interconnect

interoperable solutions connecting smart homes, buildings and grids

WP5 – Digital Platforms and Marketplace

D5.4

Prototype for the interoperable marketplace toolbox

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant agreement No 857237

DOCUMENT INFORMATION

DOCUMENT	D5.4 – Prototype for the interoperable marketplace toolbox
TYPE	O
DISTRIBUTION LEVEL	PU
DUE DELIVERY DATE	30/09/2021
DATE OF DELIVERY	30/01/2022
VERSION	V1.1
DELIVERABLE RESPONSIBLE	INESCTEC
AUTHOR (S)	Fábio Coelho (INESC), Milenko Tosic (VLF)
OFFICIAL REVIEWER/s	INESC, VLF, TNO

DOCUMENT HISTORY

VERSION	AUTHORS	DATE	CONTENT AND CHANGES
0.1	Fábio Coelho (INESC)	10/12/2021	Document Structuring
0.1	Milenko Tosic (VLF)	10/12/2021	Document Structuring
0.1	Fábio Coelho (INESC)	05/01/2022	Pre-fill of information in all sections.
0.1	Milenko Tosic (VLF)	25/01/2022	P2P section.
1.0	Fábio Coelho (INESC)	26/01/2022	Executive Summary and final review before QA.
1.0	Milenko Tosic (VLF)	27/01/2022	Final review before QA.
1.1	Fábio Coelho (INESC), Milenko Tosic (VLF)	31/01/2022	Version for submission.

ACKNOWLEDGEMENTS

NAME	PARTNER
Fábio Coelho	INESC
David Rua	INESC
Milenko Tosic	VLF
Sasa Pesic	VLF
Gjalt Loots	TNO

DISCLAIMER:

EXECUTIVE SUMMARY

Energy and IoT domain convergence through cross-domain semantic interoperability is at the forefront of the InterConnect project. The InterConnect project introduces the **Interoperability Framework (IF) as a set of tools and software components that will allow stakeholders to interconnect their semantically interoperable solutions into interoperable ecosystems** which are the basis for developing innovative services, use cases and business models.

This document provides support to D5.4 that overviews the software components developed in WP5 as part of the Interoperability Framework (IF) and finalizes milestone 8.

The development of the IF's components was based on the **specification of key enablers for facilitating semantic interoperability between digital platforms, services, and devices from both the Energy and IoT domains**, namely the Semantic Interoperability Layer, the Service Store, the Generic Adapter and the P2P marketplace enablers.

This document contextualizes the interoperability role of each one of the developed components in the wide IF and identifies each one of repositories containing source code and documentation. Moreover, it details the APIs exposed by each one of the software components and identifies, where applicable, the location for the test instance for that component. Finally, this document provides an overview of the required steps for onboarding a new interoperable service into the IF, providing a step-by-step methodology.

This document should be used as a reference point towards specific details of each one of the developed software components. All information will be maintained and updated in the projects' public Wiki page[1] to allow adopters of interoperability to follow the latest developments, which will be particularly relevant in the open calls that will provide support to entities towards the development, implementation and of interoperable features into their existing (or to be created) solutions and services.

[1] https://gitlab.inesctec.pt/groups/interconnect-public/-/wikis/home

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS AND ACRONYMS

API	Application Programming Interface
DSO	Distribution System Operator
GA	Generic Adapter
IC	InterConnect
IF	Interoperability Framework
IoT	Internet of Things
KE	Knowledge Engine
P2P	Peer-to-Peer
RDF/OWL	Resource Description Framework / Ontology Web Language
REST	Representational State Transfer
SAREF	The Smart Applications REFerence
SHBERA	Smart Energy Refence Architecture
SSA	Service Specific Adapter

1. INTRODUCTION

1.1 WP5 OBJECTIVES

Within the InterConnect project, WP5 “Digital Platforms and Marketplace” oversees the following activities and objectives:

- Establish semantic interoperability between project stakeholders (platforms, services, IoT devices) by leveraging the ontologies, standards and designed specifications (T5.1);
- Demonstrate via the Interoperability Framework how several technologies can create a pluggable and transparent approach while focusing on interfacing functionality-by-design (T5.2);
- Provide security-enabled and a privacy-by-design architecture by considering a mix of cloud-enabled services and legacy systems (T5.3);
- Leverage on the interoperability toolbox to provide P2P marketplace enablers between stakeholders (T5.4);
- Provide continuous support to the project pilots and integrators of the interoperability enablers (T5.5).

This WP is responsible for delivering InterConnect Interoperability Framework as a set of software tools and enablers for facilitating semantic interoperability between digital platforms, services and devices comprising the project pilots. The Interoperability Framework toolset is based on the ontology and the Semantic Interoperability Layer specifications introduced in WP2 and should enable pilot-specific instantiations of the use cases developed within WP1. WP5 will also work on the deployment of distributed ledger technologies tailored for supporting distributed operations, like trading and transactions management activities, by enabling the establishment of P2P marketplaces in pilots with community-based use cases.

1.2 RELATION TO OTHER WPS

As shown in Figure 1, the work carried out in WP5 is based on the work carried out in other technical WPs, while at the same time providing key enablers for the same WPs, namely:

- From WP1, this WP utilizes the use case requirements to infer the architectural requirements the IC Interoperability Framework needs to consider.
- From WP2, WP5 utilizes and develops the concepts and functions (data models, interfaces, protocols, security, and privacy requirements) introduced by the project's Secure Interoperable IoT Smart Home/Building and Smart Energy Reference Architecture (SHBERA). All ontology and semantic interoperability specifications and requirements for the IC Interoperability Framework are provided by WP2.
- WP3 provides interoperable/adapted energy and non-energy services while WP5 provides to WP3 the service store specification and generic adapter for achieving semantic interoperability of the services.

- WP4 provides specification of the DSO interface while WP5 provides integration of the service behind this interface with the Interoperability Framework and interoperable ecosystems established within the pilots.
- WP5 will provide WP7 pilots with the Interoperability Framework toolset as key input for realizing the project use cases leveraging established semantically interoperable ecosystems. The WP7 pilots will provide continuous feedback leading to further updates of the Interoperability Framework.
- WP5 will provide cascade funding projects/partners (WP8) with the toolbox necessary for making their platforms and services interoperable with the Interoperability Framework and established pilots.

