interconnect

interoperable solutions connecting smart homes, buildings and grids

WP5 – Digital Platforms and Marketplace

D5.2

Data Flow Management

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

DOCUMENT INFORMATION

DOCUMENT	D5.2 – Data Flow Management
TYPE	REPORT
DISTRIBUTION LEVEL	PUBLIC
DUE DELIVERY DATE	30/09/2020
DATE OF DELIVERY	30/09/2020
VERSION	V1.0
DELIVERABLE RESPONSIBLE	VLF
AUTHOR (S)	INESCTEC,VLF,TNO,TRIALOG
OFFICIAL REVIEWER/s	INESCTEC,VIZLORE,TRIALOG,CYBERGRID,SENSINOV, VITO

DOCUMENT HISTORY

VERSION	AUTHORS	DATE	CONTENT AND CHANGES
0.1	Fábio Coelho (INESCTEC)	10/09/2020	First draft for all sections and specific sequence diagram chart.
0.2	Amelie Gyrard (Trialog)	22/09/2020	Review
0.3	Barry Nouwt (TNO)	22/09/2020	Knowledge Engine (section 2) Data Flow Specification for Interoperability layer (section 3)
0.4	Amelie Gyrard (Trialog) Guillaume Mockly (Trialog) Antonio Kung (Trialog) Yannick Huc (Trialog)	23/09/2020	sequence diagram for security, authentication, and authorization (section 3)
0.5	Andraž Andolšek (cyberGRID)	23/09/2020	Review
0.6	Georg Jung(VITO) Dominic Ectors (VITO)	25/09/2020	Review
0.7	Milenko Tosic	25/09/2020	Review
0.8	Eliana Valles	29/09/2020	Review
1.0	Fábio Coelho (INESCTEC)	30/09/2020	Final edit and consolidation

ACKNOWLEDGEMENTS

NAME	PARTNER
Laura Daniele	TNO
Barry Nouwt	TNO
Wilco Wijbrandi	TNO
Guillaume Mockly	TRIALOG
Antonio Kung	TRIALOG
Yannick Huc	TRIALOG
Paulo Monteiro	INESCTEC
David Rua	INESCTEC
Fábio Coelho	INESCTEC
Eliana Valles	SENSINOV
Milenko Tosic	VLF
Dragan Boscovic	VLF
Ognjen Ikovic	VLF
Dominic Ectors	VITO
Georg Jung	VITO

DISCLAIMER:

The sole responsibility for the content lies with the authors. It does not necessarily reflect the opinion of the CNECT or the European Commission (EC). CNECT or the EC are not responsible for any use that may be made of the information contained therein.

EXECUTIVE SUMMARY

Deliverable D5.2 – Data Flow Management is the second deliverable to be submitted on behalf of WP5 – Digital Platforms of the InterConnect Project that received funding from the European Union's Horizon 2020 Research and Innovation program under the Grant Agreement (GA) number 857237.

This deliverable is part the outcome of the work carried out in task T5.1 - Interoperability Framework and Service Store Architecture and specification [M7-M12]. It uses and develops the output and ongoing work of other WPs. Hence, this deliverable and its related task:

- Uses the High-Level Use Cases developed within WP1 to **analyse and specify each (sub-)pilot's preliminary architectural implementation;**
- Further develop InterConnect's Secure interoperable IoT smart home/building and smart energy system reference architecture (SHBERA), developed within WP2, to **specify the project's Interoperability Framework and other interoperable resources and services**;
- Contributes to the preliminary **specification of the Semantic Interoperability Layer**, developed within WP2, to identify the set of connectors and adaptors required for ensuring interoperability on a syntactic and semantic level;
- Collaborates closely with WP3 on **defining the set of interoperable services and applications needed for pilot implementation** and validation of results, due to take place during WP7.

More precisely, D5.2 pursues essential work for other tasks in WP5, namely:

- It **provides the overall view** for the interoperability framework and its basic functionalities in terms of **data management flow**;
- It **provides the concept** for the provision of **semantic interoperability** and details and relationship with **reasoning and discovery features**;
- It **introduces and details** how data is handled by the **interoperability adapters** and how to establish **data boundaries**;
- Defines a preliminary set of **sequence diagrams** for the relevant functional interactions and functionalities of the **interoperability framework, service store and P2P communities**.

These concepts and the methodology used to achieve these results are described in detail in the next sections.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS AND ACRONYMS

IoT	Internet of Things
WP	Work Package
P2P	Peer-to-peer
IC	InterConnect
DSO	Distribution System Operator
TRL	Technology Readiness Level
ICT	Information and Communication Technologies
API	Application Programming Interface
REST	Representational State Transfer
MQTT	MQ Telemetry Transport
SAREF	Smart Appliances Reference Ontology
RDF	Resource Description Framework
GDPR	General Data Protection Regulation

1. INTRODUCTION

1.1 WP5 OBJECTIVES

Within the InterConnect project, WP5 [M7 - M48] is in charge of carrying out the following activities and attaining the following objectives:

- Establish interoperability between project stakeholders (platforms, services, IoT devices) by leveraging the ontologies, standards and designed specifications (T5.1);
- Demonstrate via the interoperability framework and toolbox how several technologies can create a pluggable and transparent approach, while focusing in interfacing functionality-by-design (T5.2);
- Provide security-enabled and a privacy-by-design architecture, by considering a mix of public and private cloud enabled services and legacy systems (T5.3);
- Leverage on the interoperability toolbox to provide P2P marketplace enablers between stakeholders (T5.4);
- Provide the interoperability framework and toolbox for third parties, enabling them to accomplish semantic interoperability of their endpoints;
- Lastly, support instantiation of the interoperability framework within project pilots and continuously manage and monitor the instantiated framework enablers (T5.5).

Moreover, this WP is responsible for designing the set of interoperable endpoints offered by InterConnect, using a scalable, and modular approach. These are 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. Furthermore, WP5 will also focus on the deployment of distributed ledger technologies, tailored for supporting distributed operations as trading and transactions management activities enabling the establishment of P2P marketplaces.

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, which is itself primarily based on the work carried out in WP1, it utilizes and develops the concepts and functions (data models, interfaces, protocols, security and privacy requirements) introduced by the project's secure smart IoT home/building and smart energy reference architecture;
- In parallel with WP3 and WP4 activities, it develops the set of adapted services to be made interoperable and the interfaces towards DSOs;

- 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 interoperable interfaces towards energy markets and especially DSOs while WP5 provides integration with the interoperability framework and services;
- WP5 will provide WP7 pilots with the interoperable digital platforms and supporting services necessary for realizing the project use cases, while the WP7 pilots will provide feedback necessary for updating and maintenance of the interoperability framework;

  WP5 will provide cascade funding projects/partners (WP8) with interoperability toolbox necessary for making their platforms and services interoperable with the interoperability framework and established pilots.

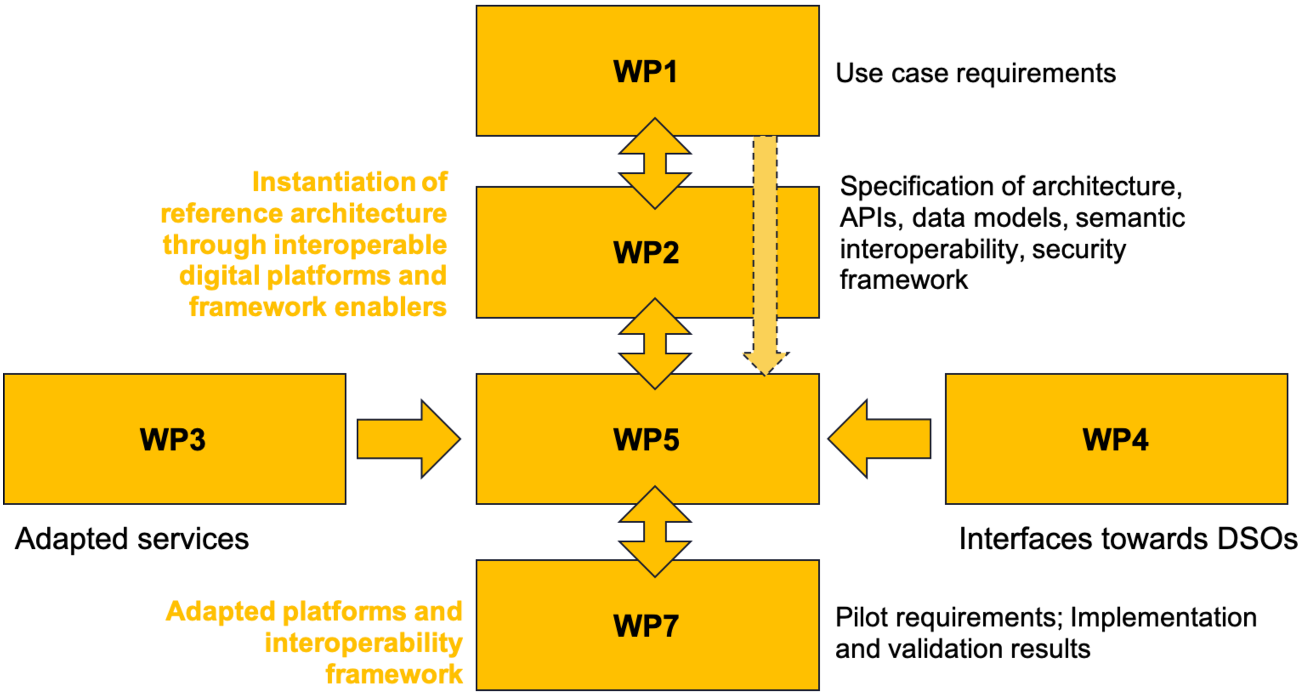
WP1
Use case requirements
Instantiation of reference architecture through interoperable digital platforms and framework enablers
WP2
Specification of architecture, APIs, data models, semantic interoperability, security framework
WP3
WP5
WP4
Adapted services
Interfaces towards DSOs
Adapted platforms and interoperability framework
WP7
Pilot requirements; Implementation and validation results

FIGURE 1 - RELATION OF WP5 TO OTHER WPS.

1.3 D5.2 OBJECTIVES

This deliverable is part of the result of the work carried out by T5.1 - Interoperability framework and service-store architecture and specification [M7 - M12]. Its main objectives can be detailed as follows:

- Overviewing what is the interoperability concept for the interoperability framework;
- Identify and detail what are the required data flows between components, in order to provide semantic interoperability;
- Detail the components that build the interoperability framework (generic adapters, service store, P2P enablers and others)
- Address the need for security measures such as authentication, authorization and the establishment of data boundaries when providing interoperability services;

- Contribute to the specification of the Semantic Interoperability Layer, by identifying the set of connectors and adaptors required for ensuring interoperability on a syntactic and semantic level.

To attain these objectives, the present document introduces:

- An overview for the interoperability concept considered in InterConnect;
- The data flow concept for the interoperability framework and its supporting services;
- The data flow specification for the interoperability framework.

1.4 DOCUMENT STRUCTURE

This document is the deliverable D5.2 Data Flow Management.

This introduction is part **of Chapter 1**. Its followed by the table of common definitions used within this document and other technical and non-technical deliverables published by the InterConnect project.

Chapter 2 – Data Flow Concept, overviews and details what is the concept behind the provision of semantic interoperability, linking it with the architectural constraints provided in deliverable D5.1.

Chapter 3 – Data Flow Specification, addresses the data flow specification between the core components of the interoperability framework.

Chapter 4 – Conclusion, concluding this deliverable.

1.5 GLOSSARY AND TERMINOLOGY

CONCEPT	DEFINITION
InterConnect Framework-related terminology	
IoT platform (provider)	A collection of tools, software and hardware that makes it possible to connect 'things' (i.e. sensors, actuators or other types of physical devices) to the Internet. Also used for managing the connection to the devices as well as the devices themselves.
(The) IC Framework	A collection of tools and enablers that describes and prescribes how to interconnect devices from different vendors and services from different providers, enabling interoperability and the intelligent interaction of many devices and services from different domains (e.g. home automation, energy management, etc.). The IC Framework includes services, like service store for all interoperable services, P2P marketplace enablers, access control mechanisms, generic interoperability adapters, reasoning, and compliance tests.
(An) IC Platform	A digital platform that complies with IC Framework requirements in terms of software and/or hardware that enables the actual interconnection of devices and services. Often implemented on the basis of an IoT platform.

Project Pilot	A collection of tools, software, hardware, building and users that provide a working demonstration one of more aspects of the generic IC Framework in one or more EU countries in terms of platform interconnected devices and services.
Project Use Case	A demonstration of application of the generic IC Framework in terms of using a specific set of services and a specific set of devices, that are interconnected by the platform, in a specific way.
Service-related terminology	
Technical Service Provider	A hardware or software component, possibly representing other components, that is capable of offering certain functionality in the form of an (IC) Service to other components. The other component could be owned by the same actor or by a different actor.
Commercial Service provider	A business actor that provides a service to another actor (e.g. consumer, but also another commercial service provider).
Service user	An entity that uses a service as provided by another entity. This can be from a commercial viewpoint or a more technical one (e.g. 'software using services offered by other technical components'). The context of this term determines the viewpoint.
Customer	A business actor that uses/consumes a service and in return (generally) rewards the (commercial) service provider for the use of that service.
Service Level Agreement (SLA)	Agreement between (commercial) service providers and users/customers
Service Level Management (SLM)	Management of agreements and commitments between (commercial) service providers and users/customers through tracking and documentation of service level delivery and usage.
(IC) Service	The offering of certain functionality from one entity/component to another authorized entity/component (e.g. service or software component) using (standardized) interfaces, compliant to certain IC Framework requirements.
(IC) Regular services	IC Services that are offered via, not by, the IC Framework. Regular services are listed in the IC Service Store.
Service interface	An (technical) interface that exposes the functionalities of an IC Service. Within the IC Framework, this includes a metadata interface for exposing service capabilities
Meta data interface	Part of a (technical) service interface in the IC Framework, that provides functionality for interacting with service at a 'meta' level. This part of the interface can be used for example to interrogate the service about its capabilities and semantical framework. Thus, it can be used for reasoning about using a service.
IC Framework Service	A service that supports offering and using services on an IC platform, as prescribed by the IC framework. Examples are registration and discovery services for interfaces, enabling humans and technical entities to find a particular regular service offered through an IC platform.
Energy service	A service that offers the ability to accomplish an objective (mainly in) in the domain of energy, like balancing demand and supply or the reduction of energy usage. This is a special category of services within the IC Framework, as energy services (often) require the coordination of tasks across different Smart Homes and Smart Buildings across the Smart Grid and thus requires multiple levels and domains of control to be interconnected.
Non-energy service	Non-energy service are services that do not relate to energy and/or do not enable clients to accomplish and energy objective (as a main objective). Examples of non-energy services are services that have as objective comfort, well-being, entertainment or safety of their users. Non-energy services can be used by and/or 'become part of' an Energy service. For example, a non-energy service that sends events when a door remains open, can be used by an Energy service to reduce loss of heat in a house by closing doors.
Technical service implementation related terminology	
Software as a Service (SaaS)	A software licensing and delivery model in which software is licensed on a subscription basis and is hosted (de)centrally. It is sometimes referred to as "on-demand software". SaaS applications are also

