Contract No. H2020 – 826172

SEMANTICS FOR PERFORMANT AND SCALABLE INTEROPERABILITY OF MULTIMODAL TRANSPORT

D5.1 –Requirements, Scenarios and Use cases For The Proof-of-Concept (C-REL)

Due date of deliverable: 31/10/2019

Actual submission date: 24/04/2020

Leader/Responsible of this Deliverable: POLIMI

Reviewed: Y

Document status		
Revision	**Date**	**Description**
1	28/11/2019	Complete version, for final revision
2	03/12/2019	Version for TMC approval
3	06/12/2019	Final version after TMC approval and Quality Check
4	24/04/2020	Final version after comments from reviewers and TMC approval

Project funded from the European Union's Horizon 2020 research and innovation programme		
Dissemination Level		
PU	Public	X
CO	Confidential, restricted under conditions set out in Model Grant Agreement	
CI	Classified, information as referred to in Commission Decision 2001/844/EC	

Start date of project: 01/12/2018 Duration: 25 months

EXECUTIVE SUMMARY

This deliverable presents the scenarios that will be implemented in the C-REL version of the SPRINT demonstrator. Each scenario is intended to highlight the features of one or more of the components of the Shift2Rail Interoperability Framework that have been identified in deliverable D3.3 - "Design of Architecture, Testing Infrastructure, Test Cases and Benchmarks of the IF (C-REL)".

ABBREVIATIONS AND ACRONYMS

Abbreviation	Description
AM	Asset Manager
API	Application Programming Interface
C-REL	Core Release
CSV	Comma-Separated Values
EU	European Union
F-com	Finland-Based Company (Scenario 4)
F-REL	Final Release
H2020	Horizon 2020 framework programme
IDE	Integrated Development Environment
IF	Interoperability Framework
ISA	IT supplier and Software Application
JAR	Java ARchive
JSON	JavaScript Object Notation
NAP	National Access Point
NeTEx	Network Exchange
RDF	Resource Description Framework
S2R	Shift2Rail
SC	S_BusTravel Converter (Scenario 5)
SPARQL	Protocol and RDF Query Language
TO	Transport Operator
TSP	Transport service provider
TrSP	Travel Service Provider

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

This deliverable is propaedeutic to the preparation of the demonstrator of the SPRINT project. More precisely, it describes, on the one hand, the scenarios that will be demonstrated through the platform delivered in D5.2 – "Software release of the proof-of-concept in its technical environment (C-REL)" (which will be validated in D5.3 – "Validation of pilot implementation (C-REL)") and, on the other hand, the technical requirements of the demonstration platform.

The scenarios have been defined starting from those already identified in deliverable D3.2 - "Performance and scalability requirements for the IF (C-REL)" (which have been used to derive the first set of requirements of the platform); those scenarios have been further detailed, in particular with respect to the set of components that have been identified as being part of the Shift2Rail (S2R) Interoperability Framework (IF) in deliverable D3.3 - "Design of Architecture, Testing Infrastructure, Test Cases and Benchmarks of the IF (C-REL)". For each scenario, the following descriptions are provided:

- An informal description of the scenario, including involved actors and IF components.
- A UML activity diagram detailing the steps of the scenario.
- A description of the demo platform required to carry out the scenario.

The scenarios are initially defined taking into account the final demonstrator that is foreseen to be delivered at the end of the project. However, for those that in the C-REL version of the platform will be demonstrated in a limited manner, the restrictions applied in the C-REL version are highlighted.

The deliverable is structured as follows. Section 2 provides a summary of the S2R IF that was first presented in deliverable D3.1 – "State-of-the-art and best practices in architecting systems processing semantic data", and which was then detailed in deliverable D3.3 - "Design of Architecture, Testing Infrastructure, Test Cases and Benchmarks of the IF (C-REL)". Section 3 then presents the scenarios to be demonstrated; the scenarios are grouped by the main components of the IF that they address: Asset Manager, Converters, Resolvers.

2. OVERVIEW OF IF ARCHITECTURE

Figure 1 represents the architecture of the IF, which has been defined in detail in previous SPRINT deliverables D.3.1 and D.3.2. In addition to various data stores, the IF is composed of firstly, a **User Manager** component that takes care of common user account management and access control aspects, and, secondly, the Asset Manager, which is the central component of the IF.

