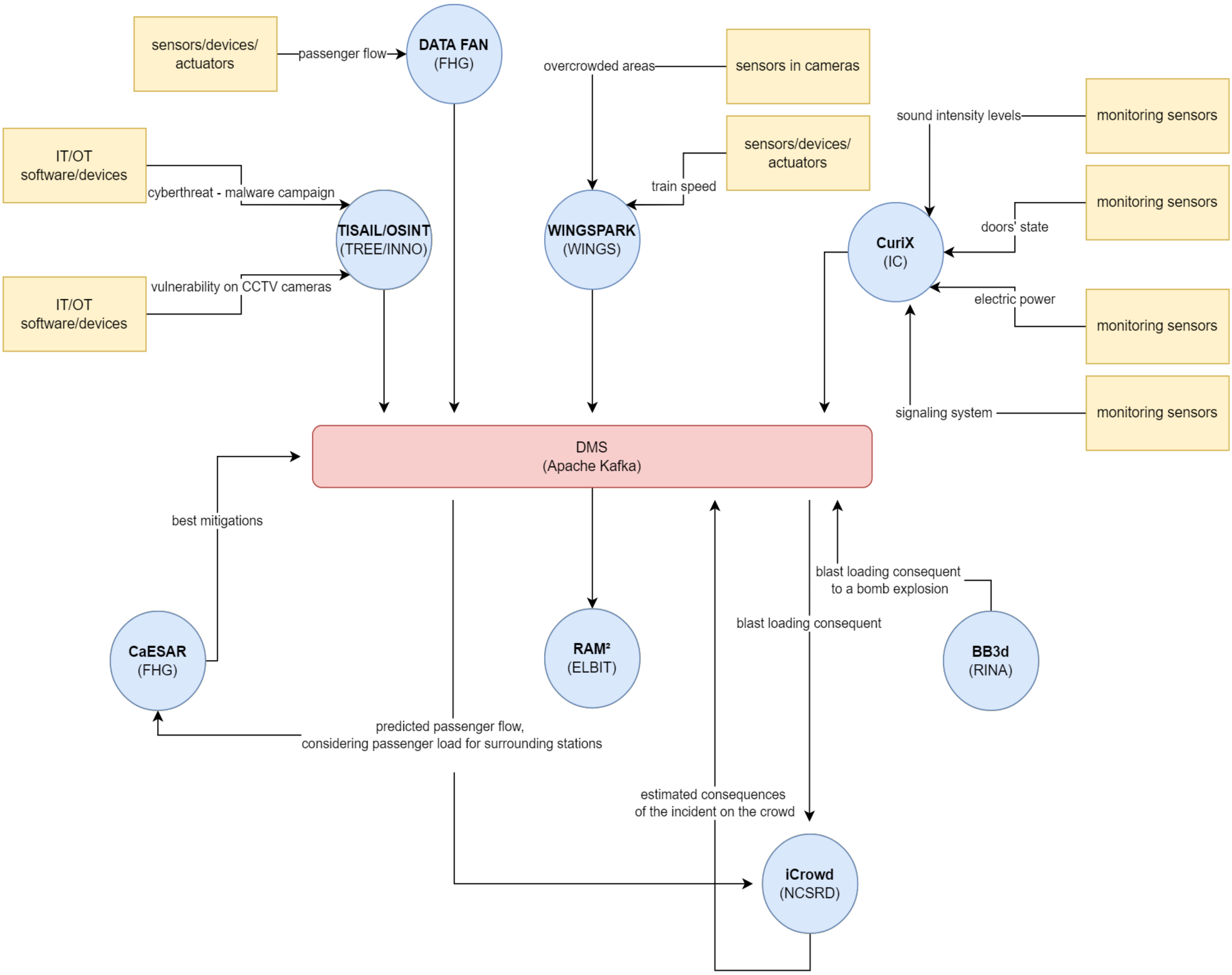

FIGURE 5: EXEMPLARY DATA-FLOW DIAGRAMM. SOURCE: D 6.2 CHAPTER 3.3 (VERSION 0.6)

For further details about this data flow see D6.2 and for further details regarding the use case see D4.6[16]. Additionally, it is already determined that SecuRail 2.0 needs input from CuriX.