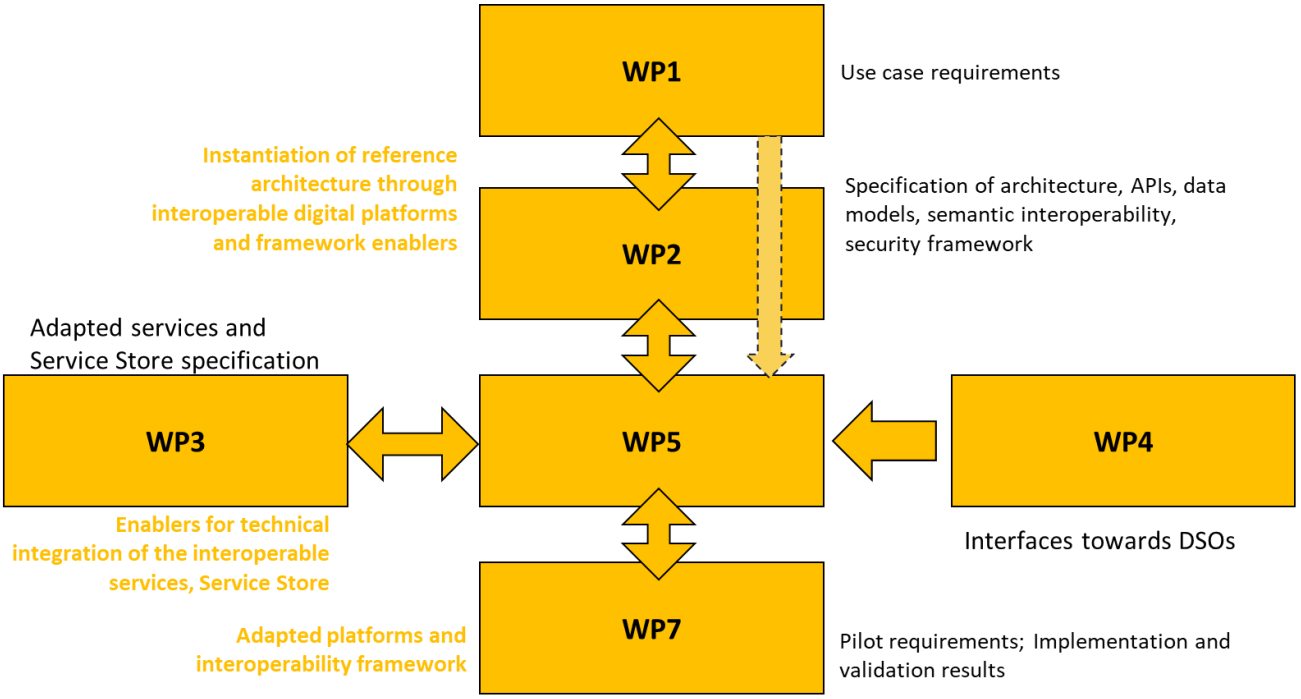


FIGURE 1 - RELATION OF WP5 TO OTHER WPS (ORANGE FONT ARE INPUTS FROM WP5)

1.3 D5.4 OBJECTIVES AND APPROACH

This deliverable is part of the result of the work carried out by T5.2 – Development, test and deployment of the interoperability framework and service store and the T5.4 – P2P marketplace enablers. Its main objectives can be detailed as the development of the specified components and features that will ultimately build the interoperability enablers in InterConnect.

This deliverable document the developed software components as part of the Interoperability Framework, identifying what is the key interoperability role they play, along with the identification of the software repositories and respective documentation resources. The technical documentation (instruction, examples, FAQs and API documentation) for each Interoperability Framework component will be maintained in the corresponding GitLab project. All the documentation is then linked in one page – project public Wiki page.

Finally, this document presents the adopted workflow towards making a service interoperable according to InterConnect approach, while being supported in the exchange of semantic knowledge according to SAREF.

This document does is not a user guide.

It serves as a supporting element with the identification of relevant developments made in the scope of WP5 and a guide and link to key resources concerning each one of the software systems developed. Detailed information and user guide references can be found in the documentation of each one of the components in the project public wiki[1]. The content published there will be regularly updated according to the improvements and additional development s carried out.

1.4 DOCUMENT STRUCTURE

This introduction is part of **Chapter 1**. The overall organization of the chapters follows the WP5 methodology and the Interoperability Framework specification.

Chapter 2 – Interoperability Framework provides a short description of the interoperability framework for the readers' reference and positioning towards the described developments.

Chapter 3 – Semantic Interoperability Layer provides the identification of all resources concerning the Knowledge Engine software system, including references to the source code and the respective documentation.

Chapter 4 – Service Store provides the identification of all resources concerning the Service Store software system, including references to the source code and the respective documentation.

Chapter 5 – Generic Adapter provides the identification of all resources concerning the Generic Adapter software system, including references to the source code and the respective documentation.

Chapter 6 – P2P Marketplace Enablers provides the identification of all resources concerning the Blockchain P2P Marketplace Enablers software system, including references to the source code and the respective documentation.

Chapter 7 – Integration Workflow provides a step-by-step description of the process required for a service provider in order to make corresponding service interoperable and onboard it into the InterConnect semantically interoperable ecosystem.

2. INTEROPERABILITY FRAMEWORK

The InterConnect Interoperability Framework provides a set of tools, enabling the provisioning of SAREF-centric semantic interoperability for digital platforms, services, and digital assets such as connected devices.
This deliverable identifies the **key components of the interoperability framework** and the respective APIs, software artifacts and documentation.

The high-level functional architecture of Interconnect's interoperability framework is depicted in Figure 2.

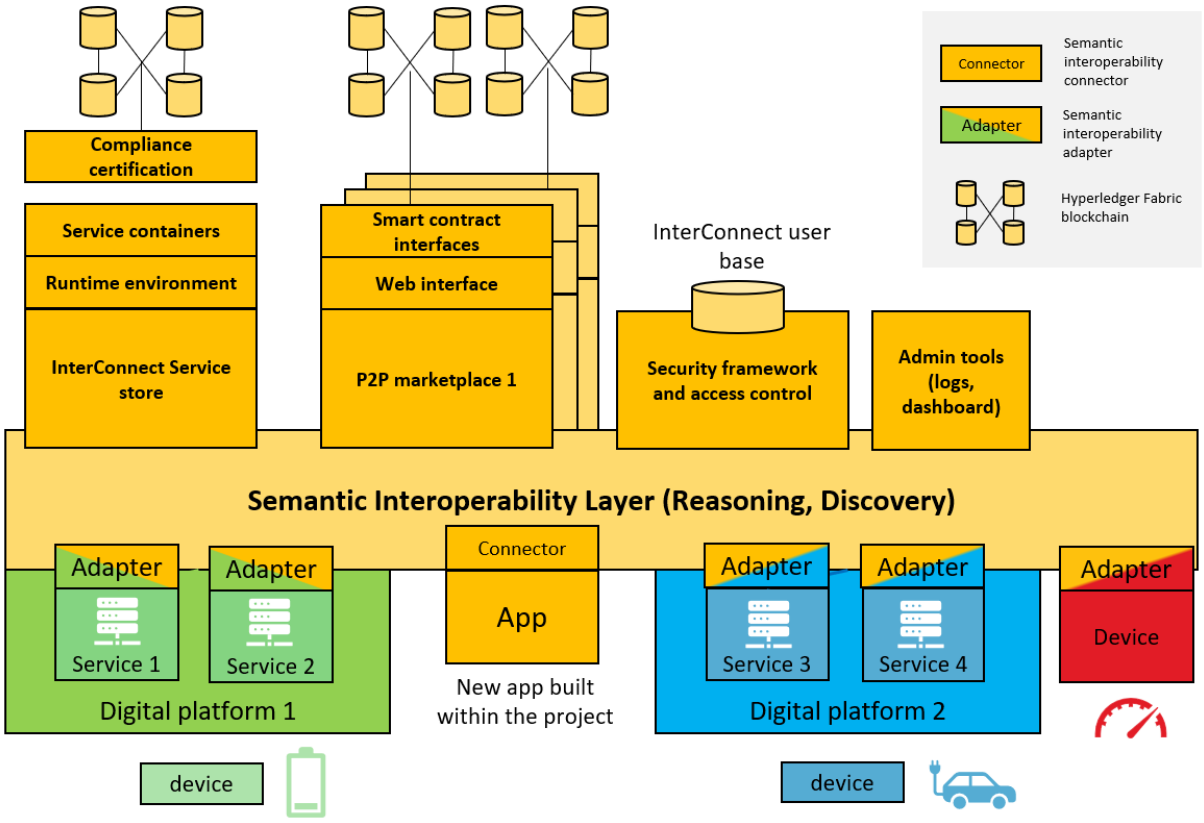


FIGURE 2 - HIGH LEVEL FUNCTIONAL ARCHITECTURE OF THE IC INTEROPERABILITY FRAMEWORK

The Interconnect interoperability framework includes the following components:

- The Semantic Interoperability Layer;
- The Service Store;
- The Generic Adapter;
- The P2P Marketplace enablers;

Each component is characterized in the following sections. More details on each component role and specification can be found in D5.1 [3].