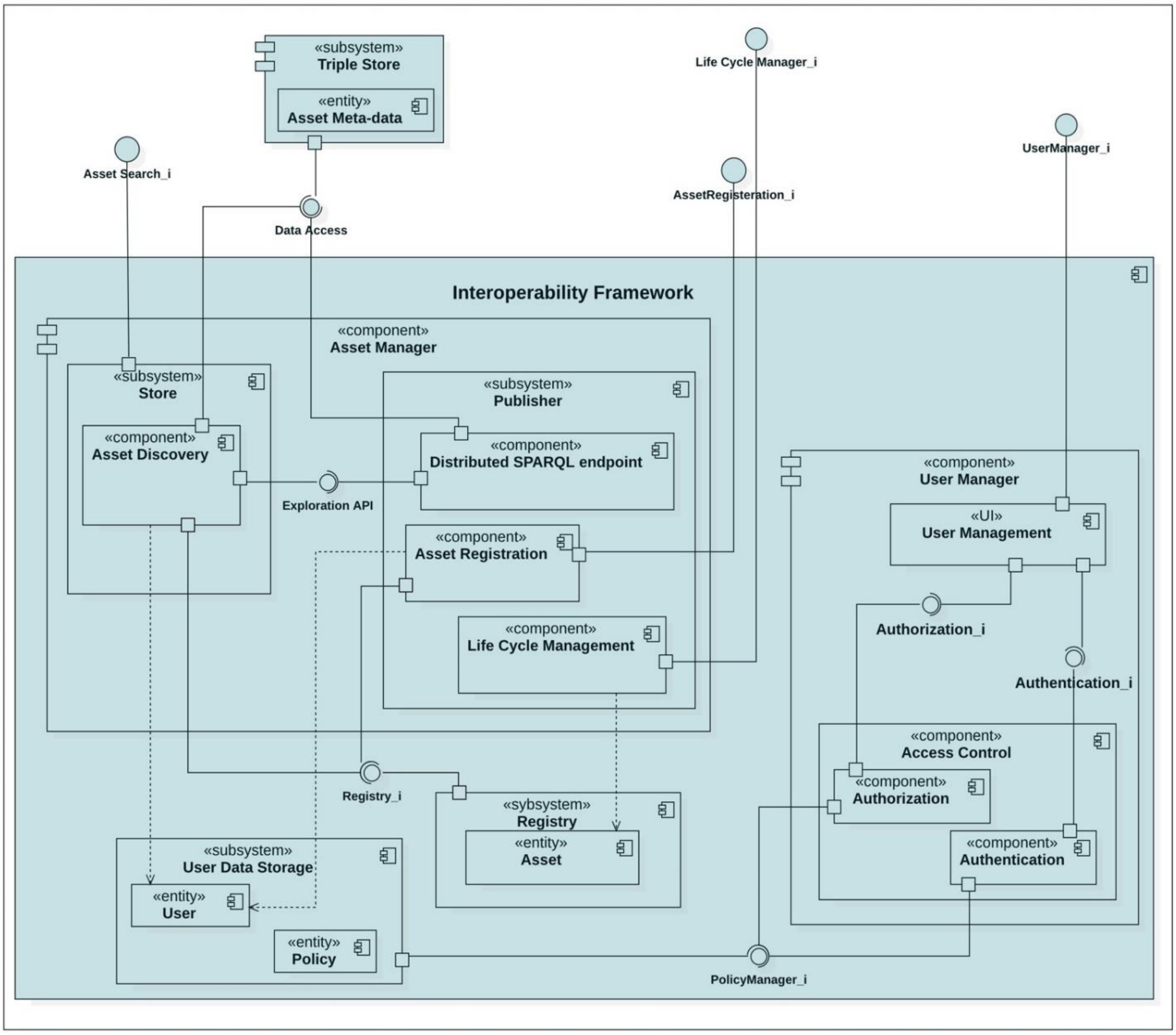

Figure 1 Component Diagram of the IF

Asset Manager offers the basic functions to publish, share, discover, maintain and manage various artefacts that might be published/utilised by external and internal components of the IF. Asset Manager itself has further subcomponents for accomplishing different tasks managing the IF, including **Life Cycle Management** and **SPARQL endpoint**. The former provides a means to formally define the workflow of stages through which different types of assets must go in their life cycle, while the latter offers a federated SPARQL query processing mechanism. In section 3.1, we introduce a set of scenarios (Scenarios S1 to S4) which illustrate the application and usage of functions offered by these components.