[15] Zacharakis, D. / Panou, K. (2022) : D 6.2 – Implementation of technical interoperability and interfaces specification. SAFETY4RAILS internal document. In preparation.

[16] Siebold U. (2022) : D 4.6 – Implementation of real-time monitoring components to S4RIS. SAFETY4RAILS internal document. In preparation.

Concerning the communication between end-user tools and data sources, particularly in a future product version, D1.4[17] states under the specifications P-06 (page 18):

"*For the productive version, the real-time monitoring tools will be connected via appropriate / proprietary interfaces to data sources (e.g. CuriX will collect relevant data from classical IT-monitoring tools like elastic or PRTG which provide also means to collect SCADA data) in an end-user specific way, i.e. customized connections, because each possible end-user will provide data in a different way."*

Additionally, D1.4[11] states under P-07 (page 19):

"In the productive version after the project, with respect to the individual circumstances at end-users' infrastructure, data exchange to end-user systems can be addressed over direct communications to APIs or by providing access to S4RIS DMS."

[17] Siebold, U, Crabbe, S. (2021): D1.4 – Specification of the overall technical architecture. SAFETY4RAILS internal document.

5. Security Concept

The D1.4[18] contains many requirements and specifications connected with security. These are primarily informed by relevant EU legislation and standards. However, it is also stated in D1.4 in Section 2.2.4 “Standards” (page 37):

“According to the fact that this project has a limited number of resources and time, not all of the identified requirements will be achieved during this project. But even when not all requirements can be achieved within the project, no development should be carried out in SAFEYT4RAILS which would negate the possibility or even make it extremely hard to fulfil one or more of the requirements determined as essential for the S4RIS product(s). This especially holds true for requirements in this section since those assure the secure operation of S4RIS itself.”

In this chapter, the security concept of the S4RIS will be discussed. Its focus is on what is targeted for implementation within the project. The security concept is presented in a top-down fashion i.e. first high-level aspects e.g. how to secure the S4RIS from the regular internet and in the end low-level aspects e.g. security of the tools. The security concept is summarized in Figure 6.