	known as Web-based software, on-demand software and hosted software. The term "software as a Service" (SaaS) is considered to be part of the nomenclature of cloud computing.
Local / Remote Services	Software services can be either implemented as code that is run at 'remote' server (i.e. on the cloud), or on a 'local' server, i.e. as code that runs on a digital platform that is in a Smart Building or Smart Home.
IC Service run-time platform	Code that is hosted on a digital platform and acts as an abstraction layer for the underlying software platform (e.g. specific operating systems). The digital platform hosting the IC service run-time platform can be any kind of digital platform, ranging from resource constrained embedded systems up to (virtual) cloud servers. IC services compliant with the IC service run-time platform are called IC² service and digital platform agnostic as they interface with IC service run-time abstraction layer and not directly with the underlying software platform.
(IC) Native Service	A service implemented as software/code that runs on a specific vendor's digital platform, making use of specific functions and characteristics of this specific platform.
(IC) IC² Service	A service implemented as software/code that runs on top of the IC service run-time platform.
Semantic and Syntactic Interoperability-related terminology	
Semantics	Semantics is the study of meaning, i.e., the meaning of the data being exchanged via the IC Framework
Semantic Interoperability	Semantic Interoperability concerns the exchange of meaningful information on the basis of agreed, formalized and explicit semantics
(IC) Semantic Interoperability Layer	A logical concept within the IC Framework that enables semantic interoperability. The semantic interoperability layer comprises ontologies, interoperability adapters and smart connectors with supporting orchestration enablers.
Ontology	The formal specification of a conceptualization, used to explicit capture the semantics of a certain domain of discourse. In the IC Framework, ontologies like SAREF are used to capture the agreed, formalized and explicit semantics for the exchange of meaningful information via the semantic interoperability layer.
IoT Platform specific Information Model	In a specific IoT platform, it is a representation of concepts and the relationships, constraints, rules, and operations to specify data semantics for a chosen domain of discourse, related to a specific IoT platform.
(IC) Sarefized Services	A Software Service whose capabilities and data for the Service Interface are expressed using the SAREF ontologies. (IC) Sarefized Services are automatically recognized by the IC Semantic Interoperability Layer. The capabilities of an (IC) Sarefized Service automatically become available to other Sarefized Services/Devices.
(TNO's) Knowledge Engine	An open-source, ontology-agnostic software component that is being developed by TNO in cooperation with VU Amsterdam. The Knowledge Engine helps improve interoperability by making data exchange more dynamic and smarter through orchestration and semantic reasoning. It creates a new way for software and devices to communicate with each other.
Knowledge Directory	A central component that registers the knowledge offered and requested by Smart Connectors. It does not perform any reasoning.
IC (Smart) connectors	Generic software responsible for orchestration and reasoning. The Smart Connectors are peers, that can communicate directly with each other through SPARQL+. Based on the information in the Knowledge Directory, each Smart Connector can perform orchestration and reasoning for itself. Smart Connectors configured to use the same Knowledge Directory can communicate with each other through SPARQL+.
IC adapters	The Interoperability Framework provides a set of adapters to allow vendors that are already compliant with industry standards to quickly connect their device/service to the Interoperability Framework. Ideally, for each industry standard (i.e., SPINE, WoT, modBUS, S2) an adapter would be available. IC adapter includes IC connector and also the underlying mapping of legacy data models and interfacing functionalities onto the InterConnect unifying protocol (SPARQL+) and SAREF based data model.
Knowledge IO	A description of a type of interaction that a Knowledge Base supports. There are five types of interactions, each with a Graph Pattern associated with it. The Function KnowledgeIO has two (one for

	input, one for output). A Knowledge Base typically has multiple KnowledgeIO's of different types. KnowledgeIO's are registered in the Knowledge Directory.
SPARQL+	Unifying interfacing protocol for the InterConnect semantic interoperability layer. It comprises the SPARQL standard and additional interfacing functionalities required for realization of the project use cases ("+" in the name).
IC Interoperability Framework-related terminology	
(IC) Service store	Complete catalogue of all interoperable services from energy and non-energy domains. The service store is implemented as a web application providing frontend interface for onboarding new interoperable services and browsing existing (already onboarded services) by category and other metadata parameters. The service store is part of the interoperability framework and can be utilized by local reasoners to find appropriate remote services (running on 3rd party platforms) needed for completing a task at hand. Service store enables users or local reasoners to find interoperable services of interest and provides them with information on how to access the services running on their hosting digital platforms.
(IC) Deployment Orchestrator	This is integral part of the service store responsible for facilitating instantiation of interoperable services packaged as containers for specific runtime environments including the service store sandbox.
P2P marketplace enablers	Set of enablers for P2P marketplaces include: Hyperledger Fabric configuration as blockchain basis for trusted data access and transaction management; set of smart contract templates representing supported transactions, reports and audits; white labelled web application utilizing blockchain network through integrated smart contract interfaces. These enablers can be configured and deployed for specific use case, on the level of a pilot or on the level of the whole project.
IC security and data protection framework	Set of best practices for ensuring data and privacy protection in integration/interoperability scenarios between two or more stakeholders with digital platforms, services, end users and databases. On the level of the project, a specific access control mechanism will be implemented with user/service/platform authentication and authorization procedures directly integrated with semantic interoperability layer (discovery and reasoning).
Interoperability compliance certification	Set of automated tests of achieved interoperability minimum defined for each service and platform category. The tests will include dummy data exchanges to showcase that defined data models are properly parsed and understood and services are capable of exchanging information through unifying communication layer/protocol. The interoperability compliance test will be part of the service onboarding process in the IC service store. After successful compliance test, a certification of interoperability compliance will be issued and written in immutable record of all interoperable endpoints based on Hyperledger Fabric blockchain established on the level of the IC project.

2. DATA FLOW CONCEPT

The provision of interoperability in InterConnect project is founded on the concept of providing the digital means for contained, high TRL digital platforms and services to expose their capabilities beyond the ecosystems of their manufacturers. This is possible via the adoption of two requirements, namely: (*a*) the adoption of a common technology serving as an ICT gateway for data and metadata exchange and (*b*) elevating the interoperability level to consider semantic reasoning.

From the perspective of WP5, digital platforms are software packages owned and maintained by distinct vendors that offer an array of ICT services, that in an isolated or in an agglomerated way, exposes one specific capability. Services themselves can also be depicted in a standalone approach, meaning that they do not encompass a digital platform. Often, a digital platform as a collection of services exposes (either internally for a specific ecosystem, or for external parties) a set of application programming interfaces (*i.e.,* API) that allow it to offer a low level of interoperability. This means that when a foreign ICT entity respects the convention imposed by one specific API (*i.e.,* protocol, data modelling), data exchange is possible. This is currently a very common scenario and it is depicted as *syntactic interoperability*. The concept implies that, beyond the fact that data exchange is possible, there is no interpretation of the underlying meaning of the exchanged data. Data is traded between entities in an oblivious way by ICT devices, being this a result from a strict mapping of instructions produced by a human being, the actor responsible for the interpretation and adjustment of the underlying meaning of data and triggered actions.

InterConnect adheres to semantic reasoning capabilities and places them at the heart of the components responsible for providing interoperability.

The current chapter starts with addressing the concepts regarding data flow and its management between the components and services within the interoperability framework. The interoperability framework is built from a series of internal components that will facilitate data exchange and discoverability of new software services, features and capabilities required within the ecosystem.

The main consideration regarding the conceptual data flow is that any digital platform or device could enrol into the software services, provided that they adopt one of the available adapters or implement one of their own. The generic adapters provide a gate towards the basic functionalities within the interoperability framework, providing means for data exchange and data interpretation.

This section addresses the overall interoperability concept, highlighting the role for each key functionality, namely: the semantic reasoning and data exchange and the generic adapter. Moreover, it addresses the need and means for operational data exchange and required data boundaries for data protection, in line with InterConnect’s Data Management Plan [7].

2.1 SEMANTIC INTEROPERABILITY OVERVIEW

Interconnect's semantic interoperability layer is envisioned as a distributed network of interoperability adapters and connectors hosted on digital platforms provided by project partners and other solution integrators. The IC interoperability framework services will also feature semantic interoperability adapters and connectors. This will create a semantic/knowledge layer where all interoperable services and endpoints can discover each other and perform reasoning to create new connections and data exchange paths. Note – in the figures in this and subsequent sections a colour coding will be used to depict InterConnect interoperability framework/layer with orange colour. When presenting an interoperability adapter, the orange colour depicts the unified interoperability layer and the other colour represents existing/legacy interface implementation.

The interoperability adapters build one of the focal agents within InterConnect's semantic interoperability vision. They condense and act as the bridge for digital platforms and standalone services to reach out to the available ecosystem of platforms and services. The overall positioning for the interoperability adapters within a generic environment is depicted in Figure 2. Adapters will integrate with digital platforms and/or software services, providing a gateway towards the interoperability layer and overall framework with its own services and enablers. Adapters will be available in a series of pre-configurations, respecting and providing means for easy integration, making available versions for several software platform stacks and allowing for several transport protocols to be adopted.

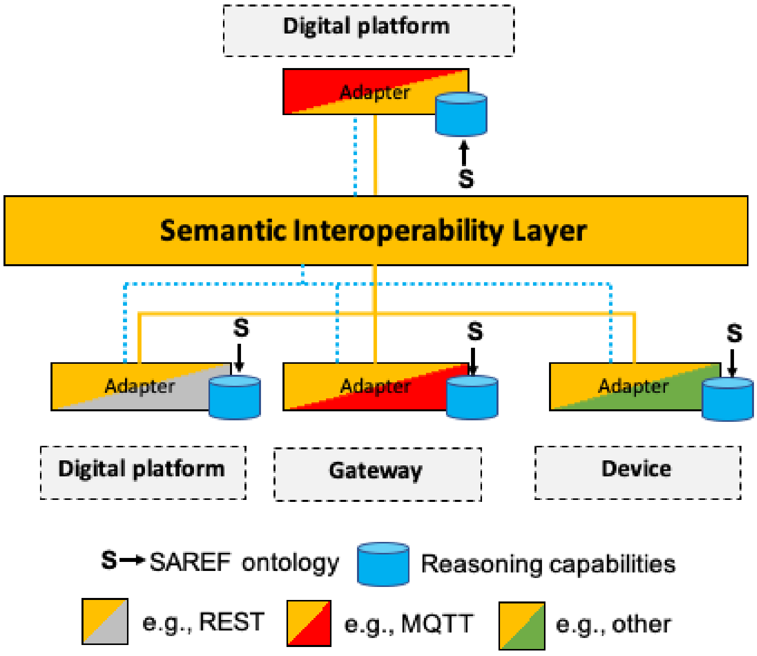

FIGURE 2 - INTERCONNECT ADAPTER - HIGH LEVEL OVERVIEW OF THE CONCEPT

The interoperability adapters will provide the gateway towards semantic reasoning and discovery of foreign capabilities and will also allow operational data forwarding in-between adapters. One given digital platform may comprehend several adapters, depending on the nature of the platform, offered services and capabilities. For instance, one given digital platform may integrate an adapter to export REST based services, and, at the same time, an adapter to export MQTT capabilities. Multiple adapters of the same type may also coexist if that is the intention of the adopting platform or service.

In essence, an adapter builds a proxy-like entity that allows transfer protocol translation and a gateway to reach-out to reasoning capabilities. The goal of WP5 is to develop a set of generic adapters addressing the needs of the catalogued digital platforms and their services (see D5.1 for details). Figure 3 depicts the structure for one generic adapter. As a gateway for digital platform enrolment, generic adapters will be available in a series of distinct implementations to be chosen by adopting digital platforms and services. The pool of generic adapters will make available options according to the underlying software framework (*i.e.,* Java, Python, etc) and will have a modular construction to aid the link with transport and common interfaces (*i.e.,* REST, MQTT, Web Sockets, etc).

The generic adapter construction allows for flexible adoption, offering digital platform owners the capability to select the generic adapter that better fits the software framework of their platforms, and focus only on the required integration to expose software services and software service capabilities. The adapter is logically split into two parts, the northbound part, that is responsible to establish the required communication and interaction with the Semantic Interoperability layer common services, and, the southbound, that holds a configurable part that will require integration with the underlying digital platform or software service. The adapter is customised via a configuration file, that will allow to create dynamic mappings to the underlying system, but respecting the active ontology (*e.g.,* SAREF ontology).

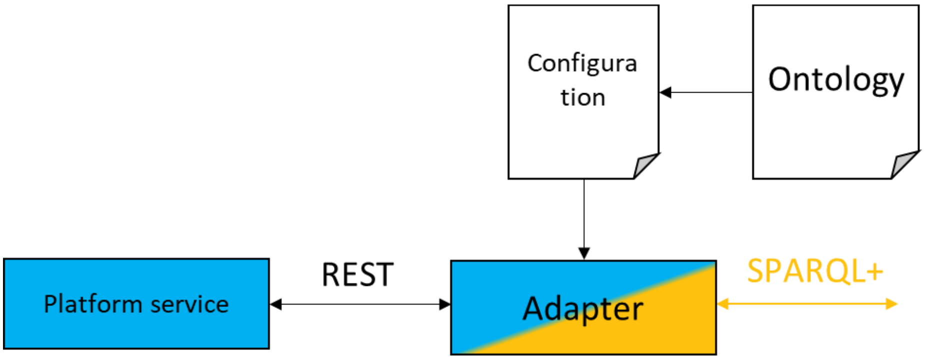

FIGURE 3 - GENERIC INTEROPERABILITY ADAPTER CONFIGURATION.