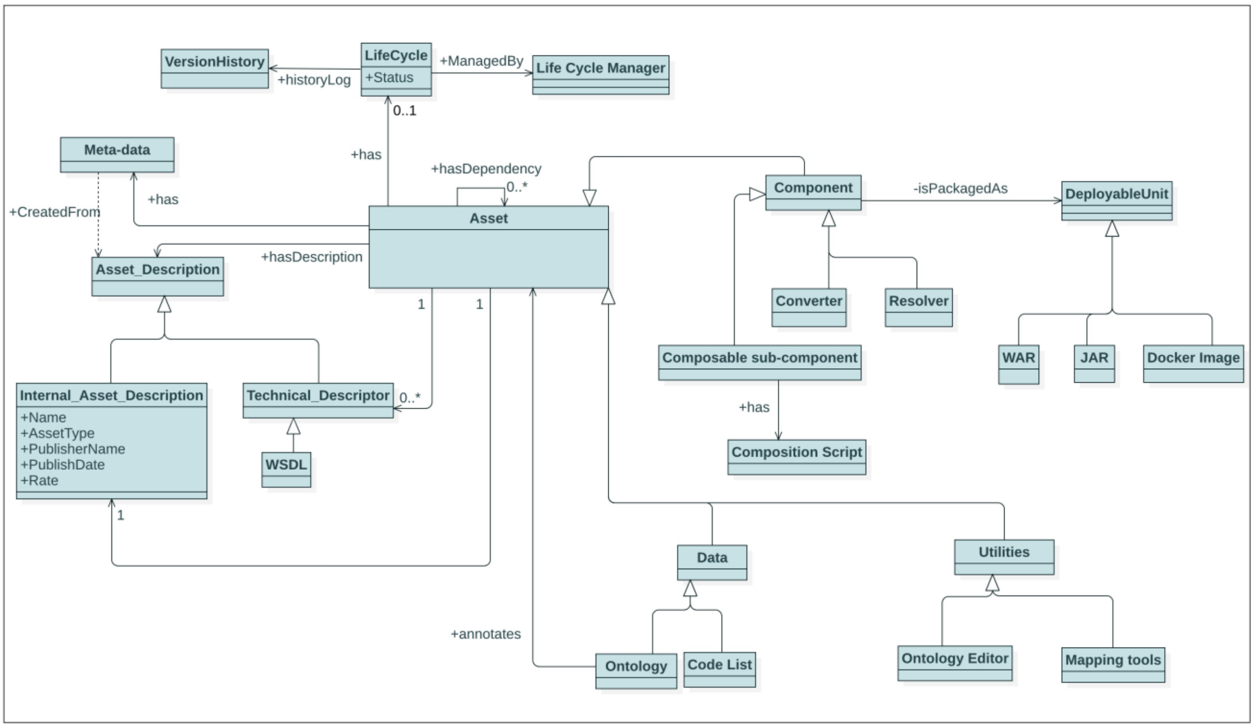

Figure 2 Asset Domain Model

Section 3.2 then covers use cases for IF tools such as the mapping IDE, and additional services and features of the IF, including multiple deployment options to engage with different types of assets. Figure 1 represents the architecture of the IF, which has been defined in detail in previous SPRINT deliverables D.3.1 and D.3.2. In addition to various data stores, the IF is composed of firstly, a **User Manager** component that takes care of common user account management and access control aspects, and, secondly, the Asset Manager, which is the central component of the IF. represents a more detailed specification of the notion of Asset along, with some asset types supported by the IF. Among them, Converters and Resolvers are highlighted in this document.

With reference to our requirement analysis for the design and development of the IF, the lack of standardisation and harmonisation transportation data format and specification was evident [1]. In this context, the capability of conversion/translation between heterogeneous standards and data format is the key point to achieve interoperability. A system is semantically interoperable when exchanged data could be understood unambiguously and interpreted uniformly. In this direction, we have anticipated the Converter as a component of the IF, which is a software acting as an adapter between two distinct formats to map the information expressed in one format to the other.