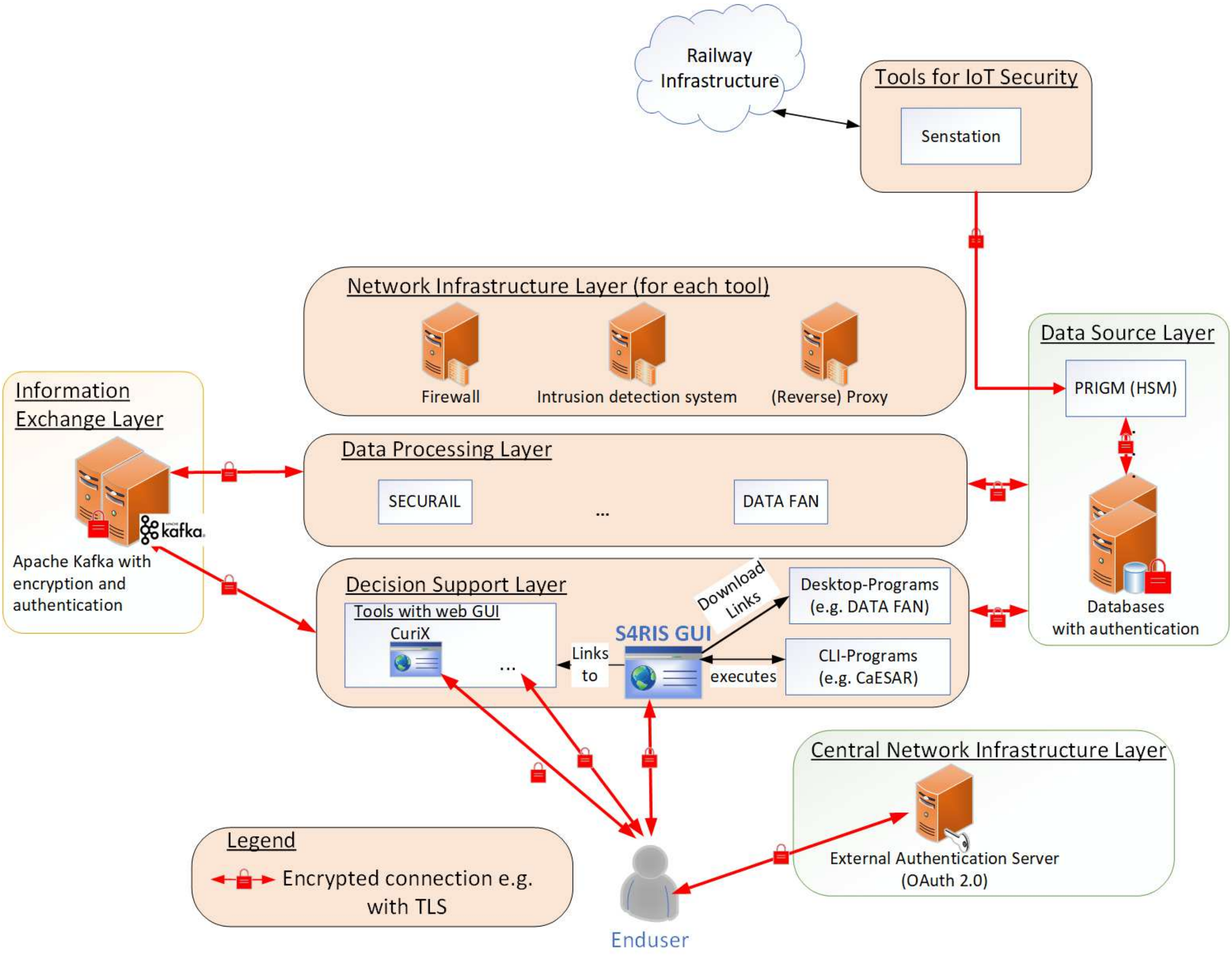
Railway Infrastructure
Tools for IoT Security
Senstation
Network Infrastructure Layer (for each tool)
Firewall
Intrusion detection system
(Reverse) Proxy
Data Source Layer
PRIGM (HSM)
Databases with authentication
Information Exchange Layer
kafka
Apache Kafka with encryption and authentication
Data Processing Layer
SECURAIL
...
DATA FAN
Decision Support Layer
Tools with web GUI
CuriX
...
Links to
S4RIS GUI
Download Links
executes
Desktop-Programs (e.g. DATA FAN)
CLI-Programs (e.g. CaESAR)
Central Network Infrastructure Layer
External Authentication Server (OAuth 2.0)
Legend
Encrypted connection e.g. with TLS
Enduser

FIGURE 6: OVERVIEW OF THE SECURITY CONCEPT OF S4RIS.

[18] Siebold, U, Crabbe, S. (2021): D1.4 – Specification of the overall technical architecture. SAFETY4RAILS internal document.

5.1 Network infrastructure

During the project the components and tools are hosted on the servers and data centres of the corresponding tool providers (as noted above, in a product version all tools are expected to be hosted with the end-user remains valid unless a “virtual” product version is implemented, should this prove to be a marketable solution). For communication between tools via Apache Kafka as well as for providing web application to the end-user, a connection to the internet is necessary. Therefore, each tool provider must consider security aspects regarding their IT infrastructure individually. This includes, for example, isolating the hosting servers from their internal network e.g. with firewalls to create a demilitarized zone (DMZ) as shown for example in Figure 7.