3. SEMANTIC INTEROPERABILITY LAYER

3.1 INTEROPERABILITY ROLE

The InterConnect's Semantic Interoperability Layer is based on Knowledge Engine technology supplied by partner TNO and developed for InterConnect use in the scope of WP2 and WP5. The Knowledge Engine is a technology aimed at providing semantic interoperability by means of two features: *translation* and *discovery*. Both these features require a common ontology. The ontology of choice for the InterConnect Interoperability Framework is SAREF and its extensions. Notice that the Knowledge Engine is ontology agnostic and, in principle, can work with any ontology if it is expressed in the RDF/OWL format. From here on we consider SAREF as the common ontology used by the Knowledge Engine and in InterConnect.

3.2 SOFTWARE ARTIFACTS AND REPOSITORY

The Knowledge Engine is built from a series of sub-components, identified in the following table. All sub-components correspond to the latest version available – Release 1.0.3.

TABLE 1 - KNOWLEDGE ENGINE SOFTWARE ARTIFACTS.

KNOWLEDGE ENGINE – RELEASE 1.0.3		
SUB-COMPONENT	**DESCRIPTION**	**URL**
Smart-connector	Knowledge Engine connector to interact with the semantic exchange API	https://gitlab.inesctec.pt/interconnect/knowledge-engine/-/tree/master/admin-ui
Smart-connector-api	Knowledge Engine Smart Connector API	https://gitlab.inesctec.pt/interconnect/knowledge-engine/-/tree/master/smart-connector-api
Smart-connector-rest-server	Knowledge Engine Rest Server – Non-Distributed version	https://gitlab.inesctec.pt/interconnect/knowledge-engine/-/tree/master/smart-connector-rest-server
Smart-connector-rest-dist	Distributed Knowledge Engine Runtime Server	https://gitlab.inesctec.pt/interconnect/knowledge-engine/-/tree/master/smart-connector-rest-dist
Admin-ui	Command-line administer	https://gitlab.inesctec.pt/interconnect/knowledge-engine/-/tree/master/admin-ui
Knowledge-directory	Knowledge Engine distributed instance locator	https://gitlab.inesctec.pt/interconnect/knowledge-engine/-/tree/master/knowledge-directory

The software artifacts and source code are available in the project repository in INESC TEC's Gitlab instance. The instance with source code is available with partner login. Public group access is provided with access to the wide public to the software artifacts, documentation, and examples.

If technical constraints prevent you from accessing the repositories, please contact the support team[2].

3.3 DOCUMENTATION

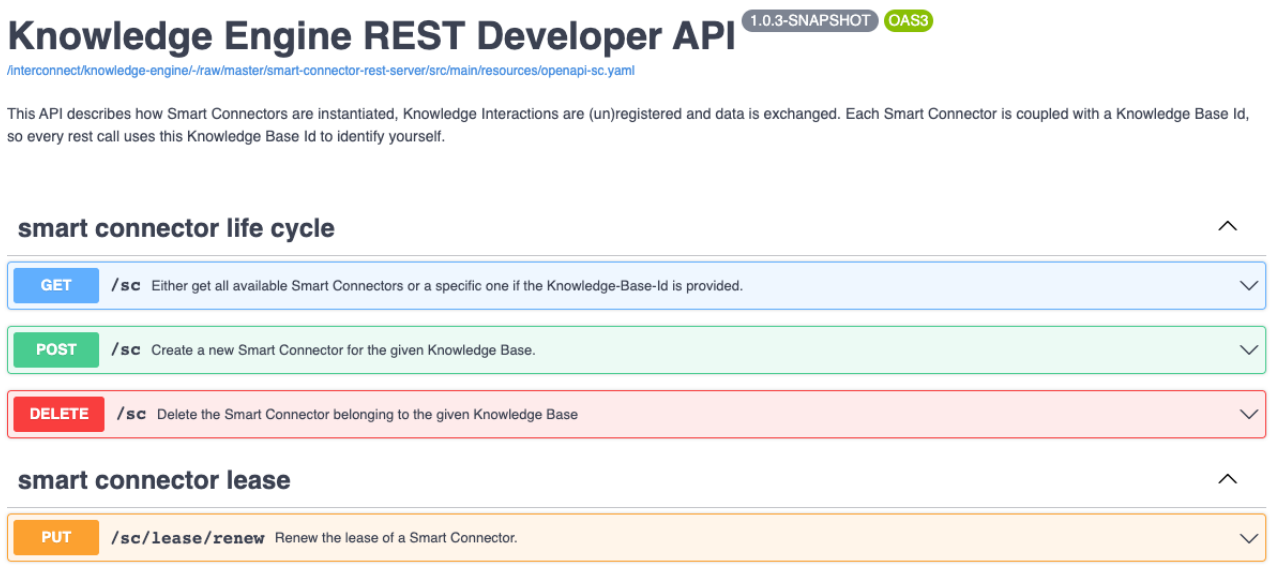
The documentation is available in the project repository in INESC TEC's Gitlab instance. Public group access is provided with access to the wide public to the software artifacts, documentation, and examples.

TABLE 2 - KNOWLEDGE ENGINE DOCUMENTATION.

KNOWLEDGE ENGINE – RELEASE 1.0.3		
SUB-COMPONENT	**DESCRIPTION**	**URL**
docs	Knowledge Engine documentation	https://gitlab.inesctec.pt/interconnect/knowledge-engine/-/tree/master/docs

3.4 API SPECIFICATION

Figure 3 shows snapshot of the OpenAPI specification of the Knowledge Engine REST developer API.



[2] interconnect-support@lists.inesctec.pt

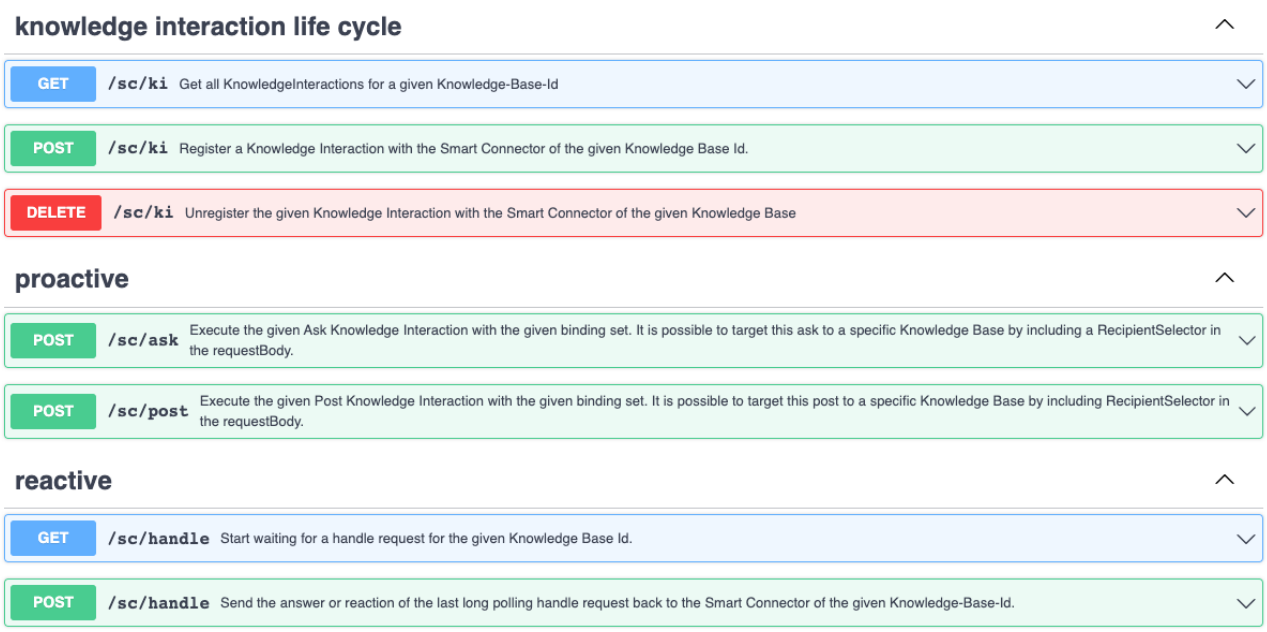


FIGURE 3 - KNOWLEDGE ENGINE REST DEVELOPER API

3.5 RUNNING TEST INSTANCE

A test running instance of the Knowledge Engine is available in the URL detailed in Table 3. API requests can be directed at this endpoint for testing purposes.