The conversion process, however, could be implemented following various architectural patterns, and with the help of different technologies and tools, as discussed in depth in companion deliverables. In short, the SPRINT converter is an extension to ST4RT converter [2], implemented to make the internal Converter architecture more modular and loosely coupled (For more information about the Converter component please refer to SPRINT deliverables D3.1, D3.2 and D3.3).

Resolver is a type of asset (of type Component) dedicated to providing access, location, relocation and replication transparency to interacting applications, masking them from the physical distribution, access protocols and formats of meta-data and data resources available in the Data Abstraction. A resolver could be seen as a special-purpose discovery component focused on the discovery and access of a specific resource. For instance, Location Resolver returns a list of Stop Places within a requested radius from a point specified by its geographical coordinates; Travel Expert resolver returns a list of web services that create mobility offers for a specified departure and destination pair of Stop Places. All Resolvers have a common structure and perform the same activities: they are distinguished only by the specific data or service discovery that is requested, i.e. by the specific query specified by the requesting Actor. This common activity flow is described in scenario S10.

As elaborated in SPRINT deliverable D3.3, one of the interesting features of the IF is that it provides multiple options for the deployment of its components such as Converters and Resolvers. It extends the functionality and applicability of the IF for a wider range of situations, since it lets the consumer choose the optimal/application solution for their specific use case. Hence, the scenarios in section 3.2 (specifically through S5, S6, S8 and S10), mainly focus on the realisation and use of Converters and Resolvers.

Finally, we have one use case scenario (S7) for the use of the Mapping IDE, which will be exposed as a stand-alone tool in the IF. The Mapping IDE facilitates an automated mapping between two formats which then could be used for any purpose, including the construction of Converter components (by IF developers as well as any interested contributor/consumer of IF services).

3. PROOF OF CONCEPT FOR BASIC USE CASES

This chapter describes the details of the use case and demo scenarios sorted based on the involved components. In specific, Section 3.1 covers scenarios related to Asset Manager, while Section 3.2 is devoted to scenarios related to services offered, and in particular those related to Converter and Resolver components. The scenarios described in this chapter describe advanced functions and features of the IF that might be more relevant for F-REL, and the C-REL demonstration of each scenario might be mainly focused on the core aspects as described in following; hence, for each scenario we specify the parts that will be demonstrated in C-REL.

3.1 IF BASIC FUNCTIONS

The scenarios of this section cover various use cases for the basic functions of the IF, which involve the following components/tools of the IF:

Table 1 List of Involved Component/tools of the IF for basic function scenarios

IF Component/Tools	Use Case Scenario
Asset Manager	Scenario S3 : Service/Asset Discovery (Simple Discovery) Scenario S4: Service/Asset Discovery (Distributed SPARQL endpoints)
User Manager	Scenario S1 : Joining the IF Use case (User Registration) Scenario S2 : Joining the IF Use case (Provider Registration)
SPARQL endpoint	Scenario S4: Service/Asset Discovery (Distributed SPARQL endpoints)
Asset Discovery	Scenario S3 : Service/Asset Discovery (Simple Discovery)

3.1.1 Joining the Interoperability Framework Use case

With the reference to the Basic Scenario 1 in Deliverable 3.2, here we investigate the very first step of communication and interaction with IF which includes User Registration and Login Process.

Scenario S1:

Description

Table 2 Scenario S1: Joining the IF Use case (User Registration)

Actor	**HT-train (TSP):** a train service provider
Target Component/Sub-system/Entity	User Management

Description	This scenario illustrates the registration process for a user that intends to join IF as a “Service Provider”
Story	Bill is an employee in HT-train which is responsible to register this operator to IF. He goes to IF website of Italy *[IF].it* and selects to register to IF as Service Provider Role. To create an account, he inserts all the required information related to himself as well as his company, including username, password, type of the company, etc. After successful registration and confirmation of the identity of the registered user, he is then redirected to the **Back-Office** view of **Asset Manager (AM). Back-Office** is conceived as the provider's panel in IF and presents required interfaces to the functionalities available for a service provider such as Asset Registration.