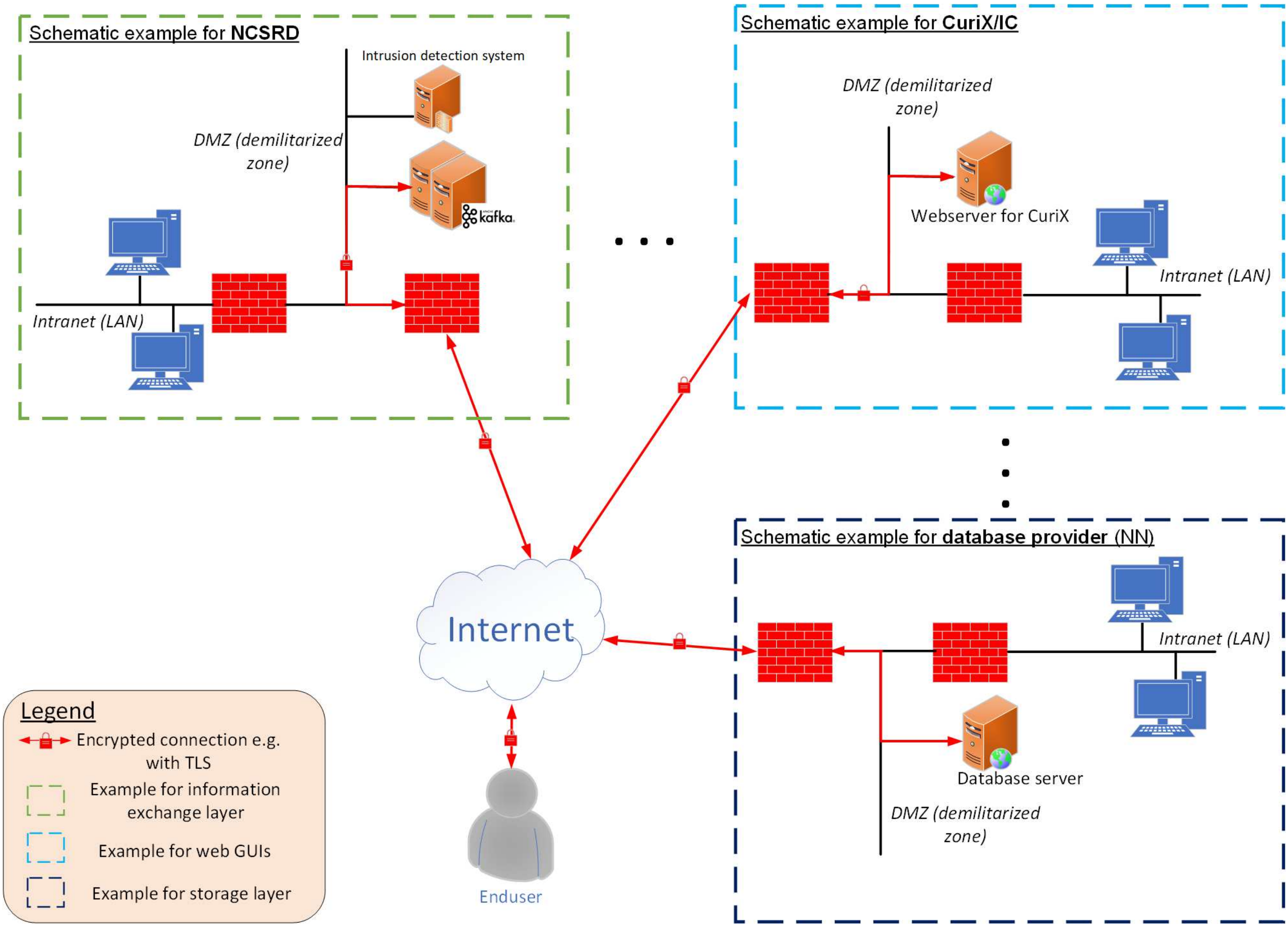
Schematic example for NCSRD
Intrusion detection system
DMZ (demilitarized zone)
kafka
Intranet (LAN)
Schematic example for CuriX/IC
DMZ (demilitarized zone)
Webserver for CuriX
Intranet (LAN)
Schematic example for database provider (NN)
Intranet (LAN)
Database server
DMZ (demilitarized zone)
Internet
Enduser
Legend
Encrypted connection e.g. with TLS
Example for information exchange layer
Example for web GUIs
Example for storage layer