This implies that, on one hand, the southbound component for the adapter, the transport protocol in use will be the one characterising the adapter (*i.e.,* one of the available generic adapter is selected, for instance REST, MQTT, SPINE, etc.). On the other hand, the northbound communication and encoding will be achieved via SPARQL+, the improved version of SPARQL for the need of this project (to be defined in D2.1 at 31.12.2020.). This common language between adapters will allow their interchangeability without compromising interoperability and compliance. Moreover, the use of a base protocol (*i.e.,* SPARQL) that can support multiple ontologies makes the generic adapter independent of the chosen ontology (*i.e.,* SAREF family for the present case). This means that if by any reason there is the intention to change the current ontology (*e.g.,* change in versioning, or complete change for other ontology), the SPARQL+ protocol, along with the northbound part of the adapters will not require new development cycles. Nevertheless, for the highlighted case, new bindings to the southbound part of the generic adapter (*i.e.,* mapping towards a digital platform) will have to be adapted, via the provision of a new configuration profile. The set of generic adapters provided by WP5 will be utilized by platform owners to make their digital platforms and corresponding services interoperable. This instantiation will take place in WP5 for digital platforms in whole and WP3 for energy and non-energy services.

Figure 4 shows a typical pilot ecosystem comprising: two different digital platforms, each with its own set of services (concept of services is presented in the next subsection), managed devices and interfaces; a service running on a platform that might not be part of the InterConnect digital platform catalogue (defined in D5.1 [6]); application (i.e. web or mobile) developed for the purpose of a project use case and utilizing the interoperable services (not necessarily providing additional services); IC interoperability framework where specific focus is put onto the IC semantic interoperability layer. The semantic interoperability layer is represented with InterConnect interoperability adapters instantiated for each participating platform/service. Between the adapters the semantic discovery and operational data exchange (following the unifying interoperable protocol SPARQL+ and SAREF based data models) is enabled. The orchestration and reasoning (elaborated in the next subsection) are presented as a centralized component, but it can also be distributed among the InterConnect adapters. Concept of a semantic connector is presented as component enabling semantic reasoning for endpoints (i.e. services and apps) which already expose SPARQL+ interfaces and utilize SAREF based data models developed within InterConnect WP2.

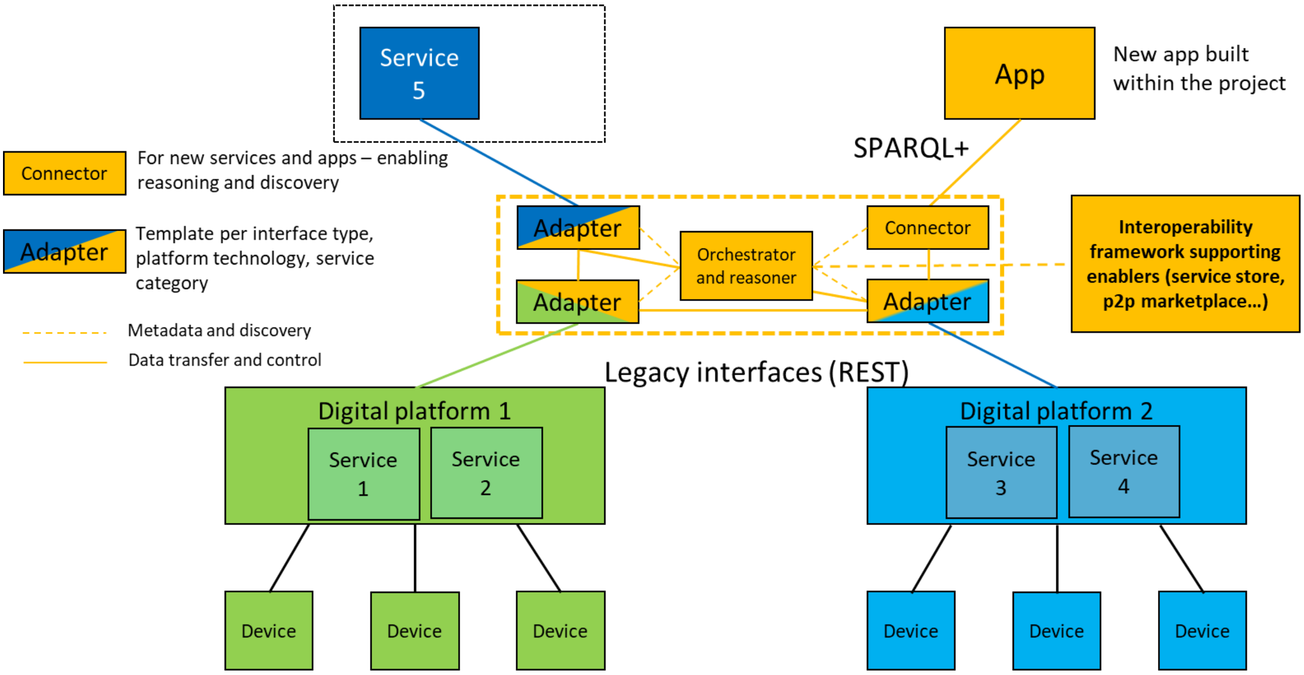
Service 5
App
New app built within the project
SPARQL+
Connector
For new services and apps – enabling reasoning and discovery
Adapter
Template per interface type, platform technology, service category
Metadata and discovery
Data transfer and control
Adapter
Connector
Orchestrator and reasoner
Adapter
Adapter
Interoperability framework supporting enablers (service store, p2p marketplace...)
Legacy interfaces (REST)
Digital platform 1
Service 1
Service 2
Digital platform 2
Service 3
Service 4
Device
Device
Device
Device
Device
Device

FIGURE 4 - SEMANTIC INTEROPERABILITY LAYER IN CONTEXT OF A TYPICAL PILOT ARCHITECTURE

The InterConnect semantic interoperability adapters are not hosted on some dedicated central digital platform provided by the project. They are hosted/instantiated on the digital platforms provided by the project partners, and which are running the services which need to be adapted and made interoperable. The IC interoperability adapters can be instantiated on a level of a service (each service with its own adapter), or on a level of the whole digital platform running multiple services. Approach on how to instantiate the adapters will be decided by the platform and service operators.

The deployment of a series of IC adapters builds a network of distributed entities, each one of them with relative autonomy to take decisions and assert actions. The base technology for an IC adapter is a smart connector (section 2.2.1) with reasoning capabilities, that together with other smart connectors, forms a knowledge engine. Each IC adapter enables software service specific features or properties to make use of the semantic reasoning capabilities, via the registration of its functions. When registering specific capabilities, software services via the IC adapters have to consider one reference ontology (*e.g.,* SAREF family for the present case), in order to connect and allow triggering intelligent relationships between software service properties and required data translations, via the smart connectors.

The present approach allows for a flexible system, as the underlying dissemination of capabilities and push/pull software service requests are mapped to a changeable representation of the environment (*i.e.,* one ontology) and not to a static realization of one concrete information domain. This allows for increased maintainability, but also for naturally occurring revisions of that particular representation/ontology over time.

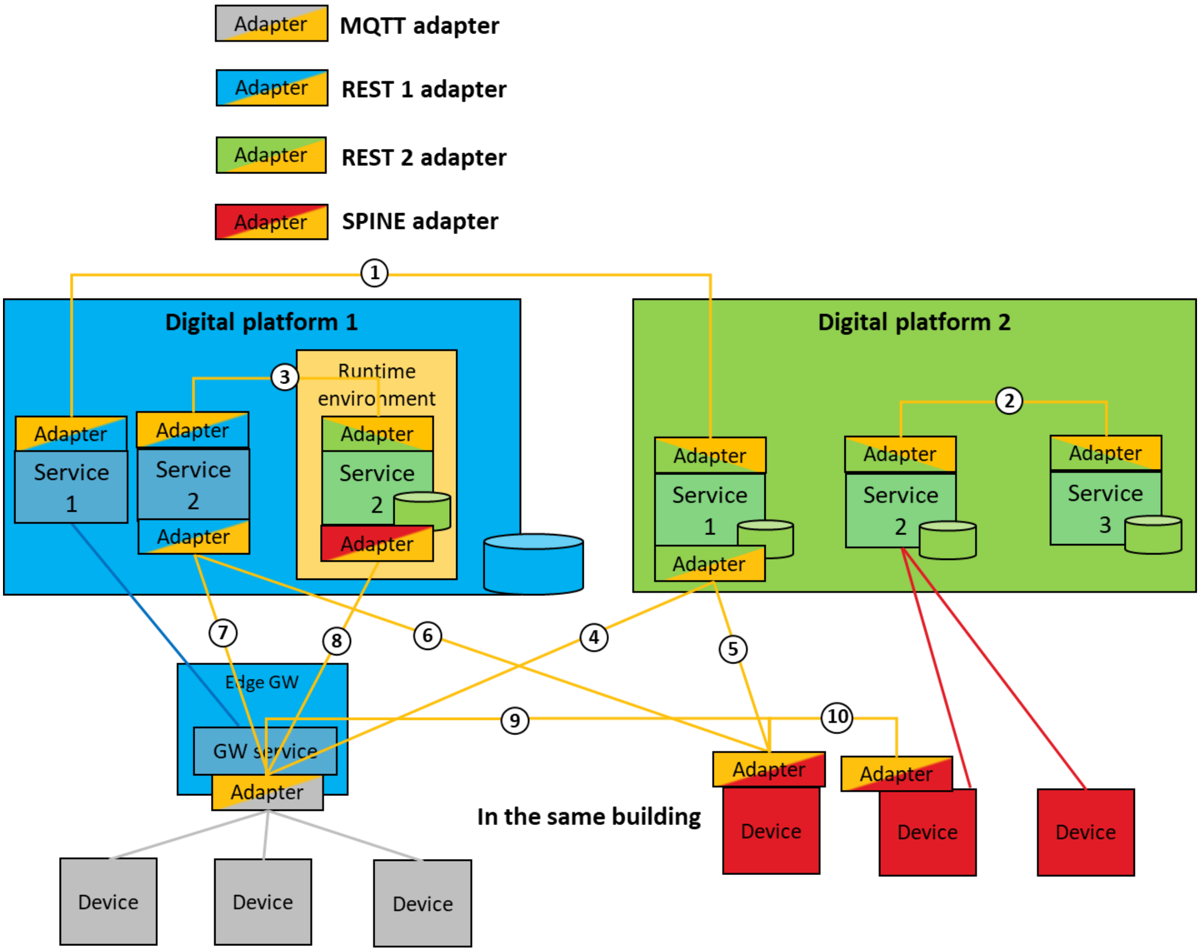

FIGURE 5 - SEMANTIC INTEROPERABILITY LAYER AND ENABLED DISCOVERY AND DATA EXCHANGE INTERLINKS

Figure 5 showcases two digital platform with running services and IC semantic interoperability adapters deployed at different system levels and for different interfacing technologies. It does not directly depict a particular demonstrator or business case but showcases a diverse configuration where multiple generic adapters are deployed and attached to digital platforms or devices to sponsor interoperability between services. Consider that the two platforms manage devices and edge elements deployed at the same building. Four different IC adapters are presented: REST adapter configured for two different REST interface implementations, adapter for MQTT and adapter for SPINE protocol (i.e. SPINE over MQTT). Each service in the figure has a corresponding adapter for northbound interface (for interoperating with other services) and some services have adapters for southbound interfaces (for interoperating with edge services and interoperable devices). With deployment of the InterConnect semantic interoperability layer the two digital platforms can exchange data end execute control procedures between their managed resource and service pools. Without this semantic interoperability layer, the two platforms and their services could not manage and access each other’s resources without custom integration APIs and negotiating data models. Even if this kind of integration would be straightforward, any future updates to the smart building ecosystem (replacing managed devices, service updates or inclusion of additional digital platforms and services) would require manual updates to the established interoperability procedures.

By applying the IC interoperability enablers, the depicted smart building ecosystem can grow with new platforms, devices and services running interoperability adapters. Also, the established smart building system can access other IC interoperability framework services like service store and interoperable services provided within. Figure 5 shows interoperability paths opened with the instantiated IC semantic interoperability layer:

1. Interoperability of services running in different platforms;
2. Interoperability of services within the same platform (i.e. this was not implemented during the platform/service development);
3. Interoperability of services within the same platform, but one of them is instantiated as a container in dedicated runtime environment (this option will be further elaborated in the subsection on IC Service Store – see D5.1 [6] for more details);
4. Services of one platform with services running on edge gateway of the other platform;
5. Service in platform 2 with device operated by platform 2, but previously direct service access was not supported;
6. Service in platform 1 with device operated by platform 2;
7. Service from platform 1 to service at edge gateway of platform 1 – possibility that this communication path was not implemented originally;
8. Service running in container runtime environment to the edge gateway service of the hosting platform;
9. Device to edge gateway service;
10. Device to device interoperable link.

Not all these possible interoperability links will be explored in the InterConnect project pilots. Nevertheless, the IC semantic interoperability layers will support all of them for future use, or cascade funding projects.

Next, details about semantic reasoning are presented followed by overview of the two enabling technologies for realization of the InterConnect semantic interoperability layer and interoperability adapters are presented.

2.2 SEMANTIC REASONING AND DATA EXCHANGE

Beyond the ability of two or more systems to exchange information with correct syntax (*i.e.,* grammatically correct), semantic understanding concerns the (automatic) correct interpretation of the meaning of information. To achieve semantic interoperability, at least two entities must refer to a common information exchange reference model. This reference model must define the meaning of the exchanged information (the words) in detail. This is the only way to ensure that the communicating systems will correctly interpret the information and commands contained in the transferred data and will correctly act or react.

Reference ontologies, such as SAREF, can be used to represent the common reference model. They may also model constraints about the information concepts by specifying assertions and inferences that can be used in reasoning mechanisms (*e.g., if this, then that*). This allows resolving interpretation conflicts in situations where two differently named classes in different models mean the same or when a class is a subset or superset of another class.

2.2.1 KNOWLEDGE ENGINE

The *Knowledge Engine* (KE) provides semantic interoperability by means of two features: **translation** and **discovery**. Both these features require a common ontology, such as SAREF. Figure 6 depicts the positioning for the Knowledge Engine rationale.

The knowledge engine is able to InterConnect different *Knowledge Bases* (KB), which are depicted in the Figure 6 as cylinders. Knowledge bases can be anything, from devices and services to algorithms, apps, machine learning models or platforms from different vendors, representing one particular domain of knowledge.

A *Knowledge Base* (KB) loosely couples three main principles, namely:

- **metadata**, containing data from devices, services that characterise their classification in *type* and capabilities;
- **data,** containing operational and business related information, together with queries and replies towards third party data holders;
- **reasoner**, enabling devices and services to draw new relationships between concepts.

By providing a loose coupling between concepts, each domain represented by each one of the knowledge bases is able to tolerate a dynamic ecosystem, where the number of knowledge bases can freely change or tolerate some level of device and service churn. The reasoner provides then the coupling between knowledge bases and maps such relationships when deemed necessary. Relationships are represented by Knowledge I/O (input/output) representations of data being added or queried from the graph representation of a domain, where several knowledge I/O movements can be associated with a single knowledge base

To become semantically interoperable with other knowledge bases, each knowledge base is provided with a specific component, the **Smart Connector (SC)**, which realizes the translation mechanism to/from a common ontology (*e.g.,* SAREF).