Activity Diagram

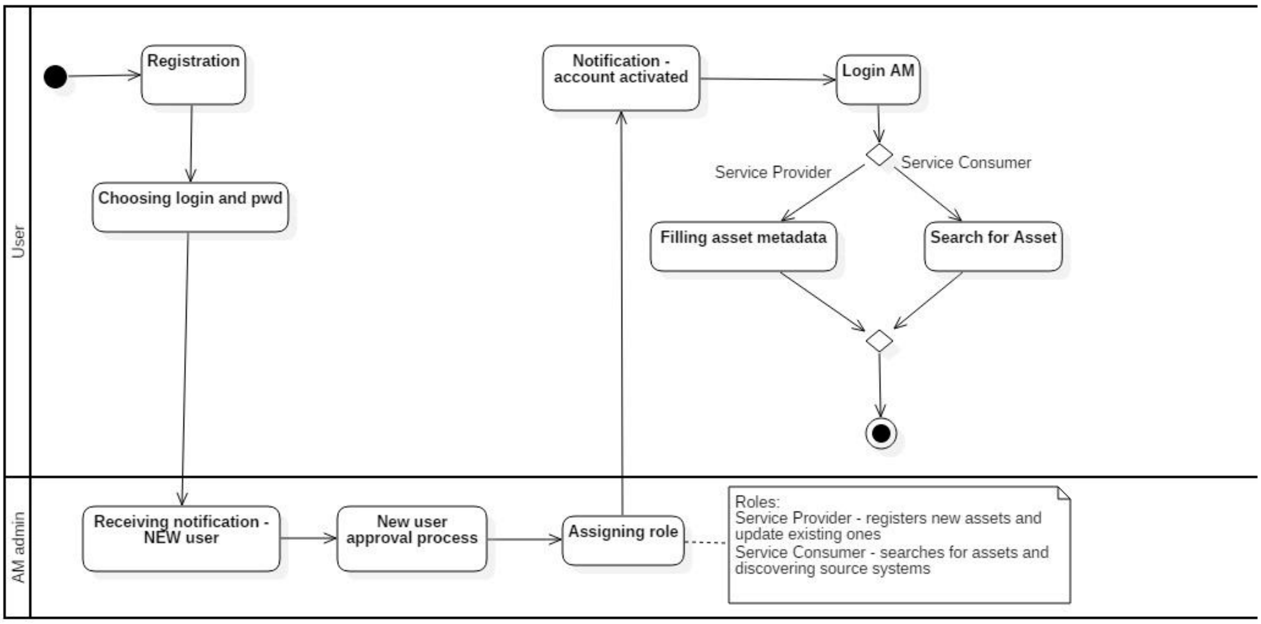

Figure 3 - User Registration Activity Diagram

Demo/Deployment platform description

This scenario is mainly concerned with the user management aspect of the Interoperability Framework. To demonstrate it, an instance of the Asset Manager will be deployed and configured with the initial set of required roles (“administrator” and “service provider”). The Asset Manager registration process will need to be flexible enough to be easily reconfigurable in a production environment with different user registration policies. The user will register using the Asset Manager Web interface, and both the user and the Asset Manager administrator will be notified about the registration progress using email messages.

Scenario S2:

Description

Table 3 Scenario S2: Joining the IF Use case (Provider Registration)

Actor	**SafeTravel (TrSP):** Travel Applications for smartphone
Target Component/Sub-system/Entity	User Management
Description	This scenario illustrates the registration process for a user that intends to join IF as a "Service Consumer"
Story	Alice is an employee in **SafeTravel** company that develops a smartphone application for ticket search and booking. Alice is responsible to register this transport application to IF. She goes to IF website and selects to register to IF as Service Consumer Role. To create an account, she inserts all the required information related to herself as well as the company, including username, password, type of the company, etc. After successful registration and confirmation of the identity of the registered user, she is then redirected to the Front-end view of **Asset Manager,** which is conceived as the consumer's panel in IF and presents required interfaces to the functionalities available for a service provider such as Asset Discovery.

Activity Diagram

This diagram corresponds to the previous scenario for user registration process.

Demo/Deployment platform description

This scenario shows another basic usage of the Asset Manager. To demonstrate such scenario, we will use the same Asset Manager instance which will be used to demonstrate Scenario 1. As in Scenario 1, the user will register using the Web interface provided by the Asset Manager. We will adapt the configured user roles by adding the "service consumer" role and by assigning it different rights than the “service publisher”. A user with the “service consumer” role will not be able to access to the Asset Manager Publisher Web interface, nor to its Web API.