FIGURE 7: SCHEMATIC NETWORK PLAN

The need for a secure IT infrastructure can differ from tool to tool, especially tools with open interfaces and ports to the internet that must be secured by an appropriate IT infrastructure. This is the case for Apache Kafka and all tools which are providing a web GUI. Tools with no open ports and which are only communicating with Apache Kafka do not need a complex IT infrastructure.

In the product version, when everything is on the end-user infrastructure, this secure IT infrastructure is not needed any more.

5.2 Encryption and authentication for external Communication

S4RIS has two different external communication points, which must be addressed separately. First, the communication with end-users and second the communication with the railway infrastructure.

5.2.1 Communication with End-users

As mentioned before each tool provider secures its web applications with appropriate firewalls and security gateways. All connections between the end-user and to the corresponding web application must be encrypted e.g. with the use of HTTPS and TLS.

As noted, above the D1.4 provides a whole set of requirements and connected specifications. As a minimum for the S4RIS prototype, before getting access to the S4RIS GUI and the web applications, the user must authenticate with a username and a strong password in line with the requirement STD-R01 from D1.4. When choosing a password, the user must consider the following ISO 27001, ISO 27002 standards:

- Minimum length of 8 characters
- At least one uppercase letter
- At least one digit or a special character is required

Additionally, two-factor authentication is necessary. For implementing two-factor authentication, the service Authy[19] can be used. On the one hand, Authy offers end-users a smartphone or desktop app for known websites e.g. for Amazon. On the other hand, Authy also provides the necessary tools[20] for website developers to integrate two-factor authentication e.g. with QRCodes or with SMS.

To provide authentication, an additional OAuth 2.0 authentication server is necessary. It is foreseen to apply OAuth 2.0 unless problems are detected with the software connections. In this case, alternatively, the following options Google Authenticator[21], Duo Security[22] or onelogin[23] can be used. For further detailed information concerning the implemention of the SAFETY4RAILS Information System (S4RIS), the reader is referred to the SAFETY4RAILS Deliverable D 6.3[24].

5.2.2 Communication with Railway Infrastructure

The IoT data of the railway infrastructure can be encrypted by Senstation, which is installed at the client-side (outside the S4RIS platform). The encrypted data is then sent via the Internet to the Storage Layer where PRIGM is installed. PRIGM can decrypt the data and is installed at the server side (inside the S4RIS platform). Normally, Senstation and PRIGM are hardware-based devices enabling encryption and decryption of IoT data. But as mentioned in chapter 4.1 this is currently not possible. Therefore, instead of hardware, a software solution from ERARGE is planned to be applied for encrypting and decrypting the IoT data. The central system in this context is the storage layer which is required to be installed on a server (see Figure 4) and the corresponding technical descriptions for Senstation and PRIGM are used.

For none IOT data e.g. network data, the solution will depend on the tools and data to be connected also respecting what the user already has (i.e. legacy systems), see also section 4.4.

5.3 Encryption for intra-communication between S4RIS tools

Depending on the communication method and channel the security aspects must be considered differently.

5.3.1 Security Aspects for Apache Kafka

All connections between the Kafka cluster and the producers and the consumers will be encrypted with TLS. All technical providers have dedicated accounts for authenticating with the platform. In general, all tools can access all Apache Kafka topics. At the user level access to specifics may be restricted through the definition of user roles. Details will be further outlined as part of D6.2[25].