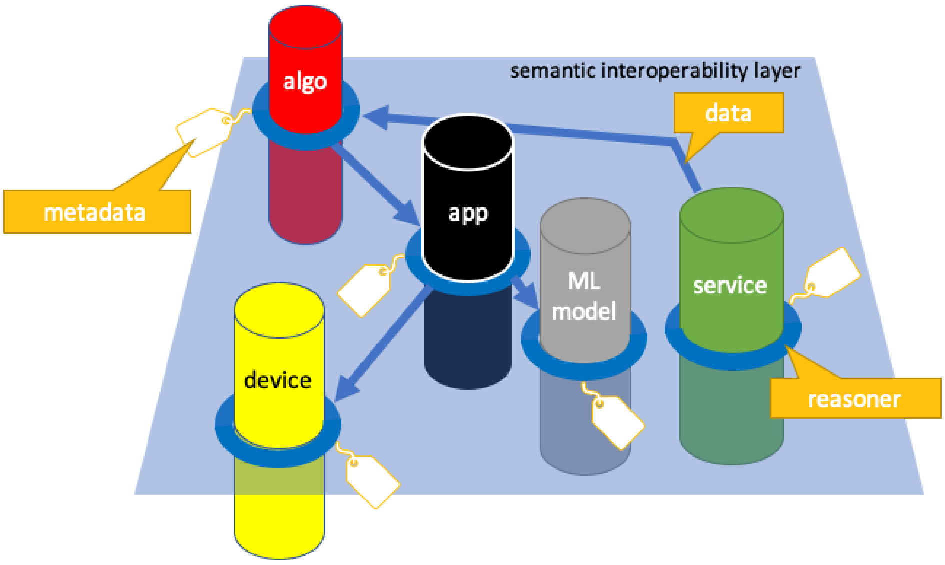

FIGURE 6 - KNOWLEDGE ENGINE REASONING RATIONALE.

As a requirement, smart connectors must know both SAREF and the specific language that needs to be translated to SAREF. Each smart connector registers itself within a *Knowledge Directory* (KD) (not shown in the figure), that acts as local data repository, with a description of the capabilities that it wants to make available to other smart connectors. This description is defined as a graph pattern in SPARQL that refers to concepts in SAREF. These patterns are used for the discovery of knowledge by other smart connectors. When a smart connector (and its corresponding knowledge base) is no longer available, or when a new smart connector becomes available, the Knowledge Directory is dynamically updated. With this up-to-date information, the knowledge exchange among knowledge bases can take place. This is shown by the arrows in Figure 6. Knowledge is exchanged using a combination of SPARQL and RDF messages that refer to SAREF concepts.

The adoption of the knowledge engine as one of the key technologies behind the interoperability layer adopts the following principles/guidelines:

#	Guideline	Details / Data Movement
1	All communication that should be interoperable, should go via the Knowledge Engine.	Data exchange should go through the KE for fully interoperable connections. Interoperability level can be configured.
2	Each Smart Connector is configured with the **capabilities** of its Knowledge Base	Multiple knowledge bases could represent several services/capabilities per smart connector.
3	**Knowledge** is described via Graph Patterns.	Graphs allow the construction of relationships in terms of data triplets.

4	At start-up the Smart Connector **registers** itself with the Knowledge Directory	The service capabilities are registered by pushing details to the knowledge directory.
5	The Smart Connector **periodically** polls the Knowledge Directory for updates on the available Smart Connectors and their capabilities.	When new connectors register or changes service details, updates are periodically fed to other knowledge directories.
6	Each **KnowledgeIO** describes a single capability of a Knowledge Base and translated into **rule** for the **reasoner.**	Corresponds to an input and output graph pattern, where each pattern may contain one or more triples.
7	The rule reasoner orchestrates the **Data Exchange**	After relationships are described, data is forwarded and translated if required.
8	Smart Connectors do not store data, they are only a proxy. All storage should happen within the Knowledge Bases.	Smart connectors only proxy data between entities. If data storage is required, that is dealt by each knowledge base (per domain).
9	Whenever a Knowledge Base A requires data, it asks Smart Connector A.	Data is pulled if deemed necessary.
10	Whenever a Knowledge Base A has data, it sends it to Smart Connector A.	Upon request the smart connector pushes missing data.

TABLE 1 - KNOWLEDGE ENGINE DATA MOVEMENT STEPS.

The adoption of the knowledge engine, together with the reasoning capabilities to be described afterwards, allow the exchange of operational data between smart connectors, ultimately allowing operational data forward between digital platforms and services that attach one of InterConnect's generic adapters. The actual data exchange is made possible via two possible paths, namely: a publish/subscribe mechanism that establishes a set of data queues for data delivery; and, the SPARQL querying mechanism.

2.2.2 REASONING

The reasoning mechanism is bundled within the smart connector and allows for new relationships to be drawn at each time from the existing graph pattern. Graph patterns allow for relationship representation, according to a given ontology, for instance, SAREF.

In order for the reasoning mechanism, several steps are required, namely:

- The Smart Connector collects the KnowlegeIOs of all other Smart Connectors (via the Knowledge Directory)
- The Smart Connector compares its own KnowledgeIOs with those of the other Smart Connectors.
- If applicable, the Smart Connector translates the KnowledgeIOs into rules for the Reasoner.
- Rules act as proxies as during application of the rules, other Smart Connectors are consulted for the actual knowledge.

- If the Smart Connector receives a **new question/query** for certain knowledge, its reasoner will use the rules to answer the question.
- If the Smart Connector receives **new knowledge,** its reasoner will use the rules to forward this knowledge to subscribers.

2.2.3 REALIZATION OF INTERCONNECT INTEROPERABILITY ADAPTERS WITH KNOWLEDGE ENGINE

A simplified version for the internals of an InterConnect generic adapter, showcasing the interface adapter and the smart connector is depicted in Figure 7. The Adapter is responsible to create bindings between the northbound and southbound sides of the generic adapter, linking between the native interfaces of the underlying digital platform or service and the SPARQL+ interface towards the interoperability layer.

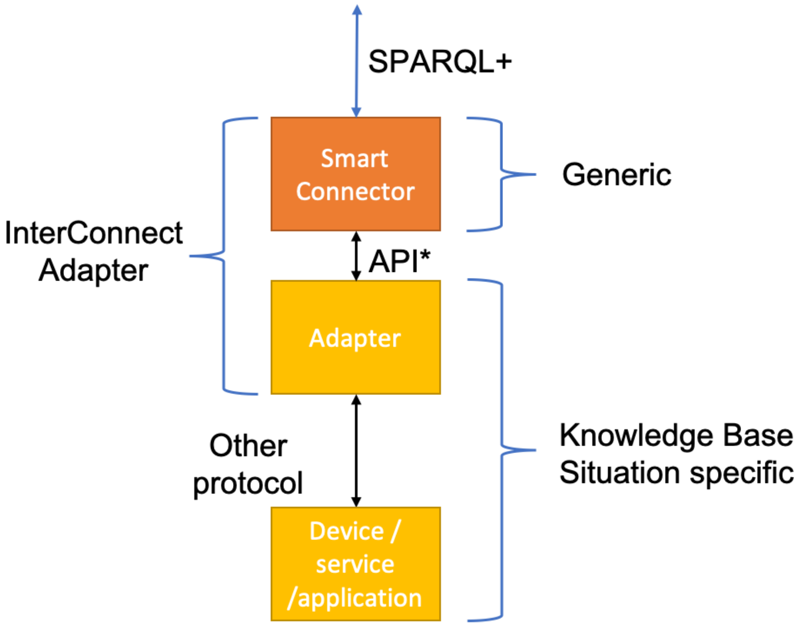

FIGURE 7 - INTERCONNECT GENERIC ADAPTER.

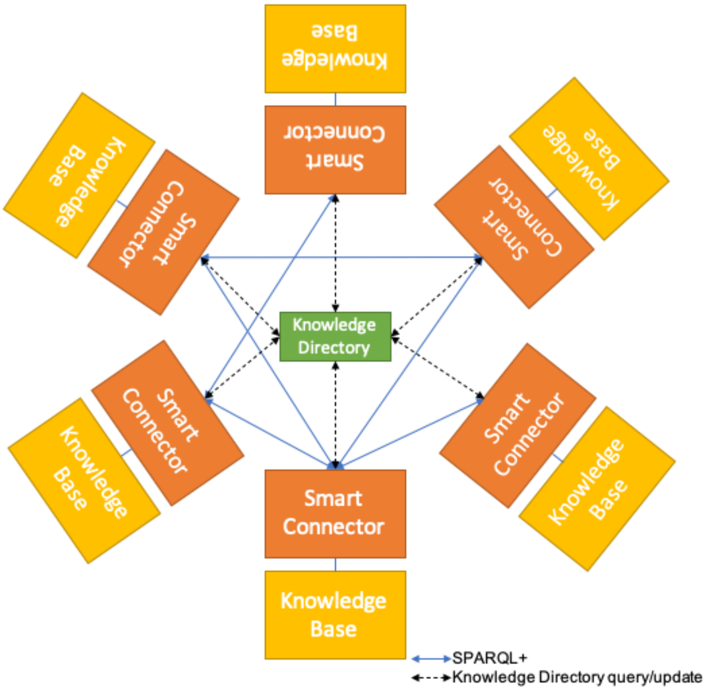

FIGURE 8 - INTERCONNECT GENERIC ADAPTER MESH VIEW AND KNOWLEDGE ENGINE.

The smart connector is responsible to implement the data model translations that are maybe required to adjust to the requirements imposed by the use of a given ontology.

Individual InterConnect generic adapters have the capability to be arranged in a mesh topology, allowing for a distributed setup, without requiring a generalized central approach for data exchange. Figure 8 depicts a distributed setup for several generic adapters together with the knowledge engine capabilities. Each generic adapter will embed the capabilities of the knowledge engine for upstreaming data via SPARQL+ and knowledge/discovery data via the smart connector in each generic adapter. This approach allows generic adapters to become peers that can communicate directly with each other via SPARQL+. Based on the information in the knowledge directory, each generic adapter can perform orchestration and reasoning for itself.

2.2.4 SAREF-COMPLIANT RULE-BASED REASONER

A semantic reasoner [Gyrard et al.] for IoT (Sensor-based Linked Open Rules - S-LOR) has been introduced and explained in D2.1 - Semantic Interoperability architecture and D5.1 - InterConnect Interoperability Framework Architecture. The S-LOR semantic reasoner is a rule-based reasoner compliant with ontologies (e.g., the M3 ontology that extends the W3 SSN ontology V1). Figure 9 below explains each step hereafter that illustrates the data workflow.

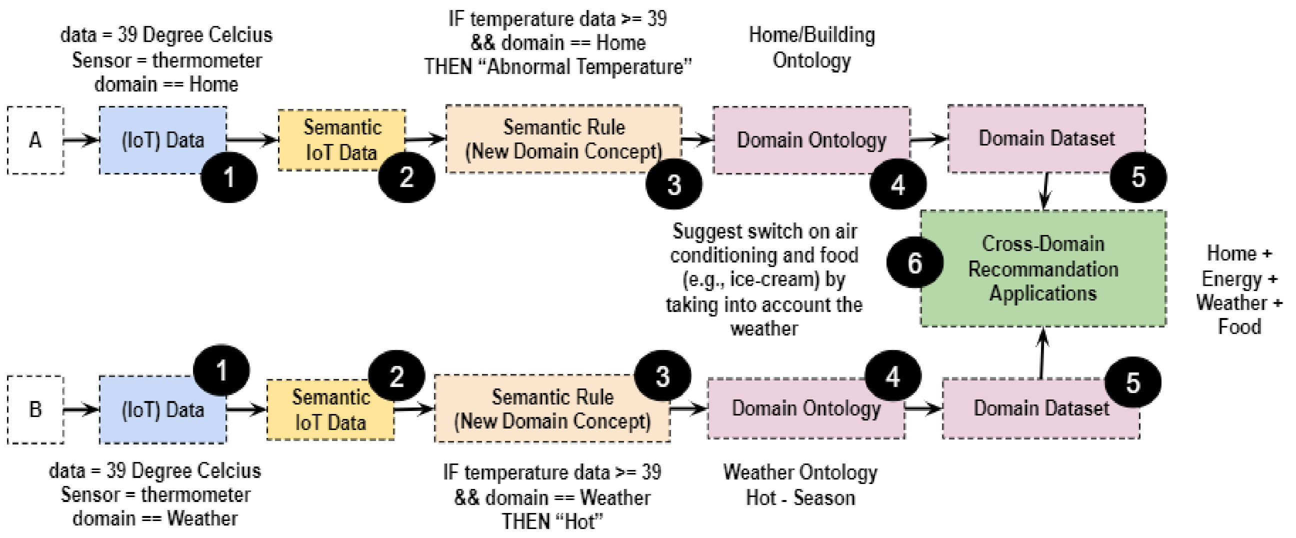

FIGURE 9 - THE IOT KNOWLEDGE-BASED CROSS-DOMAIN RULE-BASED ENGINE AND DATA WORKFLOW

Table 2 explains each step that illustrates the data workflow [1] [2] .

Steps	Description
Step 1	The raw measurements generated by the sensors are transformed into metadata with additional attributes: (1) Unit of Measurement, (2) Timestamp, (3) Software Version, (4) Name, (5) Type, and (6) Domain of Operation. Ideally, it could support heterogeneous data formats (e.g., JSON, XML), but requires wrappers to unify sensor metadata descriptions.
Step 2	The framework encodes the metadata using Sensor Markup Language (SenML) to unify sensor metadata before converting into RDF compliant with ontologies (e.g., M3, SAREF ontologies), a key step to later execute the rule-based reasoner.
Step 3	Semantic reasoning drives higher level abstractions as new domain concepts. In the health domain, the reasoning engine explicitly deduces the "flu" concept; in the weather domain, the "hot" concept.
Step 4	The respective domain ontologies are used to classify these new concepts; "flu" as a disease and "hot" as a seasonal condition.
Step 5	The respective domain datasets are used to link data (e.g., food with diseases, menu with season).
Step 6	The concepts, rules, and datasets of the two domains, are combined and cross-domain semantic reasoning takes place. In this example, the cross-domain reasoning produces suggestions for recipes appropriate for a given state of health and the prevailing weather conditions. The recommendations can be acted upon both by end-users and intelligent machines.

TABLE 2 - STEP DESCRIPTIONS OF THE IOT KNOWLEDGE-BASED CROSS-DOMAIN RULE-BASED REASONER.