3.1.2 Service/Asset Discovery Use cases

Scenario S3

Description

Table 4 Scenario S3: Service/Asset Discovery (Simple Discovery)

Actor	NewRail, a rail operator which just joined the IF ecosystem
Target Component/Sub-system/Entity	Asset Manager: Asset Discovery
Description	This scenario demonstrates two basic functionalities of the Asset Manager, namely the possibility to browse the available assets by asset type, and the possibility to perform a faceted search.
Story	NewRail wants to assess the business opportunities of the IF ecosystem, which they just joined. They decide to browse the available assets in the Asset Manager as a first step, exploring the different functions made available by other operators. Then they decide to explore the technical side of the problem, by using the search functionality of the Asset Manager to find out which systems are compatible with the data model and messages specifications that they are using.

Activity Diagram

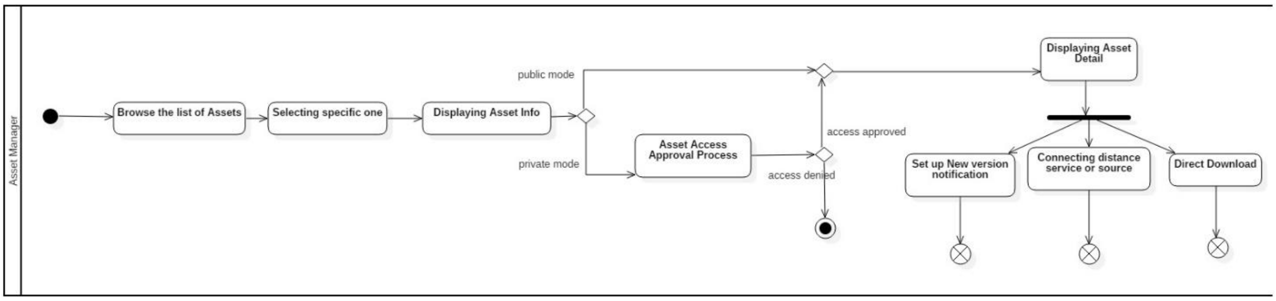

Figure 4 Asset Discovery Activity Diagram

Demo/Deployment platform description

Demonstrating the discovery feature of the IF will be performed in two different ways using the Asset Manager Web user interface. The Asset Manager will access its local RDF repository using SPARQL queries to retrieve metadata about the assets. A first possible demonstration is to let users browse the available assets by type, and let them choose the most appropriate one. Another option will be to use the asset search, that will let users narrow down the possible interesting assets by specifying a set of search filters (derived from the available metadata). If the authorisation policy of the selected asset will be "open access", then the user will view all the metadata information about the asset and will also have access to all its data attachments. If otherwise the authorisation policy of the asset will be based on user request and on the explicit permission given by the owner, the user will ask the right to access the asset. The owner will be then notified about the user's request and will provide his consent using the Asset Manager.

Scenario S4

Description

Table 5 Scenario S4: Service/Asset Discovery (Distributed SPARQL endpoints)

Actor	B-Com: A Belgian-based TrSP which it is hosted by the (National Access Point) NAP of Belgium S-com: A Spanish-based TSP named "S-com" already hosted by the NAP of Spain. S-com offers travel services within and beyond the Spain boundaries. MyMobility: A Italian company providing transport services in many regions of Europe
Target Component/Sub-system/Entity	Asset Manager: Distributed SPARQL endpoint
Description	This scenario illustrates querying the SPARQL engine
Story	MyMobility is interested in expanding its routes services across Europe, it wants to discover which service providers are publishing public transport data that may be interesting for them. Once that IF receives a request from MyMobility about catalogues describing data of public transport providers, a distributed SPARQL query process is started. The company requests the list of publishers (transport service providers) of the datasets containing information of the different public means of transport. Since MyMobility is focused on Belgium and Spain to expand its routes services, we treat S-Com and B-Com as two different data sources, whose resources can be combined to find metadata catalogues of the NAP datasets from Belgium and Spain. For this, IF needs to make use of the services provided by B-Com and S-Com, independently of where they are coming from. As such, IF will issue the query to the asset manager distributed SPARQL query engine, which will perform the usual steps of generation of subqueries for each selected source, generating a query plan, rewriting the subqueries considering potential inferences, translating those subqueries and executing them so that the results can be integrated and delivered to the asset manager.