[19] https://authy.com/

[20] Link to the API documentation : https://www.twilio.com/docs/authy/api

[21] https://www.google.com/landing/2step/

[22] https://duo.com/

[23] https://www.onelogin.com/

[24] David, R./ Matsika, E. (2022) : D6.3 – Mid-report on validation and evaluation of the SAFETY4RAILS Information System (S4RIS). SAFETY4RAILS internal document. In preparation.

[25] Zacharakis, D. / Panou, K. (2022): D 6.2 – Implementation of technical interoperability and interfaces specification. SAFETY4RAILS internal document. In preparation.

5.3.2 Security Aspects for any REST Interface exception

For any other REST interfaces and REST web services, the communication will be encrypted by using HTTPS and TLS. Furthermore, if a certain tool wants to use a REST web service of another tool it must authenticate itself. For this authentication, an internal authentication server (OAuth 2.0) is needed. It is foreseen to apply OAuth 2.0 unless problems are detected with the software connections. In this case, alternatively, the following options Google Authenticator[26], Duo Security[27] or onelogin[28] can be used.

5.3.3 Security Aspects for Databases

Connections to databases are also being encrypted with TLS for example for Apache Kafka the access privileges to the database are restricted and only the necessary operations are allowed for a tool. For example, for a classical SQL database, only a certain tool is allowed to create or insert data into a table whereas another tool is only allowed to read the data.

5.4 Security of the tools

Securing the connections and the communication between tools and the end-user is not enough if the tool or the server has security vulnerabilities. Therefore, security aspects must be addressed for every single tool individually.

One main aspect is that a single tool is secure by design. That means during the design and implementation phase of a tool security issues are considered from the start. To mention a few best practices: The user input has to be sanitized. SQL injections where end-user can manipulate illegitimately a database by inputting SQL statement into an input field is still very common. Other common attacks such as Cross-Site-Scripting (XSS) should also be considered. The Open Web Application Security Project (OWASP) publishes a list of the top 10 security risks, which should be addressed.

Another aspect is that all dependencies of a tool should be up to date because these dependencies also can have security vulnerabilities. These security breaches are fixed in newer versions and are then often known by the public. Therefore, the known security vulnerabilities can be used by “hacking tools” such as Metasploit[29] for penetration tests to detect vulnerabilities. For checking the vulnerability of dependencies there are several checkers e.g. the build-in mechanism of Node Package Manager (npm) for JavaScript/Typescript packages.

[26] https://www.google.com/landing/2step/ [last update: 25.02.2022]

[27] https://duo.com/ [last update: 25.02.2022]

[28] https://www.onelogin.com/ [last update: 25.02.2022]

[29] https://www.metasploit.com/ [last update: 25.02.2022]

6. Summary and conclusion

This deliverable D6.1 describes a technical concept of the interoperability and the security for the SAFETY4RAILS Information System (S4RIS) with a focus on the prototype implemented in the project. Starting in chapter 2 with a summary of the requirements and the previously designed architecture mentioned in previous deliverables chapter 3 builds on the architecture of the S4RIS platform, containing the following layers:

- Decision Support Layer
- Data Processing Layer
- Information Exchange Layer
- Data Storage / Source Layer(s)
- Network Infrastructure Layer

In comparison to the architecture of the D1.4 and D2.3 a novelty is that especially for the S4RIS prototype, the source and storage layers can be represented as combined in a “Data source layer”. This data source layer contains components to persist data e.g. databases and tools to provide data. This data source layer will be designed to fulfil the relevant requirements stated in D1.4 and for the control of data according to deliverable D 1.6[30]. It also includes tools such as PRIGM offered by ERARGE.