How this specific technology is going to be used for realization of the InterConnect semantic interoperability layer and adapters is work in progress within WP2.

2.3 DATA BOUNDARIES

The interoperability framework and the generic adapters supporting it will become gateways for data exchange, both at the level of metadata for service registration and service discovery, but also to exchange business data between sets of generic adapters that may be engaged. This scenario requires **data privacy access control** to be one critical feature surrounding the construction of a generic adapter.

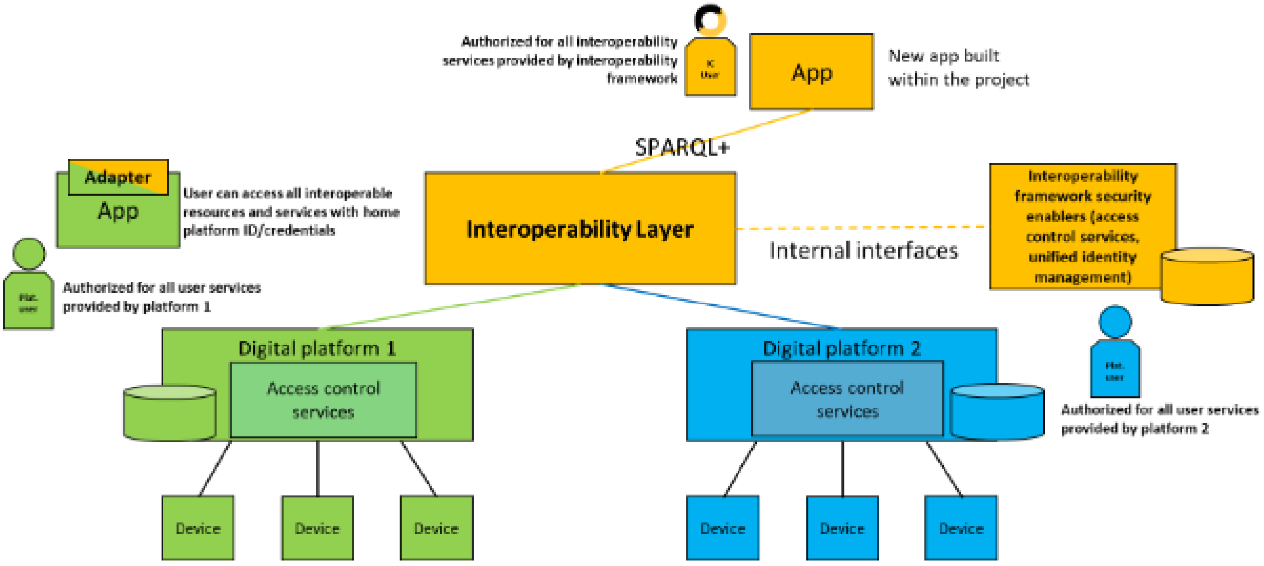
Authorized for all interoperability services provided by interoperability framework
IC User
App
New app built within the project
SPARQL+
Adapter
App
User can access all interoperable resources and services with home platform ID/credentials
Interoperability Layer
Internal interfaces
Interoperability framework security enablers (access control services, unified identity management)
Plat. user
Authorized for all user services provided by platform 1
Digital platform 1
Access control services
Digital platform 2
Access control services
Plat. user
Authorized for all user services provided by platform 2
Device
Device
Device
Device
Device
Device

FIGURE 10 - INTEROPERABILITY LAYER ACCESS CONTROL AND DATA BOUNDARIES.

One key principle is the concept for the legacy platform user and the IC (InterConnect) user. The first represents an internal user of a given digital platform or standalone service. This depicts the regular scenario where there is a set of credentials and system that allows or not one given user to perform tasks, that is often guided by the establishment of permissions and access control rules. This is often a closed mechanism, as it establishes (together with other systems such as firewall mechanism) the barrier to service provision. When integrating with an external system (either a machine-to-machine or human to machine interaction), new credentials are generated and authorized within that particular digital platform or service. As a standard feature around digital platforms, this requires custom authorization and control, delaying service adoption.

To allow interoperable services to be easily pluggable and enable a true “plug-and-play” ecosystem, there is the need to establish an IC user, that can easily be used to provide authorization roaming throughout the set of interoperable systems and platforms. Figure 10 depicts this scenario. Moreover, the adoption of such a capability will allow an autonomous and hassle-free experiencing for discovery and integration of a given software service.

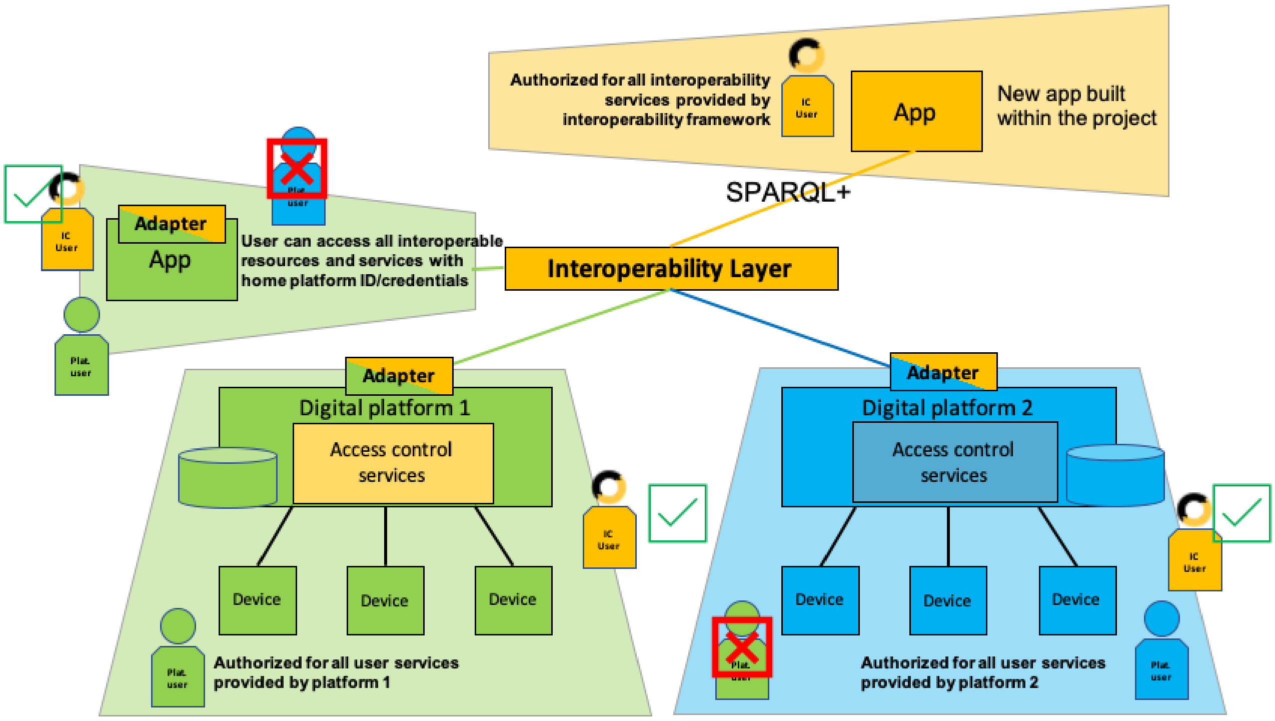
Authorized for all interoperability services provided by interoperability framework
IC User
App
New app built within the project
SPARQL+
IC User
Adapter
App
Plat. user
User can access all interoperable resources and services with home platform ID/credentials
Interoperability Layer
Plat. user
Adapter
Digital platform 1
Access control services
Device
Device
Device
IC User
Plat. user
Authorized for all user services provided by platform 1
Adapter
Digital platform 2
Access control services
Device
Device
Device
IC User
Plat. user
Authorized for all user services provided by platform 2
Plat. user

FIGURE 11 - DATA DOMAINS AND DATA BOUNDARIES.

Figure 11 highlights the data and user authorization domains that are intrinsic to the scenario depicted in Figure 10. Each colourful area surrounding a digital platform or app represents its data domain (*e.g.,* home, energy), that is, the area into which data from that platform is naturally allowed, together with the realm in which a user that is authorized for that platform can operate. The green and blue domains, respectively referring to digital platform 1 and 2 cover the standard approach, that is, without user roaming capabilities (*i.e.,* a user that can access multiple platforms with the same credentials). This implies that a green user, authorized for the green domain is not able to access the capabilities for the remainder domains, as illustrated. The happens in the opposite direction regarding the user authorized for the blue domain.

The provision of a mechanism for user roaming allows for the IC user, depicted in yellow, to be able to access all resources via the generic adapters. Nevertheless, access control rules can be applied to this roaming concept to eventually block particular users from accessing particular services classes but allowing an IC user to be mapped to an internal standard one via the adoption of the generic adapter.

In the same way as with use access and control, each digital platform or service implements restrictions of the data is possible to be exported (*i.e.,* sent to another platform or service outside its domain). This is due to data privacy, eventual service level agreements and, most importantly, user consent via the General Data Protection Regulation (*i.e.,* GDPR). Likewise, the colourful domains also address the realms where specific data protection measures are installed. The generic adapters will also work as data boundaries, collecting from the service configuration which data can be forwarded or not. This refers either to operational data or metadata supporting the service execution.

In the interest of interoperability, the required data movement needs to be addressed by each digital platform and according to each service type. This indicates that for services, data can be aggregated at the level of the digital platform and pushed to the destination platform, without compromising the overall goals for the service. Moreover, operational data, which can become the most critical in terms of data privacy is only forwarded between pairs of generic adapters. This means that the required data flow holding operational data will not be forwarded to other proxy-like intermediate structures, and if does, data will not be stored, being just forwarded between parties.

3. DATA FLOW SPECIFICATION

The current chapter addresses the data flow specification regarding the basic functionalities towards the provision of semantic interoperability. The focus is particularly given to the components assembling the Semantic Interoperability Layer, the interactions via the InterConnect Adapters and the rationale for the interactions with the semantic reasoning and discovery functionalities.

3.1 INTEROPERABILITY LAYER

The Interoperability layer provides a set of basic functionalities that enables services and digital platforms to register their capabilities, via one of the available adapters. The core capabilities within the interoperability layer, namely the ones referring to Knowledge Engine that will support the provision of reasoning capabilities are:

- Metadata registration;
- Metadata discovery;
- Data Push/Pull.

Representation of the data flow diagrams with other semantic interoperability layer base technologies (namely WoT and S-LOR) will be elaborated in subsequent WP5 deliverables.

Entity Name	Description
Service	The offering of certain functionality from one entity/component to another authorized entity/component (e.g. service or software component) using (standardized) interfaces, compliant to certain IC Framework requirements.
Service User	An entity that uses a service as provided by another entity. This can be from a commercial viewpoint or a more technical one (e.g. 'software using services offered by other technical components'). The context of this term determines the viewpoint.
Service Server	The host where a given service is hosted.
Service Store	Complete catalogue of all interoperable services from energy and non-energy domains.
Adapter	The Interoperability Framework provides a set of adapters to allow vendors that are already compliant with industry standards to quickly connect their device/service to the Interoperability Framework.
Smart Connector	Generic software responsible for orchestration and reasoning.
Knowledge Directory	A central component, that registers the knowledge offered and requested by Smart Connectors. It does not perform any reasoning. There will be at least one instantiation of the knowledge directory within the project and pilots – as part of the IC service store.
Authentication Server	The entity responsible to grant or deny access to service users.
Blockchain Ledger	The distributed set of peers that hold data blocks chained together.
P2P Adapter	The Adapter that is capable to interact with the Blockchain ledger.

TABLE 3 - DESCRIPTION OF ENTITIES PRESENT IN THE SEQUENCE DIAGRAMS.[1]

[1] Matching Entity Names in Table 3 and Section 1.5, represent the same entity. They are depicted here in a short version.

3.1.1 METADATA REGISTRATION AND DISCOVERY

Generic adapters expose the capabilities of the service(s) and digital platform(s) where they are integrated. When launching the generic adapter (and its smart connector), it will register itself within the **knowledge directory** and push the service capabilities attached to it, as depicted in Figure 12.

The metadata registration and discovery data flow can be decomposed into 4 sub-flows for service and service capability registration, request updates from the knowledge directory and to allow unregistering a service.

The register service capability is considered to map distinct capabilities or functionalities that a given service (standalone or part of a digital platform) holds. This occurs internally to the InterConnect generic adapter, particularly within the service at the knowledge base and the smart connector, where a capability is registered and confirmed.

The service registration flow is considered when the generic adapter is booted, publishing in the knowledge directory the capabilities is comprehends. This is triggered by the smart connector that decomposes the mapped capabilities, extracting and adding the needed rules to internal reasoning mechanisms and subsequently publishing the available capabilities in the knowledge directory.

Updates maybe required during the operation of the smart connectors, which consider the poll directory updates data flow. This comprehends a request that is periodically originated from a smart connector, requesting the knowledge directory for updates, and installing new rules within is local reasoning capabilities.

The unregister service data flow is considered when services require to be removed from the available set of capabilities. This comprehends a stop request that one smart connector forwards to the knowledge directory, which will be propagated during future poll directory update requests.