TABLE 3 - KNOWLEDGE ENGINE RUNNING TEST INSTANCE.

KNOWLEDGE ENGINE – RELEASE 1.0.3		
SUB-COMPONENT	**DESCRIPTION**	**URL**
Smart-connector-rest-server	Knowledge Engine instance	https://ke.interconnectproject.eu/rest

4. SERVICE STORE

4.1 INTEROPERABILITY ROLE

As one of the main IC Interoperability Framework tools, the IC service store will provide a single stop for all providers and adopters of interoperable services from energy and non-energy domains. The service store is conceptualized as a web service with its front-end and back-end modules and processes. The main objective is to enable building of the InterConnect ecosystem of service providers and adopters by allowing them to register new interoperable services and browse existing ones to identify services best suited for the challenge at hand and get all necessary information for accessing and properly utilizing selected services. Moreover, the service store offers also other features as the interoperability compliance of services, identity management, knowledge explorer or test tools to assist the integration of new services.

4.2 SOFTWARE ARTIFACTS AND REPOSITORY

The Service Store component is built from two sub-components as depicted in Table 4. All sub-components correspond to the latest public version available – Release 1.1.

TABLE 4 - SERVICE STORE SOFTWARE ARTIFACTS.

SERVICE STORE – RELEASE 1.1		
SUB-COMPONENT	**DESCRIPTION**	**URL**
Service Store – Backend	Service Store Backend System	https://gitlab.inesctec.pt/interconnect-public/service-store-backend
Service Store – Frontend	Service Store Frontend System	https://gitlab.inesctec.pt/interconnect-public/service-store-frontend

4.3 DOCUMENTATION

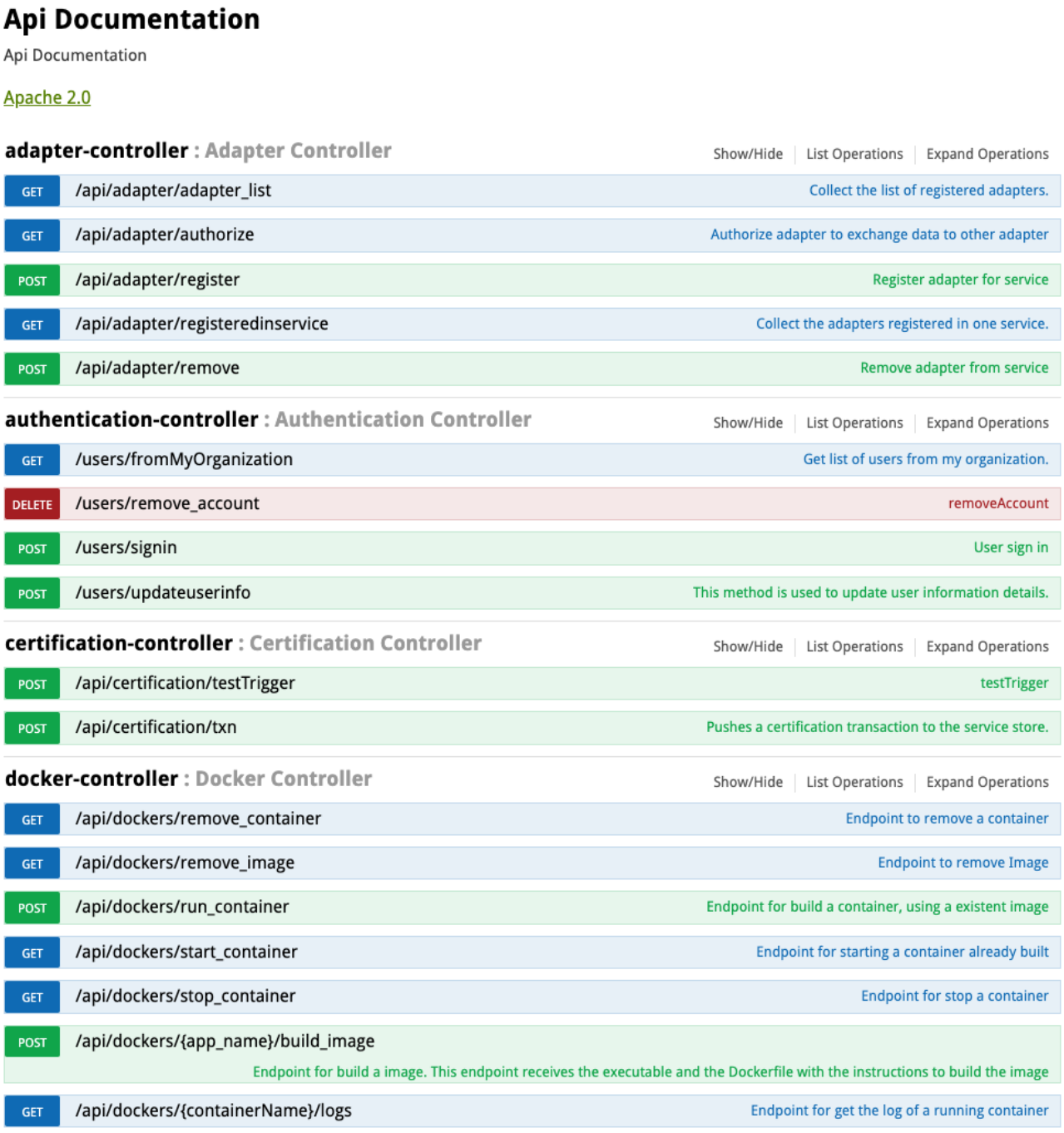
The documentation is available in the project repository in INESC TEC's Gitlab instance. Public group access is provided with access to the wide public to the software artifacts, documentation, and examples.

TABLE 5 - SERVICE STORE DOCUMENTATION.

SERVICE STORE – RELEASE 1.1		
SUB-COMPONENT	**DESCRIPTION**	**URL**
Service Store – documentation	Service Store documentation and user guide	https://gitlab.inesctec.pt/interconnect-public/service-store-frontend/-/wikis/home

4.4 API SPECIFICATION

Figure 4 shows snapshot of the OpenAPI specification of the Service Store backend system REST developer API.