Based on the architecture the interoperability of the tools and the integration in the platform was presented in Chapter 4. The individual tools will communicate with each other via Apache Kafka, any deviations subject to agreement on a case by case basis. They will be loosely integrated within the S4RIS platform, by providing links to the corresponding tools and showing highly processed and merged data on the individual tool GUIs (such as RAM2) as demonstrated via the alerts in the MDM Exercise in February 2022 and/or the central GUI

The presented security concept (see Chapter 5) comprises encryption with TLS for communication of the tools via Apache Kafka and encryption with Senstation and PRIGM for the railway infrastructure as an option. Additionally, each tool provider is responsible to secure their IT infrastructure individually.

[30] Matsika, E. (2021): Data control and management plan. SAFETY4RAILS internal document.

Bibliography

Crismer, F., Macchi, L. (2021) : D 2.4 – Specific requirements for standardization and interoperability. SAFETY4RAILS internal document.

David, R./ Matsika, E. (2022) : D6.3 – Mid-report on validation and evaluation of the SAFETY4RAILS Information System (S4RIS). SAFETY4RAILS internal document. In preparation.

Ferrando, F., De Belloy, F. (2021): D 2.1 – Grid analysis of end-user needs and Workshop minutes. SAFETY4RAILS internal document.

Matsika, E. (2021): D 1.6 - Data control and management plan. SAFETY4RAILS internal document.

Panou, K., Argyriou, L. (2021): D 2.3 – System specifications and concept architecture. SAFETY4RAILS internal document.

Siebold, U, Crabbe, S. (2021): D1.4 – Specification of the overall technical architecture. SAFETY4RAILS internal document.

Siebold U. (2022) : D 4.6 – Implementation of real-time monitoring components to S4RIS. SAFETY4RAILS internal document. In preparation.

Villamor, E. (2022) : D 8.2 – First Version – Development of a blueprint exercise handbook. SAFETY4RAILS internal document.

Zacharakis, D. / Panou, K. (2022) : D 6.2 – Implementation of technical interoperability and interfaces specification. SAFETY4RAILS internal document. In preparation.

Online References

https://authy.com/ [last update: 25.02.2022]

Link to the API documentation : https://www.twilio.com/docs/authy/api [last update: 25.02.2022]

https://www.google.com/landing/2step/ [last update: 25.02.2022]

https://duo.com/ [last update: 25.02.2022]

https://www.onelogin.com/ [last update: 25.02.2022]

https://www.metasploit.com/ [last update: 25.02.2022]

https://www.iso.org/standard/54534.html [11.03.2022]

https://www.iso.org/standard/54533.html [11.03.2022]

https://webstore.iec.ch/publication/7033 [11.03.2022]

https://www.iso.org/standard/75106.html [11.03.2022]

https://www.iso.org/standard/74033.html [11.03.2022]

https://www.iso.org/standard/53467.html [11.03.2022]

ANNEXES

ANNEX I: GLOSSARY AND ACRONYMS

TABLE 7: GLOSSARY AND ACRONYMS

Term	Definition/description
3DES	Triple Data Encryption Algorithm
AES	Advanced Encryption Standard
API	Application Programming Interface
CLI	Command Line Interface
D	Deliverable
DBMS	Database management system
DMS	Distributed Messaging System
DMZ	Demilitarized zone
DoS	Denial of Service
ECDSA	Elliptic Curve Digital Signature Algorithm
GUI	Graphical User Interface
HSM	Hardware Security Module
HTTPS	Hypertext Transfer Protocol Secure
ID	Identification
IT	Information Technology
IoT	Internet of Things
JSON	JavaScript Object Notation
NAS	Network Attached Storage
OCC	Operation Control Center
OT	Operational Technology
OWASP	Open Web Application Security Project
PC	Personal computer
PCI	Peripheral Component Interconnect
REST	Representational State Transfer

Term	Definition/description
RSA	Rivest-Shamir-Adleman
S4RIS	SAFETY4RAILS Information System
SHA	Secure Hash Algorithm
SQL	Structured Query Language
TLS	Transport Layer Security
VPN	Virtual Private Network
XSS	Criss-Site-Scripting

SAFETY4RAILS

Partners:

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 883532.