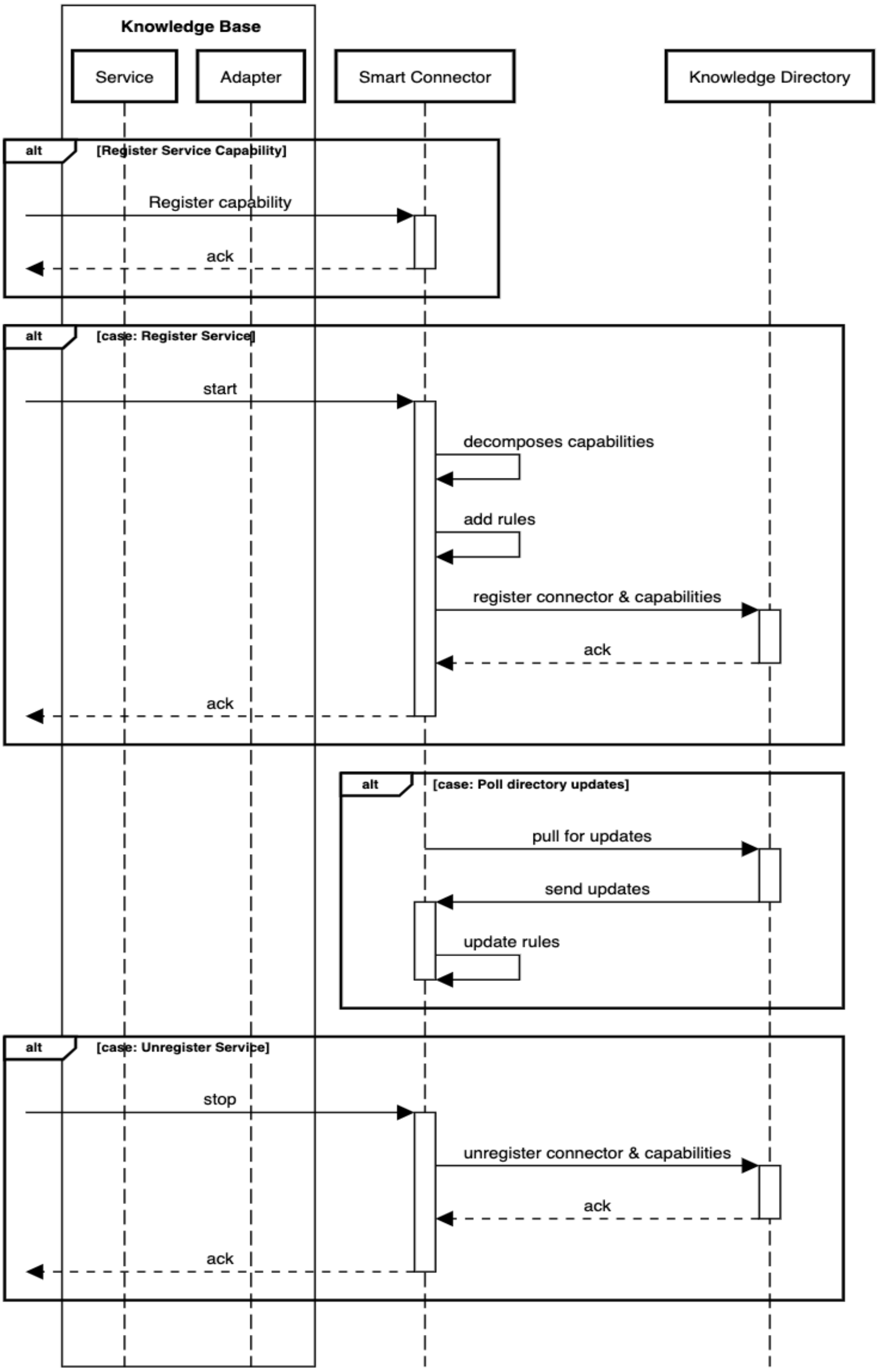

FIGURE 12 - GENERIC ADAPTER DATA FLOW.

3.1.2 DATA AND METADATA PUSH/PULL

InterConnect generic adapter, particularly within the service at the knowledge base and the smart connector, where a capability is registered and confirmed.

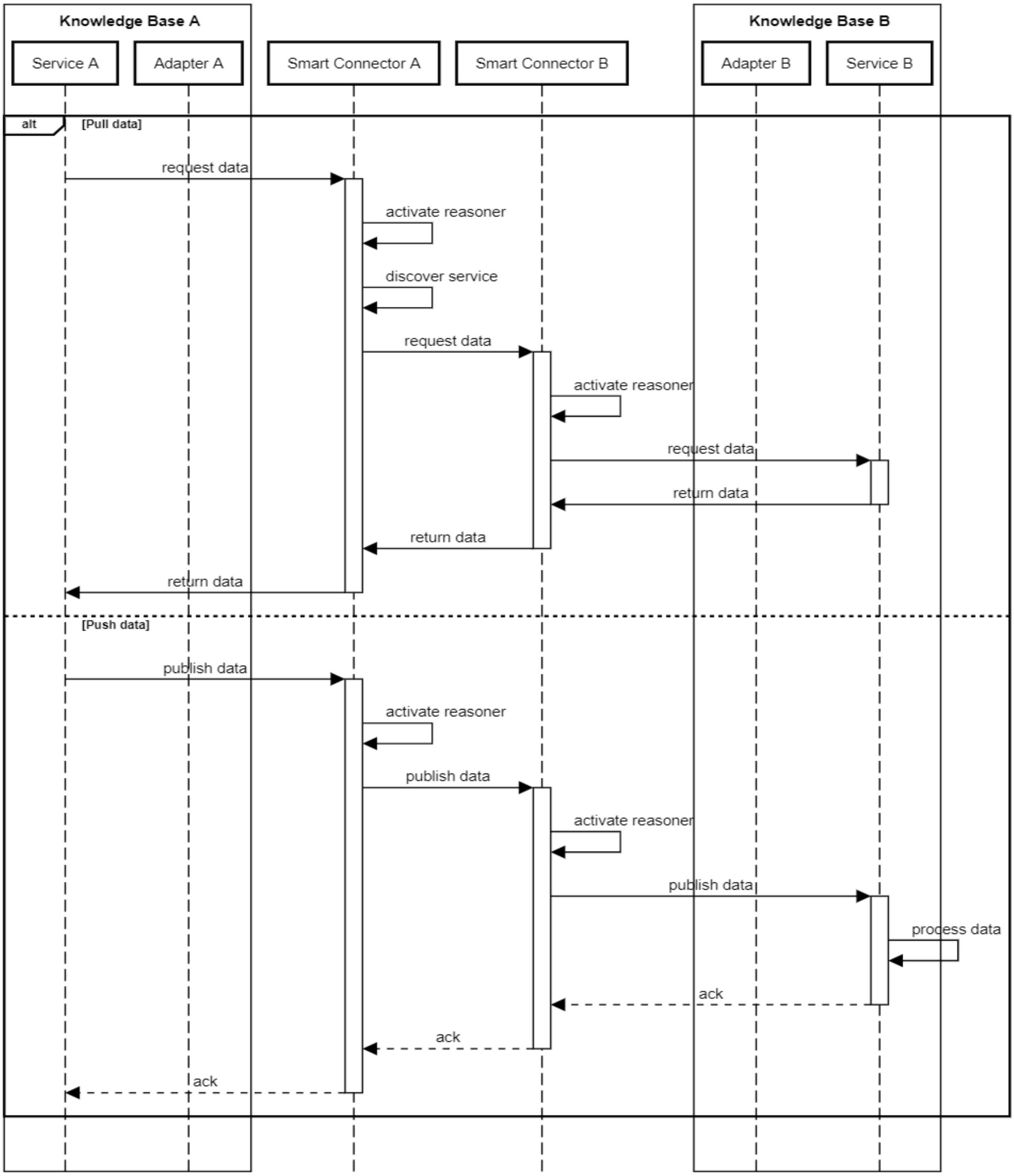

FIGURE 13 – DATA AND METADATA PUSH/PULL DATA FLOW.

The service registration flow is considered when the generic adapter is booted, publishing in the knowledge directory the capabilities it comprehends. This is triggered by the smart connector that decomposes the mapped capabilities, extracting and adding the needed rules to internal reasoning mechanisms and subsequently publishing the available capabilities in the knowledge directory.

Updates maybe required during the operation of the smart connectors, which consider pulling directory updates. This includes a request that is periodically originated from a smart connector, requesting the knowledge directory for updates, and installing new rules within its local reasoning capabilities.

The unregister service data flow is considered when services require to be removed from the catalogue of interoperable services. This comprehends a stop request that one smart connector forwards to the knowledge directory, which will be propagated during future pull directory update requests.

3.1.3 AUTHENTICATION

The authentication mechanism establishes how an InterConnect user requests and collects authentication. This mechanism is part of the capabilities provided through the generic adapters. The sequence is depicted in Figure 14.

Service User will ask the InterConnect **Service Store** (currently the goal is to host the InterConnect user/service Registration and Authorization mechanism as part of the IC Service Store) that will forward the request to the Authentication Server (provided by digital platform hosting a service or provided on the level of the InterConnect project – see D5.1 [6] section 5 for more details) that will return the authentication grant or not to the Service User. Authentication Server can be OpenID or any other solutions with support of OAuth2 (RFC 6749) mechanism where each existing digital platform (hosting services) as well as the IC interoperability framework itself can act as authenticator.

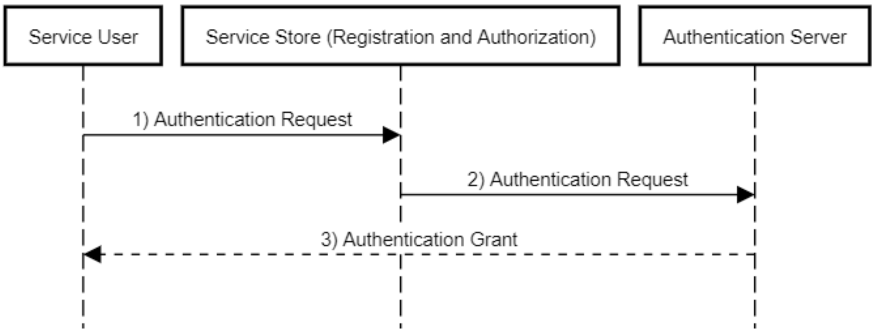
Service User
Service Store (Registration and Authorization)
Authentication Server
1) Authentication Request
2) Authentication Request
3) Authentication Grant

FIGURE 14 - AUTHENTICATION MECHANISM SEQUENCE DIAGRAM.

3.1.4 AUTHORIZATION

The authorization mechanism establishes how authenticated users can utilize and integrate InterConnect service. This is of great relevance for the services that might include data that are subjected to strict user consent or integral part of service provider's business models. The Authorization sequence diagram is displayed in Figure 15.

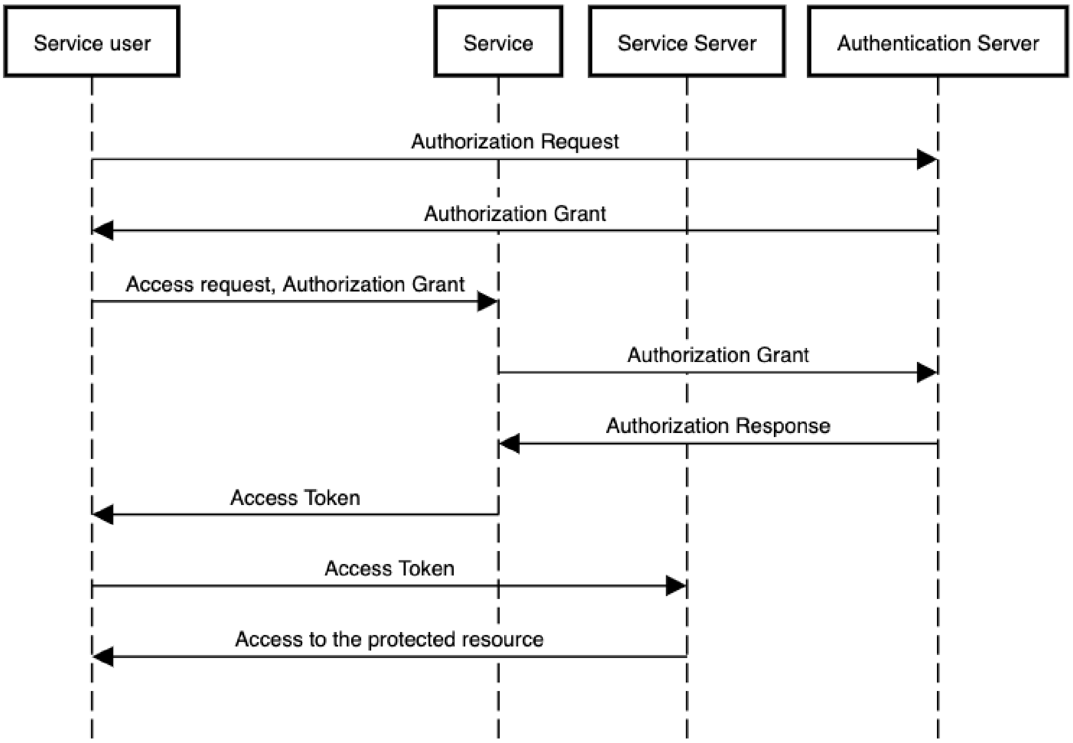

FIGURE 15 - AUTHORIZATION MECHANISM SEQUENCE DIAGRAM.

Service Users that want to access InterConnect services will ask for authorization from the Service Owner. The authorization grant will be forwarded to the InterConnect Authorization Service Server that will return a token. The token will enable Service User to access protected interoperable service .

3.2 INTERCONNECT SERVICE STORE

The InterConnect Service Store is one of the main components within InterConnect's Interoperability framework, providing a catalogue off all interoperable energy and non-energy services. The service store is designed as web application and, therefore has a frontend and backend systems, where specific processes occur. With the main objectives for the service store to be the fulfilment of requirements for both service providers and service adopters or integrators, the service store will provide means to register and manipulate service features. More details about IC Service Store architecture and functionalities it provides to different types of users can be found in D5.1 [6].

The service store exposes a set of basic functionalities, namely:

- **Service registration and onboarding**: each *service provider* registers its service by providing information regarding the specificities of the service to be considered;
- **InterConnect interoperability compliance test**: when a service is registered within the service store, services need to pass a compliance test, ensuring that semantic and data privacy constraints are met;
- **InterConnect interoperability certification**: after successful interoperability test, a compliance certificate is issued and stored within the project blockchain ledger [6].

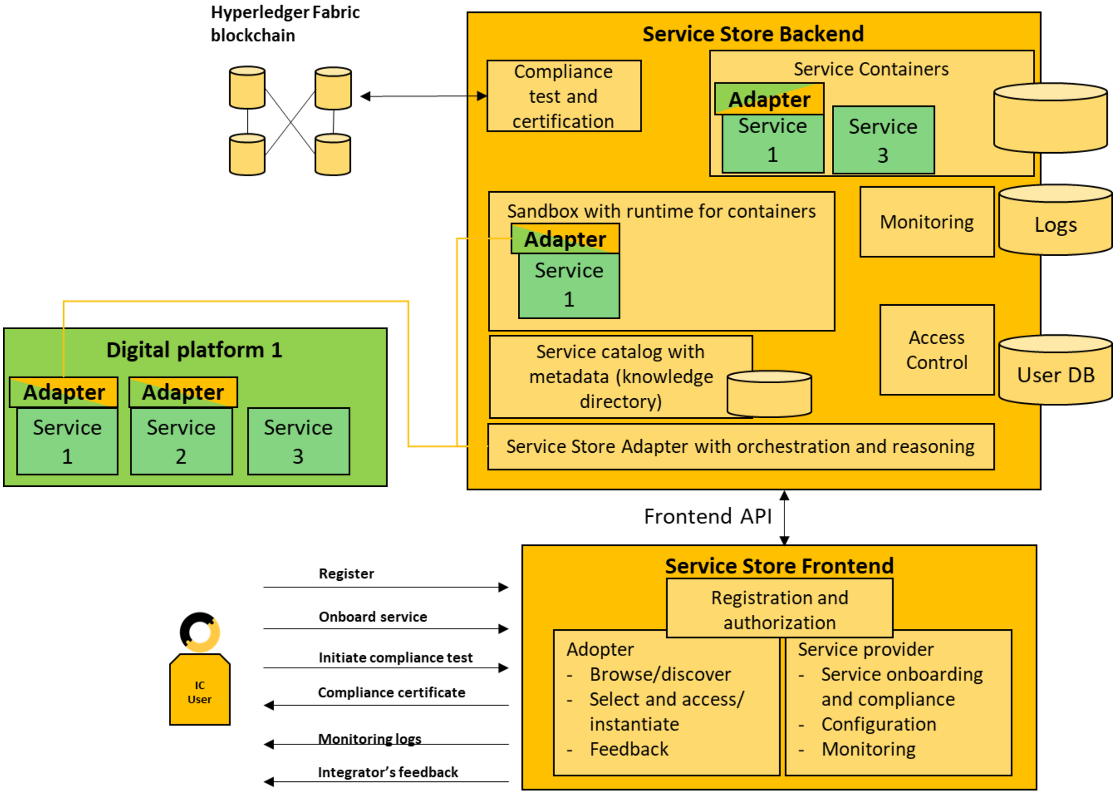

FIGURE 16 - SERVICE STORE FUNCTIONAL ARCHITECTURE AND FUNCTIONALITIES.