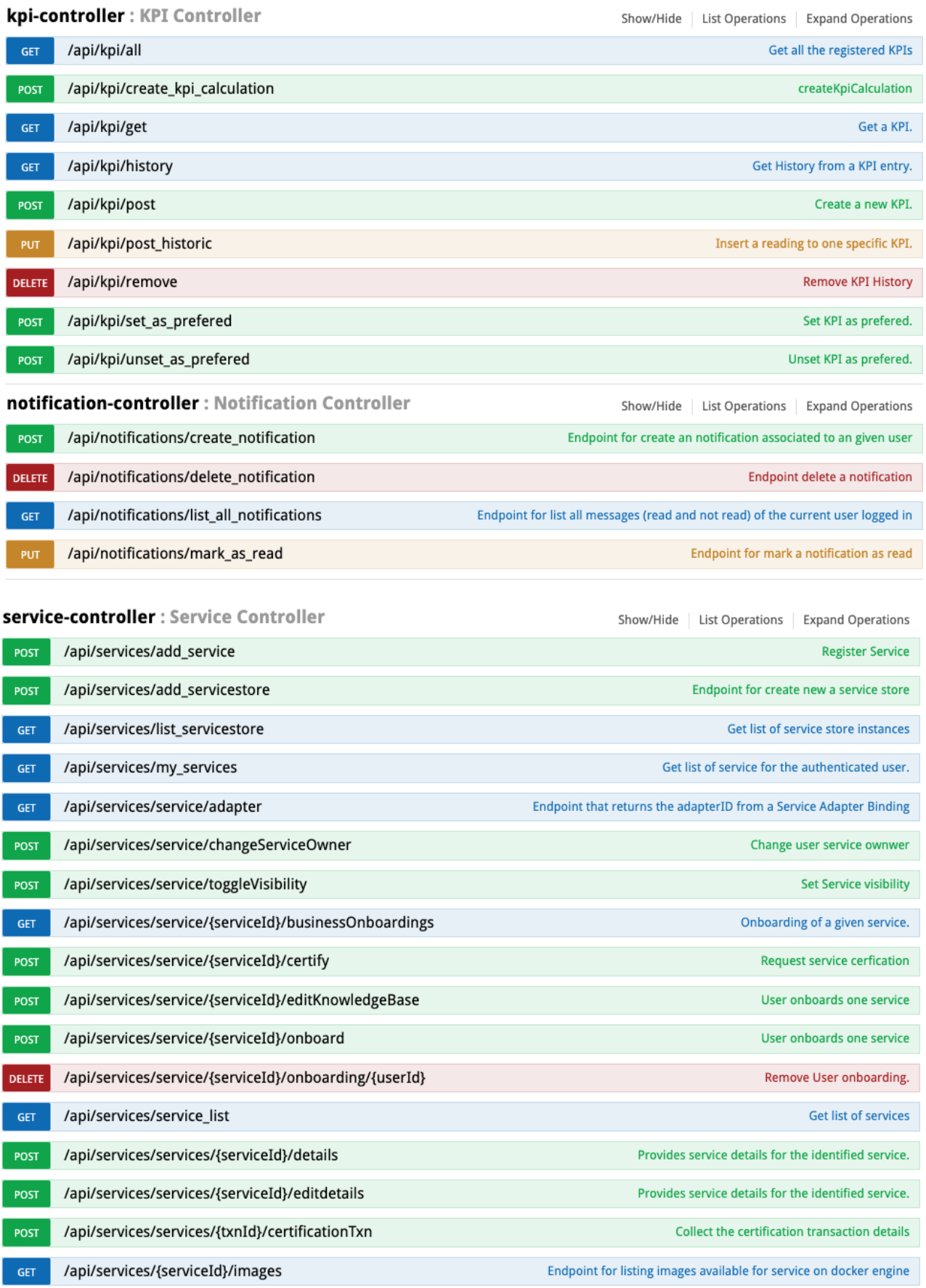


FIGURE 4 - SERVICE STORE BACKEND API

4.5 RUNNING TEST INSTANCE

A test running instance of the Service Store is available in the URL detailed in Table 6. API requests can be directed at this endpoint for testing purposes.

TABLE 6 – SERVICE STORE RUNNING TEST INSTANCE.

SERVICE STORE – RELEASE 1.1		
SUB-COMPONENT	**DESCRIPTION**	**URL**
Service Store – Backend	Service Store Backend System	https://store.interconnectproject.eu/backend
Service Store – Frontend	Service Store Frontend System	https://store.interconnectproject.eu/ServiceStore

5. GENERIC ADAPTER

5.1 INTEROPERABILITY ROLE

The GA is a software gateway for secure and trusted communication between a service and a wider Interoperability Framework instance. The GA provides unified REST API towards Service Specific Adapter. This GA REST API ensures communication with the Service Store for authentication and authorization of the service and the GA itself with the central identity provider. The GA REST API also facilitates interactions with the semantic interoperability layer (Knowledge Engine instance) by providing methods for Knowledge Base and Knowledge Interaction registration and also methods for executing corresponding knowledge exchanges.

5.2 SOFTWARE ARTIFACTS AND REPOSITORY

The Generic Adapter component is depicted in Table 7. All sub-components correspond to the latest public version available – Release 1.1.

TABLE 7 - GENERIC ADAPTER SOFTWARE ARTIFACTS.

GENERIC ADAPTER – RELEASE 1.1		
SUB-COMPONENT	**DESCRIPTION**	**URL**
Generic Adapter	Generic Adapter	https://gitlab.inesctec.pt/interconnect-public/generic-adapter
Generic Adapter - docker	Generic Adapter – Docker version	https://gitlab.inesctec.pt/interconnect/generic-adapter/-/tree/docker-bash https://gitlab.inesctec.pt/interconnect-public/generic-adapter/container_registry/313

5.3 DOCUMENTATION

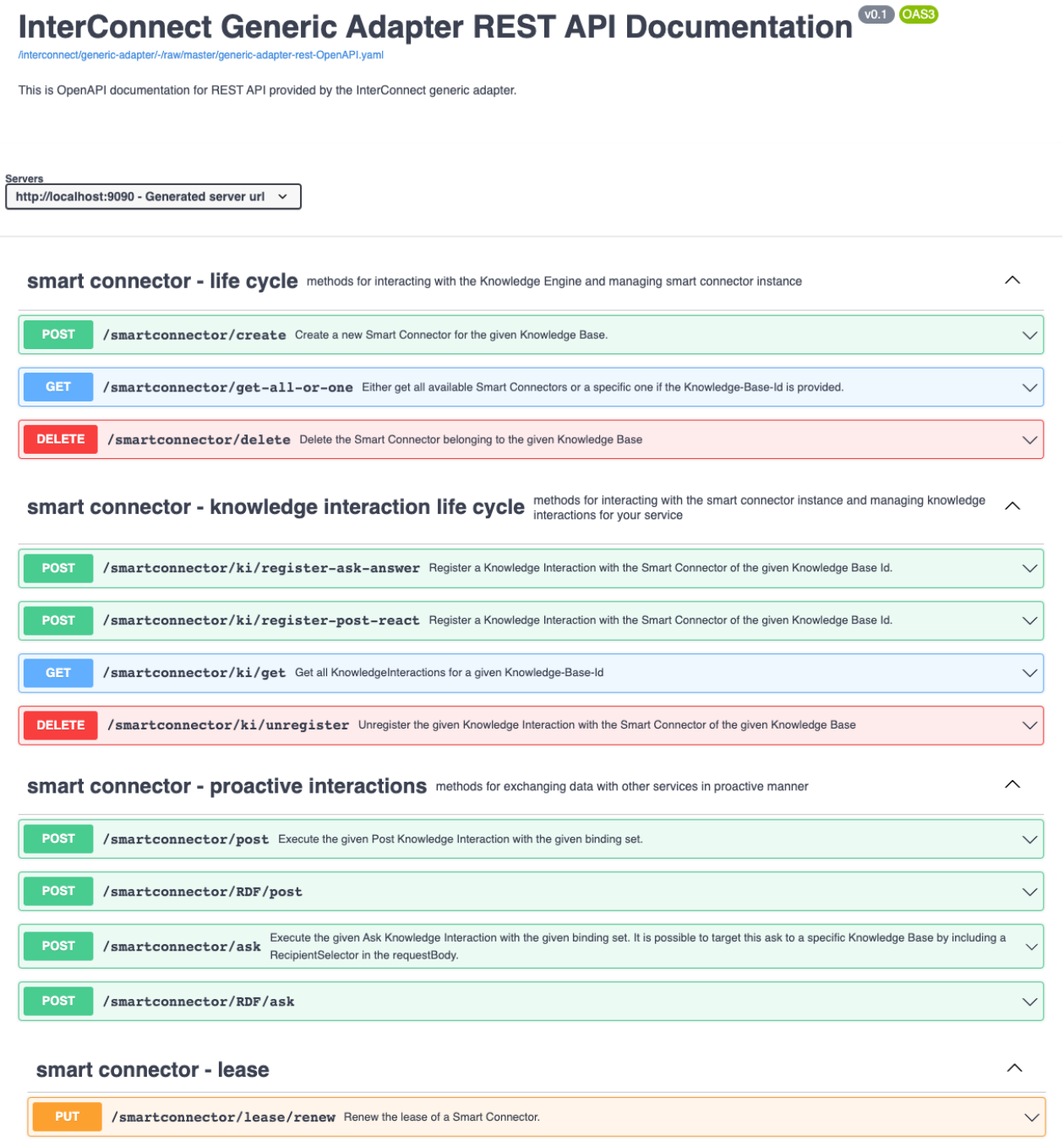
The documentation is available in the project repository in INESC TEC's Gitlab instance. Public group access is provided with access to the wide public to the software artifacts, documentation, and examples.