	To expand its route services in Madrid, MyMobility refines its search after an initial request for catalogues to discover a more detailed information about the Spain datasets comprising descriptions of the stations, bus stops and other infrastructures of S-Com.

Activity Diagram

This diagram corresponds to the preceding scenario considering asset discovering is over a distributed environment.

Demo/Deployment platform description

For the C-REL demonstration, a set of federated SPARQL queries can be executed by a distributed SPARQL endpoint like Ontario using the Asset Manager Web user interface. Similar to asset discovery process in Scenario 3, the distributed SPARQL endpoint capabilities are to browse the available assets by asset type, and to perform a faceted search but the RDF repositories are distributed in several data sources. As a proof of concept, the Asset Manager will access two different RDF repositories which host different datasets and whose resources will be combined. Datasets are stored as CSV, JSON, XML and relational databases. Ontario will execute the federated query on RDF repositories and combine resources from two RDF repositories to process the results. Additionally, the Demo platform will not address access control policies, but it will just assess the technical possibility to do querying on multiple metadata RDF repositories.

3.2 PROVIDED SERVICES/TOOLS

The scenarios of this section focus on use cases for services and tools offered by the IF. For example, it highlights various deployment options for engaging with a particular component. To present the use cases for such services (e.g., direct download deployment option) we designed scenarios involving the following components of the IF:

Table 6 List of Involved Component/tools of the IF for scenarios focusing on services/tools

IF Component/Tools	Use Case Scenario
Asset Manager	Scenario S8 : Automatic converter building Use case Scenario S9: Fast Adaptation to Peaks Use case Scenario S10 : Special Purpose Asset Discovery Package: Resolver
Asset Discovery	Scenario S10 : Special Purpose Asset Discovery Package: Resolver
Registry	Scenario S5 : Direct Download use case for Batch Data Conversion Scenario S10 : Special Purpose Asset Discovery Package: Resolver
Converter	Scenario S5 : Direct Download use case for Batch Data Conversion

	Scenario S6 : Direct Access Use Case for Runtime Data/Message Conversion Scenario S8 : Automatic converter building Use case Scenario S9: Fast Adaptation to Peaks Use case
Mapping IDE	Scenario S7: Automated Mapping Process for the Conversion use case
Resolver	Scenario S10 : Special Purpose Asset Discovery Package: Resolver

3.2.1 Direct Download use case for Batch Data Conversion

Scenario S5

Description

Table 7 Scenario S5: Direct Download use case for Batch Data Conversion

Actor	**S_BusTravel (TSP):** A Service Provider
Target Component/Sub-system/Entity	Converter
Description	This scenario describes Converter discovery and deployment (direct access) for Batch-Data Conversion
Story	**S_BusTravel** is interested to publish a relatively huge set of their data which is in Standard-A to a representation/data model compatible with the target consumers' systems and standards, say Standard-B. Sara is an IT engineer in **S_BusTravel**. She searches within the IF to find out an A-B converter. The IF returns two different Converters with the ability to convert Standard A to B. One of them is a service which is offered by a famous transport operator **TO.** The other one is developed by some startup company **Best_Travel** and it exposed as JAR file which could be download and run locally by **S_BusTravel**. Based on the reputation of **TO**, Sara has more trust in the effectiveness of their Converter service versus the Converter of **SC**. She collects the URL to call the **TO** converter service from the discovery result.