3.2.1 SERVICE STORE - REGISTRATION AND ONBOARDING

Services that are provided via the integration of a generic semantic interoperability adapter will require to register both the service and its capabilities within InterConnect's Service Store. The registration process occurs as depicted in Figure 17, where a service registration request is sent via the digital platform's generic adapter (*i.e.,* the generic adapter chosen by a given digital platform or service to enable interoperable services). The register request is then forwarded to the service store backend system that triggers a validation process. The validation process will ensure that the semantic interoperability requirements are met, according to the ontology in use (*i.e.,* SAREF), and will log the process by launching a new transaction within the blockchain ledger for latter validation and persistency. The exact requirements will be detailed in the upcoming release of D2.1 in M15.

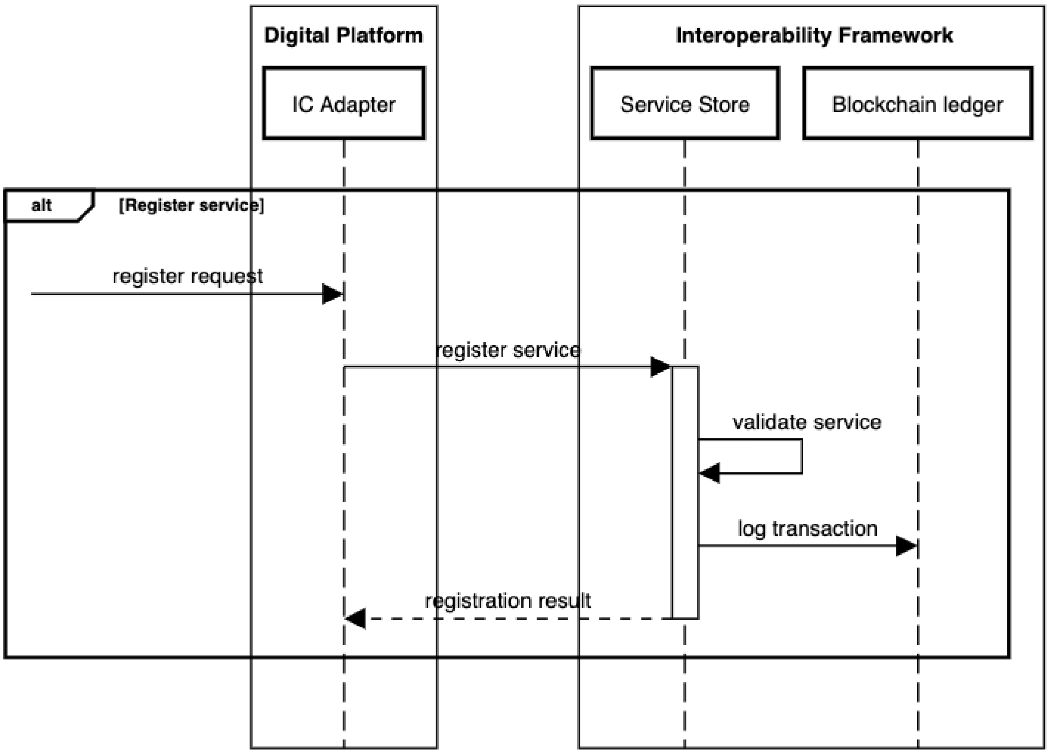

FIGURE 17 - SERVICE STORE SERVICE REGISTRATION.

3.2.2 SERVICE STORE - INTEROPERABILITY COMPLIANCE TEST AND CERTIFICATION

When a service is registered in the service store, it has to go through a compliance test, evaluating and ensuring that the service specification complies at the semantic level with the prescriptions asserted by the chosen ontology (*i.e.,* SAREF) and in terms of the data boundaries in order to ensure *data privacy and protection*.

The compliance test and its specific requirements will derive from the actions in WP2 [5] but will provide the means to make an assessment and unlock the certification stage, which in case of a successful result from the compliance test, will generate a compliance certificate.

When issuing a compliance certificate, a new record will be inserted into the blockchain ledger as a new transaction. This transaction once validated by the ledger and its peers, will provide means for any service adopter to be sure that a service is compliant with InterConnect’s interoperability strategy.

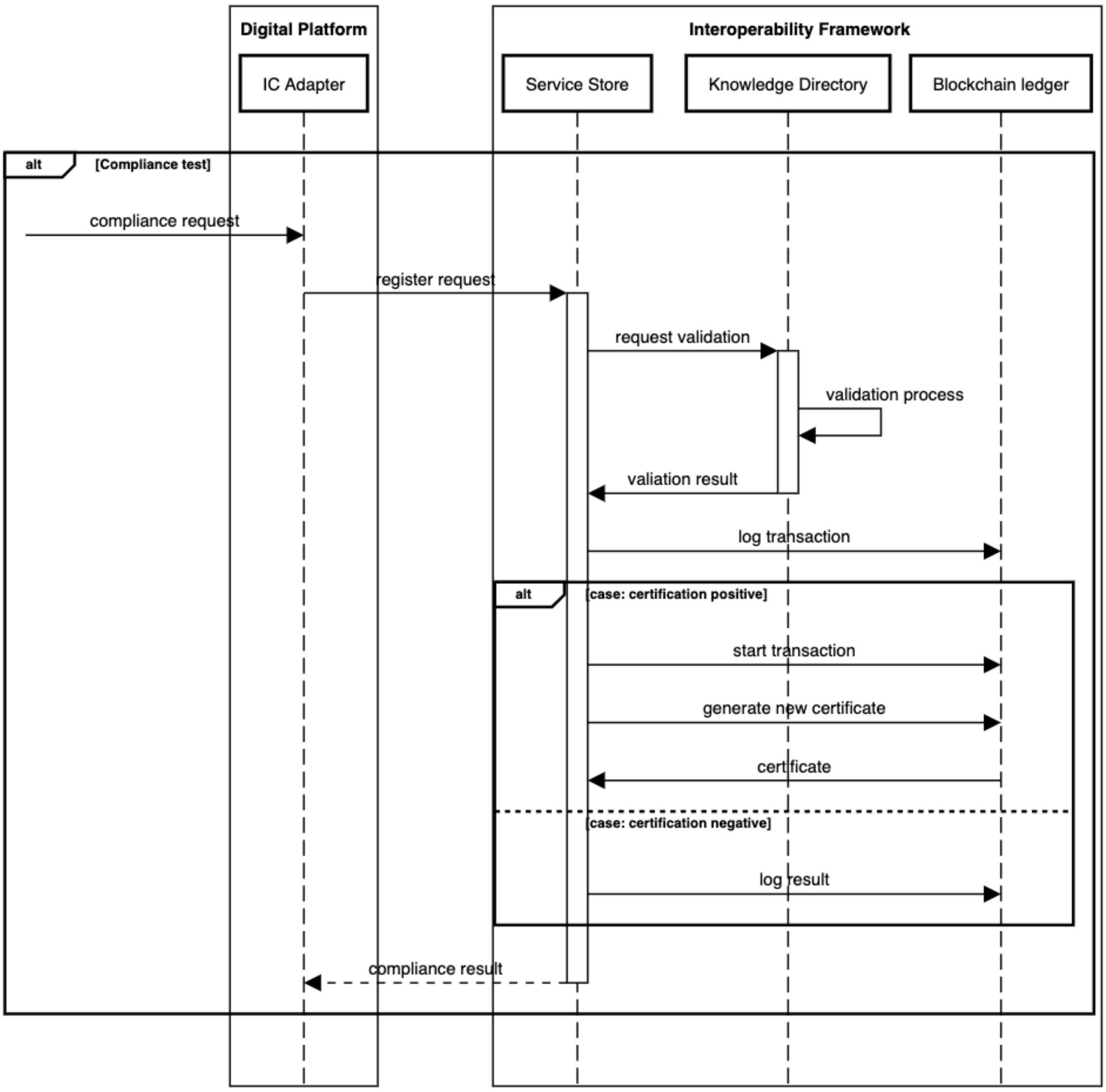

FIGURE 18 - SERVICE STORE COMPLIANCE AND CERTIFICATION.

3.2.3 SERVICE STORE - SERVICE OWNED OR HOSTED BY DIGITAL PLATFORMS

Most of the interoperable energy and non-energy services will be provided from their hosting digital platforms which are operated by the consortium partners. Services that are offered via the InterConnect Service Store can be accessed via its generic adapter. In this case, the service store operates as a registry and repository of interoperable services to be integrated within the ecosystem.

In this case the access is performed via the generic adapter that is integrated with the service hosting digital platform, it is via this entity that discovery of new services and/or capabilities occurs, by contacting the reasoning services. Operational data is then forwarded from the adopted generic adapter and set towards the destination generic adapter, once the reasoning mechanism identified the candidate destination. This interface also allows services to monitor their interaction via the common monitoring services within the service store and allow to access the community-oriented services and certification mechanisms made available by the P2P enablers within the framework.

A second possibility is for digital platforms to host service containers. In this case, service containers, made available via the service store catalogue, can be instantiated within the hosting environment of a private digital platform. For this case, metadata and operational data concerning the service deployed inside the container is available via the generic adapter that has been built-in inside the container image for that particular service. From this moment, this particular service is registered and made available within the knowledge directory of the service store. Different service hosting options in relation with the IC service store are elaborated in D5.1 [6].

3.3 P2P ENABLERS

The P2P (peer-to-peer) enablers will provide the required means for the usage of blockchain and smart-contracts to support new business models for decentralized arrangement of SAREF compliant energy markets. The provision of SAREF compliant services is achieved through the adoption of a Hyperledger Fabric (the chosen private permissioned and consortium blockchain ledger) adapter, enabling the blockchain ledger to reach the service definition in the Service Store Interoperability Layer. More details about the IC P2P marketplace enablers are provided in D5.1 [6].

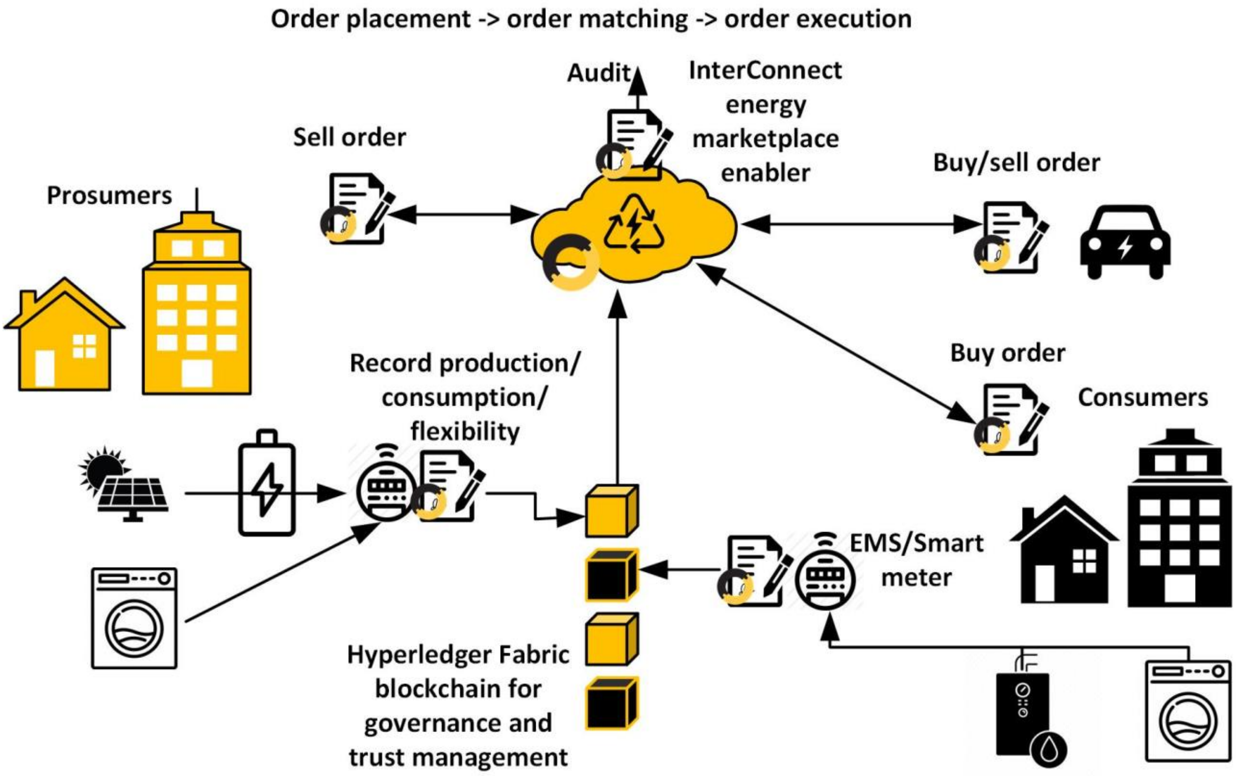
Order placement -> order matching -> order execution
Audit
InterConnect energy marketplace enabler
Sell order
Buy/sell order
Prosumers
Record production/ consumption/ flexibility
Buy order
Consumers
EMS/Smart meter
Hyperledger Fabric blockchain for governance and trust management

FIGURE 19 - P2P ENABLER'S CONCEPT FOR ACCESSING BLOCKCHAIN SERVICES.