TABLE 8 -GENERIC ADAPTER DOCUMENTATION.

GENERIC ADAPTER – RELEASE 1.1		
SUB-COMPONENT	**DESCRIPTION**	**URL**
Generic Adapter – documentation	Generic Adapter documentation and user guide	https://gitlab.inesctec.pt/interconnect-public/generic-adapter/-/tree/master/docs

5.4 API SPECIFICATION

Figure 5 shows snapshot of the OpenAPI specification of the Generic Adapter backend system REST developer API.



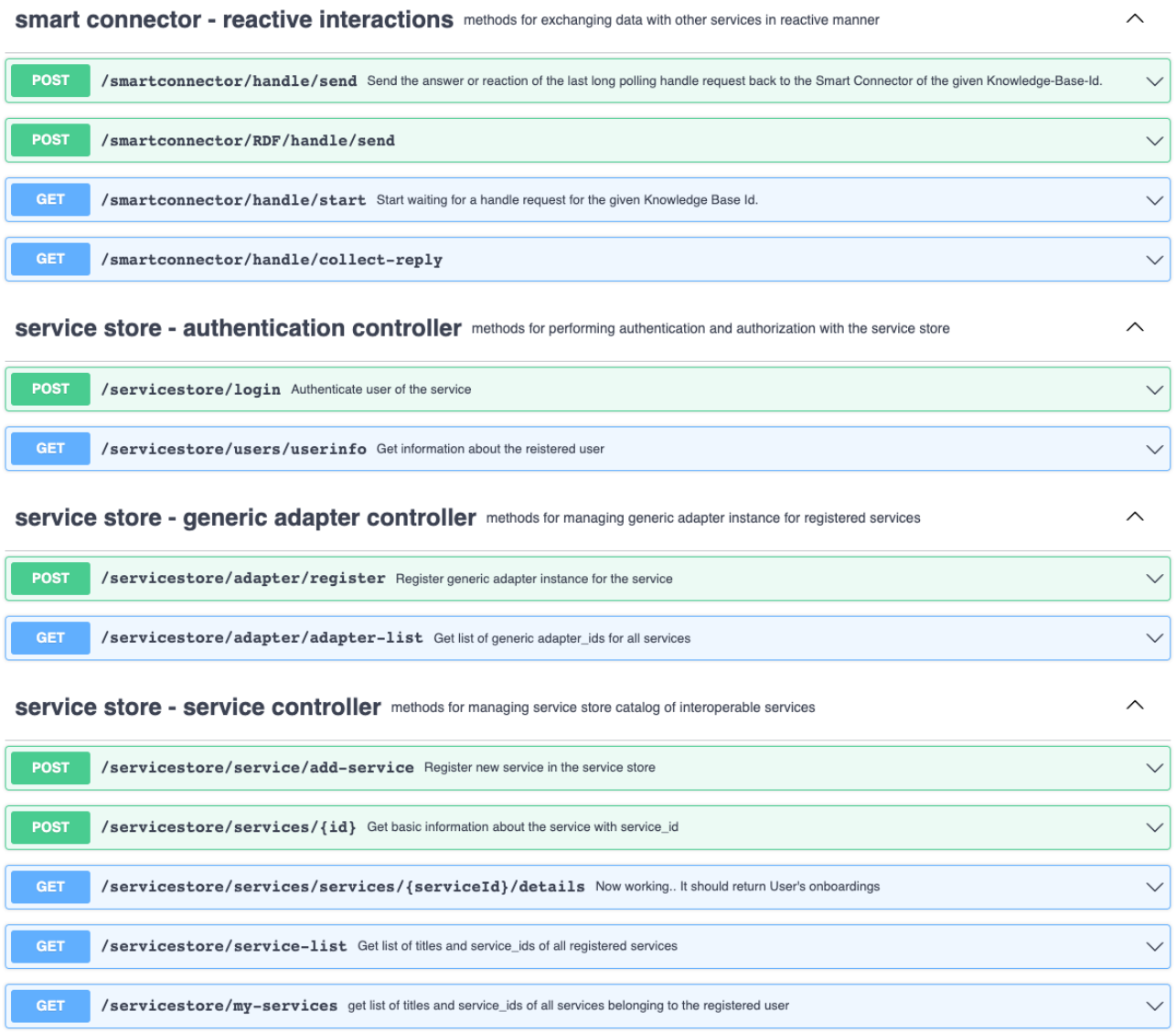


FIGURE 5 – GENERIC ADAPTER REST API

6. P2P MARKETPLACE ENABLERS

6.1 INTEROPERABILITY ROLE

The Interoperability Framework provides P2P marketplace enablers as part of its toolbox. The enablers are provided as deployable containers that allow pilot owners and integrators to deploy and fully manage P2P marketplace instances. The established P2P marketplaces are in full control and under jurisdiction (regulatory, market wise, data privacy protection) of the integrators. The goal was to develop enablers which would allow establishment of blockchain ledgers shared between community members and supporting community specific services for data exchange in the project pilots. The P2P marketplace enablers include:

- **Hyperledger Fabric blockchain configurations** for different types of P2P marketplaces (different hierarchies, consortium organizations, stakeholders, nodes, and channels).
- **Smart contract templates** for different types of orders and transactions to be featured in the marketplace. Smart contracts will include APIs for end user GUIs (web application) and APIs for services for automated P2P trading.
    - Smart contract templates for generating reports and audits about status of the marketplace and executed transactions - in line with regulatory and business requirements.
    - Smart contract implemented as semantic interoperability adapter for interfacing with the wider InterConnect Interoperability Framework.
    - Smart contracts for registering and identifying key actors and resources constituting P2P marketplaces.
    - Smart Contracts for integration of interoperable services which write data to or read data from the Hyperledger Fabric. This ensures that the P2P marketplaces are integrated with a wider instance of Interoperability Framework.
- **Configurable order matching engine** for managing regulatory constraints, transaction priorities and conflict resolutions. The ordering engine also chains the smart contract calls performed by services participating in the P2P marketplace.
- **White-labelled web application** for providing interface through which end users place orders. The web application can be instantiated and adapted to specific needs of a community establishing the P2P marketplace.

The P2P marketplace can be an energy marketplace or a marketplace for data transactions required for the realization of the community-based use cases. In the scope of the Task 5.4 the following P2P marketplace configurations are implemented and made available for the pilots and 3rd party integrators:

- Generic data marketplace configuration that can be easily adapted to specific use cases in energy and IoT domains. This implementation also features a web application for demonstration purposes.
- Specialized configurations based on participating pilots (each having specific set of requirements for integration of P2P marketplaces into instances of the Interoperability Framework):

- Dutch pilot - IoT data exchange with loyalty tokens realized through P2P marketplace.
- Portuguese pilot - flexibility and energy profile trading/aggregation in communities through P2P marketplace instance.
- Belgium pilot from Think E! and VUB - P2P energy trading within and between energy communities through instantiated P2P marketplace.

6.2 SOFTWARE ARTIFACTS AND REPOSITORY

The P2P marketplace configurations are listed in Table 7. All the P2P marketplace configurations are supplied to the corresponding pilots for integration with their instances of the Interoperability Framework.

TABLE 9 - P2P MARKETPLACE ENABLERS SOFTWARE ARTIFACTS.

P2P MARKETPLACE ENABLERS		
SUB-COMPONENT	**DESCRIPTION**	**URL**
Portuguese pilot configuration	P2P marketplace configuration for the Portuguese pilot use case focusing on flexibility and energy profile trading/aggregation in communities	https://gitlab.inesctec.pt/interconnect-public/p2p-marketplace/-/tree/p2p-flex-sharing
Belgian pilot configuration	P2P marketplace configuration for the Belgian sub-pilots with focus on energy trading within and between communities.	https://gitlab.inesctec.pt/interconnect-public/p2p-marketplace/-/tree/p2p-energy-trading
Dutch pilot configuration	P2P marketplace configuration for the Dutch pilot and its use case on IoT data exchange with loyalty tokens.	https://gitlab.inesctec.pt/interconnect-public/p2p-marketplace/-/tree/p2p-data-mp-activities-mg
Generic energy P2P marketplace configuration	P2P marketplace configuration that can be used as basis for building new P2P marketplaces in other pilots and for cascade funding. Also contains a PoC web application for texting and demonstrating purposes.	https://gitlab.inesctec.pt/interconnect-public/p2p-marketplace/-/tree/p2p-sample-marketplace

6.3 DOCUMENTATION

The documentation is available in the project repository in INESC TEC's Gitlab. Public group access is provided to the software artifacts, documentation, and examples.

TABLE 10 - MARKETPLACE ENABLERS DOCUMENTATION.

P2P MARKETPLACE ENABLERS		
SUB-COMPONENT	**DESCRIPTION**	**URL**
P2P Marketplace Enablers – documentation	P2P Marketplace Enablers documentation - root for all branches	https://gitlab.inesctec.pt/interconnect-public/p2p-marketplace

Portuguese pilot P2P marketplace documentation	Documentation with diagrams, instructions, and link for tutorial video	https://gitlab.inesctec.pt/interconnect-public/p2p-marketplace/-/tree/p2p-flex-sharing
Belgian pilot P2P marketplace documentation	Documentation with diagrams, instructions, and link for tutorial video	https://gitlab.inesctec.pt/interconnect-public/p2p-marketplace/-/tree/p2p-energy-trading
Dutch pilot P2P marketplace documentation	Documentation with diagrams, instructions, and link for tutorial video	https://gitlab.inesctec.pt/interconnect-public/p2p-marketplace/-/tree/p2p-data-mp-activities-mg
Generic energy P2P marketplace documentation	Documentation with diagrams, instructions, and link for tutorial videos	https://gitlab.inesctec.pt/interconnect-public/p2p-marketplace/-/tree/p2p-sample-marketplace

6.4 API SPECIFICATION

In P2P marketplace enablers and their use case specific configurations, the smart contracts expose REST APIs. Documentation of REST API methods is prepared with the help of Postman collections. Through these REST APIs, the P2P marketplaces can be integrated with other interoperable services and support deployment of custom UIs for engaging end users. Instructions

1. Download Postman from this link: https://www.postman.com/downloads/
2. Open the Postman application and import all files from the folder 'postman'. 1 collection and 4 environments should appear.
3. Test if the REST APIs are responding properly by issuing the 'Register EndUser' API from Postman. The response should return a success message & token.

TABLE 11 - P2P API SPECIFICATION.

P2P MARKETPLACE ENABLERS	
SUB-COMPONENT	**DESCRIPTION**
Portuguese P2P marketplace configuration	https://gitlab.inesctec.pt/interconnect-public/p2p-marketplace/-/tree/p2p-flex-sharing/postman
Belgian P2P marketplace configuration	https://gitlab.inesctec.pt/interconnect-public/p2p-marketplace/-/tree/p2p-energy-trading/postman
Dutch P2P marketplace configuration	https://gitlab.inesctec.pt/interconnect-public/p2p-marketplace/-/tree/p2p-data-mp-activities-mg/postman

7. INTEGRATION WORKFLOW

The Interoperability Framework provides a set of software tools that enable semantic interoperability between digital services or components within a digital platform. This enables data exchange to occur based on ***knowledge dissemination***. This fundamental change eases the integration process with other services or digital platforms, as the focus shifts from the effort to integrate a foreign data model and protocol – the common approach to *syntactic interoperability* – to address what is the ***knowledge*** that is required to be disseminated.

This conceptual difference should ease the adoption of interoperable technology as it shifts the discussion of key stakeholders from technical or data modelling discussions to a domain representation discussion.

Figure 6 depicts the concept of *knowledge base*, where all the knowledge exchanged to a from a digital service will reside. This conceptual representation is built from several software components, namely: the digital service that will become interoperable, the service specific adapter (SSA) and the generic adapter (GA).

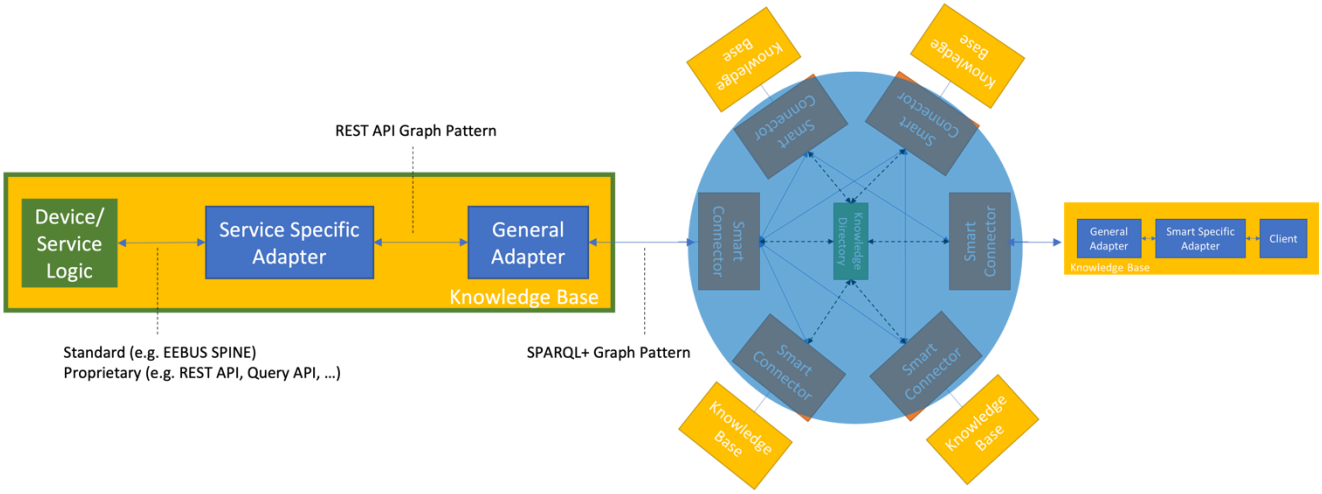


FIGURE 6 - KNOWLEGE BASE AND GENERIC ADAPTER CONCEPT.