3.3.1 P2P ENABLERS - LAUNCH SERVICE SMART CONTRACT

The P2P enablers will provide a base configuration and template for the creation of smart-contracts, that is, autonomous agents the live within the blockchain ledger. This will provide the necessary gateway for distributed applications, specially geared towards distributed energy markets to reach-out to the semantically driven SAREF-based services. In line with the concept for the generic adapters for the digital platforms and services, the P2P enablers smart contract template will provide the means for stakeholders to implement their services and deploy them in the distributed ledger.

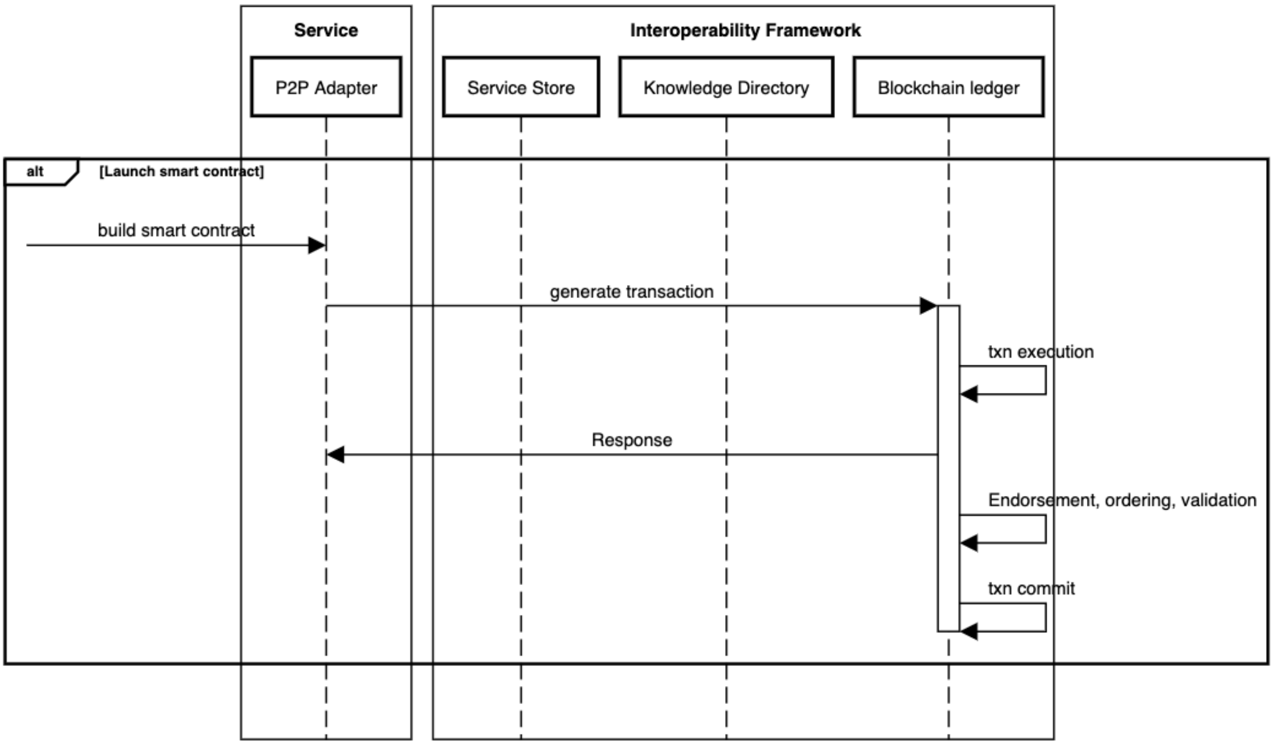

FIGURE 20 - LAUNCHING SMART CONTRACT

In order to deploy distributed services to autonomously run in the ledger, once a service owner details the business logic for the service in the interoperable smart contract template, that asset has to be deployed in the blockchain ledger. Deploying the service is depicted in Figure 20. Once the request is made via the P2P adapter, a transaction will be generated holding the smart contract for the service and sent to the Blockchain ledger. Then internal steps are taken to ensure transaction execution, verification and validation (via the endorsement and ordering of the transaction) and committed in the ledger.

3.3.2 P2P ENABLERS - STANDARD TRANSACTION EXECUTION

The execution of service actions defined within the construction of a smart contract are mapped into transactions that are launched in the blockchain ledger. Despite the action type, or parties that it encompasses, executing a transaction comprises a set of steps to launch, require peer participation, ordering and validation, before a transaction can be finally committed in the distributed ledger. These steps are depicted in Figure 21.

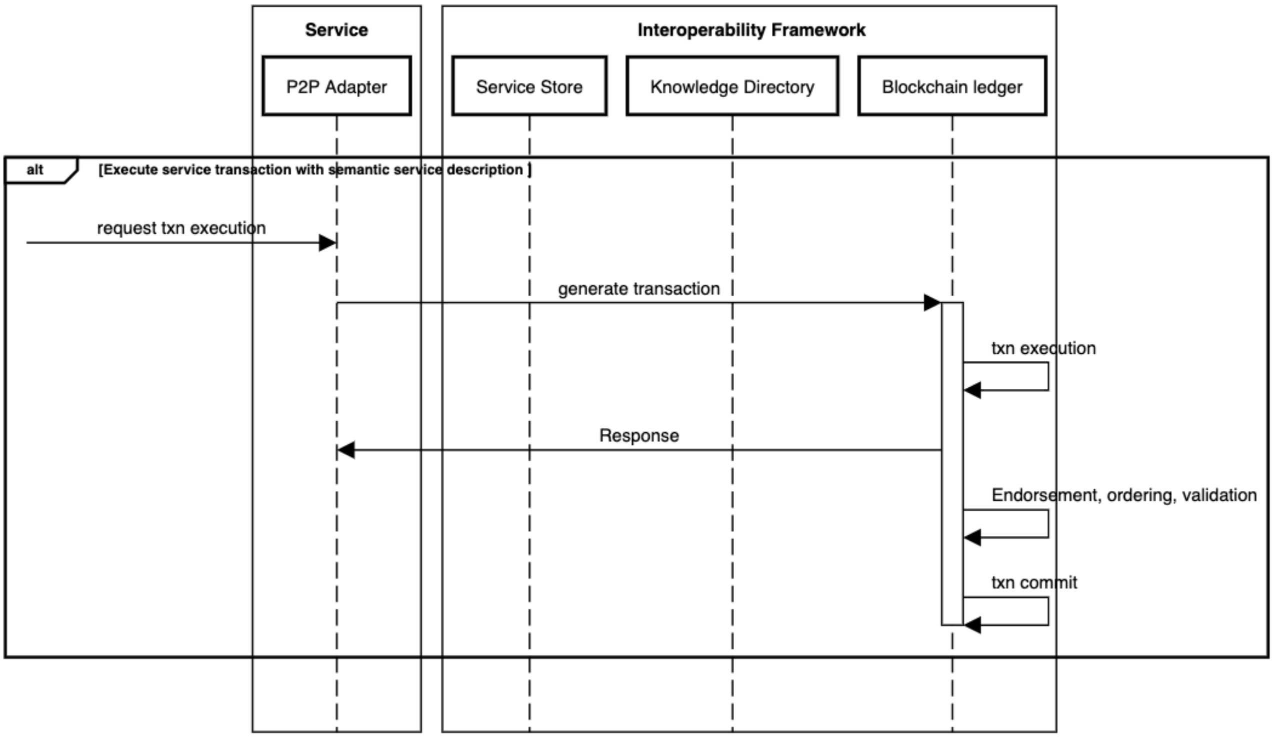

FIGURE 21 - STANDARD P2P TRANSACTION EXECUTION.

For a given transaction, the P2P adapter invokes the smart contract function in the peers that are associated with that action (endorsing peers), according to the endorsing policy of the smart contract. The endorsing peers execute the transaction and verify if the local data they hold supports the transaction request. The request is then cryptographically signed by each one of the endorsing peers and sent to be ordered and validated. This is depicted in Figure 21 as the endorsing, ordering and validation step. Briefly, this process allows all distributed peers to ensure the validity of that transaction before it is persistently written in the blockchain ledger, as part as one of the blocks in the chain. This process is the standard process that supports all transaction execution.

3.3.3 P2P ENABLERS - SERVICE TRANSACTION EXECUTION WITH REASONING

Each smart contract mapping a service will hold several actions that within that service's business hold the logic for the service. Each action is mapped into a transaction pattern in the smart contract and allows it to be executed in an autonomous way within the blockchain ledger. Executing a service request within the smart contract is depicted in Figure 22.

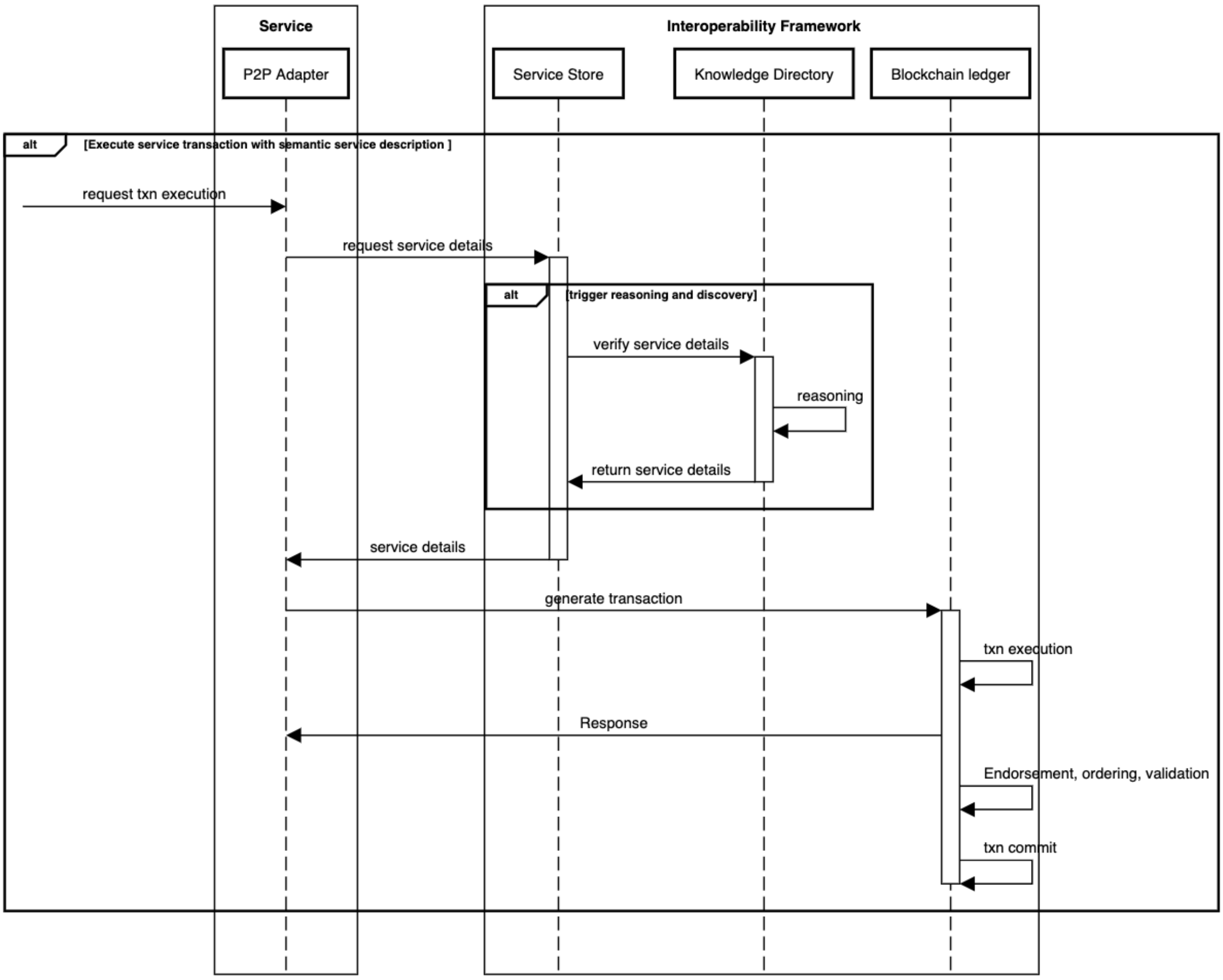

FIGURE 22 - EXECUTION SERVICE TRANSACTION WITH REASONING.

Once the service request is triggered, the P2P adapter will request the service store for the services properties and, in case there is need to activate reasoning or discovery, the service store will request that execution from the knowledge directory. After the service details are returned to the P2P adapter, the transaction execution follows the standard path in order to be committed.

4. CONCLUSION

This deliverable provides definitions for data flow management approaches and key concepts enabling data exchanges in semantic interoperable manner. First, the general concepts about semantic interoperability are presented with overview of the InterConnect semantic interoperability layer. This layer comprises interoperability adapters and connectors utilizing the selected ontology and unifying interfacing protocol. Two enabling technologies for the semantic interoperability layer are presented (Knowledge Engine and S-LOR). Also, data boundaries, with respect to the semantic interoperability framework and already existing digital platforms and services, are defined.

Nest, this deliverable provides and discuss the message exchange diagrams for key functionalities and enablers behind the IC interoperability framework. The message diagram entities are defined based on the Knowledge Engine technology as the basis for semantic interoperability layer.

This deliverable document should be used together with D5.1 “Concept, design and architecture of the interoperable marketplace toolbox” since they complement each other. These two deliverables report on the key outcomes of WP5 Task 5.1 – Interoperability framework and service store architecture and specification. Detailed elaboration of the data flow management for all functionalities and enablers of the InterConnect interoperability framework will be reported in subsequent WP5 deliverables.

REFERENCES

EXTERNAL DOCUMENTS

[1] Sensor-based Linked Open Rules (S-LOR): An Automated Rule Discovery Approach for IoT Applications and its use in Smart Cities. 3rd International ACM Smart City Workshop (AW4city) in conjunction with 26th International World Wide Web Conference, April 3-7, 2017, Perth, Australia. Amelie Gyrard, Martin Serrano, Soumya Kanti Datta, Joao Bosco Jares, Muhammad Intizar Ali (2017)

[2] Semantic Interoperability for the Web of Things (2016)

[3] Semantic IoT Solutions - A Developer Perspective (2019)

[4] Towards semantic interoperability standards based on ontologies (2019)

INTERCONNECT DOCUMENTS

[5] Deliverable D2.1 -"Secure and interoperable IoT smart home/building and smart energy system reference architecture", upcoming version due in M15.

[6] Deliverable D5.1 – "Concept Design and architecture of the interoperable marketplace toolbox".

[7] Deliverable D11.2 - "Data Management Plan".