The Service Specific Adapter (SSA) defines the client code that the digital service will use to integrate with the API from the Generic Adapter. It will hold all the knowledge descriptions and will be used to feed data into the ecosystem.

The Generic Adapter (GA) establishes the gateway towards the wider InterConnect interoperability Framework.

Figure 7 depicts the digital service that will be made interoperable. The legacy interface of that service will be required to undergo a process to identify and map the service concepts into the ontology in use. Interconnect departs and extends the SAREF family of ontologies. With all parameters and service concepts identified, the SSA will hold the logic/ data wrapper that implements the ontology-based semantic interaction scheme and data models. The SSA will also interface with the Generic Adapter API.

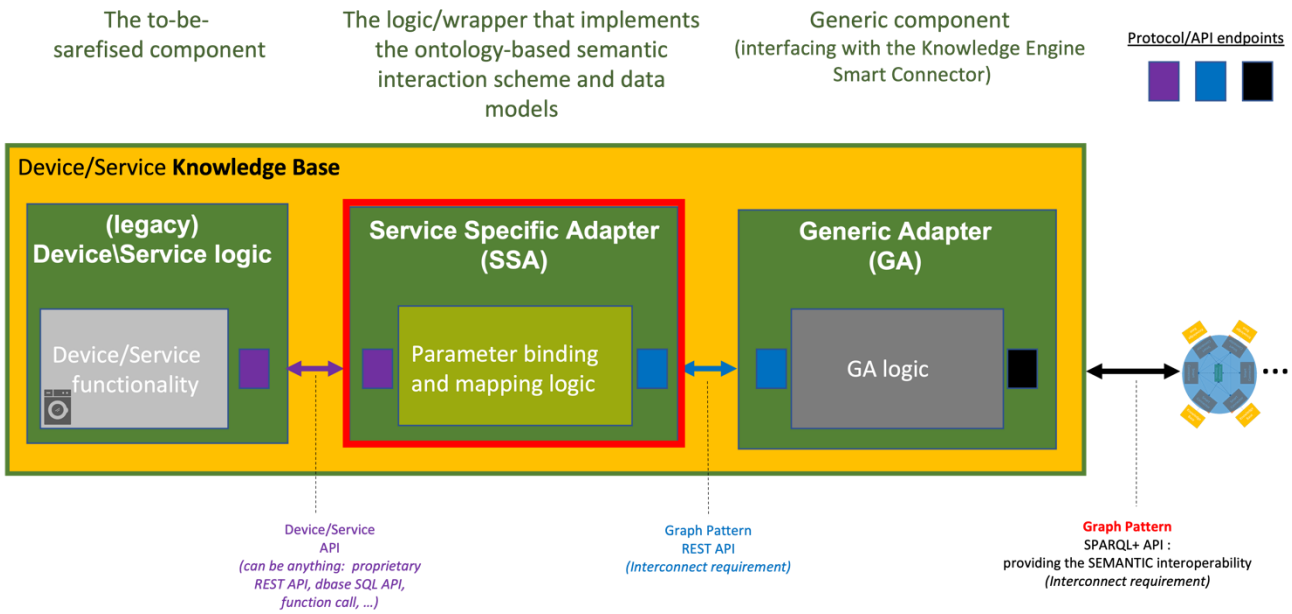


FIGURE 7 - SERVICE SPECIFIC ADAPTER CONCEPT.

7.1 PROCESS TO INTEGRATE A SERVICE

Based on the previous concepts, a digital service requires to undergo as series of steps. This section is not exhaustive. In step, a URL is provided with specific instructions for each step.

1. Create an account and register a service using Interconnect's Service Store. This step establishes the first step into the Interoperability framework, creating the registry for each interoperable service.

#	Where	Action	Documentation for Action
1.1	Service Store	Create and validate Account	https://gitlab.inesctec.pt/interconnect-public/service-store-frontend/-/wikis/1.1-User-Guide-:-Register-User-Account
1.2	Service Store	Register Service	https://gitlab.inesctec.pt/interconnect-public/service-store-frontend/-/wikis/1.3-User-Guide-:-Register-Service

2. Download and instantiate a Generic Adapter instance for your service. Acquiring the Generic Adapter will allow services to start the integration by instantiating the gateway to the Interoperability framework in setting the privacy choices for each service.

#	Where	Action	Documentation for Action
2.1	Interconnect Repository	Download Generic Adapter	https://gitlab.inesctec.pt/interconnect-public/generic-ada
2.2	Local instance	Start the Generic Adapter	https://gitlab.inesctec.pt/interconnect-public/generic-ada /blob/master/docs/01_preparations.md

3. Register the Generic Adapter instance for your service. By registering the generic adapter, a binding will be established between a generic adapter instantiation and a service record in the Service Store.

#	Where	Action	Documentation for Action
3.1	Generic Adapter	Register the Generic Adapter by including the service reference.	https://gitlab.inesctec.pt/interconnect-public/generic-adapter/-/blob/master/docs/02_service_store.md

4. Instantiation of the KE or KER and KD. Instantiating a privately hosted instance of the KE will allow (e.g.,) an instance per pilot or per building. Moreover, the KE can leverage from its distributed design when using the KER, composed of a series of federated KE instances. The latter requires a KD deployment, acting as a discovery point to all available KER instances.

#	Where	Action	Documentation for Action
3.1	Knowledge Engine	Instantiation of the KE or KER + KD	https://gitlab.inesctec.pt/interconnect/knowledge-engine

5. Build the Service Specific Adapter for your service. The creation of the SSA will establish the kind of knowledge interactions, that is, the knowledge a service is able to *serve* and how it will relate with other services in the interoperable ecosystem. This is achieved through the ontological graph pattern descriptions, based on SAREF.

#	Where	Action	Documentation for Action
5.1	Interconnect Repository	Refer to the Interconnect ontology repository.	https://gitlab.inesctec.pt/interconnect-public/ontology
5.2	Interconnect Repository	Get acquainted with the ASK, ANSWER, POST,REACT knowledge interactions.	https://gitlab.inesctec.pt/interconnect/knowledge-engine/-/blob/master/docs/01_concept.md
5.3	Interconnect Repository	Follow the process to build the SSA.	https://gitlab.inesctec.pt/groups/interconnect-public/-/wikis/home#service-specific-adapters

6. Test the SSA integration with the Generic Adapter instance. The test stage allows requests to be proxied by the generic adapter and to collect the feedback from the knowledge sent and exchange via the SIL, particularly via the KE. It also takes part in the certification flow of services, allowing other services to acknowledge what is the development stage of a given service.

#	Where	Action	Documentation for Action
6.1	Generic Adapter	Test the SSA integration with the GA by checking logs.	https://gitlab.inesctec.pt/interconnect-public/generic-adapter/-/blob/master/docs/08_faq.md
6.2	Service Store	Verify the certification results that also hold the logs.	https://gitlab.inesctec.pt/interconnect-public/service-store-frontend/-/wikis/1.4-User-Guide-:-Service-Certification

REFERENCES

INTERCONNECT DOCUMENTS

[1] InterConnect Grant Agreement number 857237.

[2] InterConnect project. “D5.3 Security, cyber-security and privacy protection action plan and results”. 2021.

[3] InterConnect project. “D5.1 Concept, design and architecture of the interoperable marketplace toolbox”. 2022.

[4] Interconnect Public Wiki and repositories, https://gitlab.inesctec.pt/groups/interconnect-public/-/wikis/home