	The rest of the process is outside of the boundary of IF

Activity Diagram

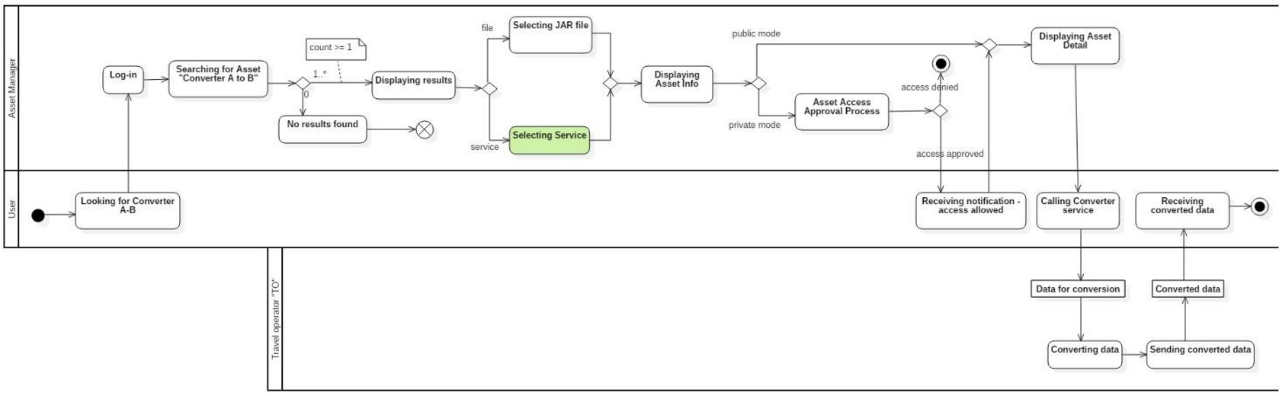

Figure 5 Activity Diagram of Direct Download use case for Batch Data Conversion

Demo/Deployment platform description

This scenario shows usage of the Asset Manager. To demonstrate such scenario, we will use the same Asset Manager instance which will be used to demonstrate previous scenarios. The user will login Web interface provided by the Asset Manager as Service Consumer (= role in AM).

Users search for specific Converter in the asset list. It allows to user to focus on assets of interest by specifying a set of search filters (derived from the available metadata).

If the authorisation policy of the selected asset will be “open access” (= public mode), then the user will view all the metadata information (complete detail) about the asset and will also have access to all its links/data attachments.

If otherwise (= private mode asset) the authorisation policy of the asset will be based on user request and on access approval. After approving the access, the user is notified and can display detail info about an asset. The user can then read the specification/instruction of usage of a provided service in the Asset detail.

If the selected asset has deployment type as Service (which is the case for this scenario), interaction of a user with IF would be terminated after the discovery phase explained above. The user then would reach out to the endpoint of the conversion service and rest of the conversion process would be handled in the premises of conversion service provider. The user sends data in batch to the provided conversion service according to published specification. The user receives converted data. In case that these converted data should be published as the Asset in AM, then the user has to log in as Service Provider (= role in AM) and insert this “data” asset into the AM.

3.2.2 Direct Access Use Case for Runtime Data/Message Conversion

Scenario S6

Description

Table 8 Scenario S6: Direct Access Use Case for Runtime Data/Message Conversion

Actor	**BE-Service (TrSP):** an offer building service for land (rail, bus, etc.) travels within central Europe. Its front-end API is used by mobile and web applications and its backend has access to, and, engaged with many train/buses operators in the covered zones.
Target Component/Sub-system/Entity	Converter
Description	This scenario describes Converter discovery and deployment (Direct download and then local deployment by consumer) for Runtime Data/Message Conversion
Story	Various smartphone and web applications that are providing means for end-users to search and book tickets for train and bus within Central Europe rely on **BE-Service.** **BE-Service** endpoints receive discovery and booking request from transport applications and return them a list of the available itinerary offered by various transport operator for the requested path. Upon to request of user (through the application), it initiates the booking procedure by forwarding user's request to the ticket provider and completes the booking procedure. Accordingly, the format, specification and standardisation of the booking process differs based on the provider operator. **BE-Service** hence required to convert the source booking request/confirmation format to the target model – and vice versa – instantly at runtime. Bob, the IT engineer in **BE-Service,** searches within IF to find out desired converters. In specific he is looking for A-B converter, A-C converter, and M-C converter. He starts by searching A-B converter. IF returns two different Converters with ability to convert Standard-A to Standard-B. One of them is a service which is offered by a famous transport operator **TO.** The other one is developed by some start-up company **SC** and it exposed as JAR file which could be download and run

	locally. To make such JAR far accessible, **SC** has been uploaded it in the IF repository. Since such message conversion is highly frequent and it is part of a live and runtime transaction, Bob prefers to find some way to integrate such mechanism inside its business logic. Hence, he decides to use the converter provided by **SC.** He downloads the JAR file from IF repository and starts engaging with it accordingly (this process is outside of the boundary of IF). Bob initiates another search for A-C converter and repeats the same procedure.

Activity Diagram

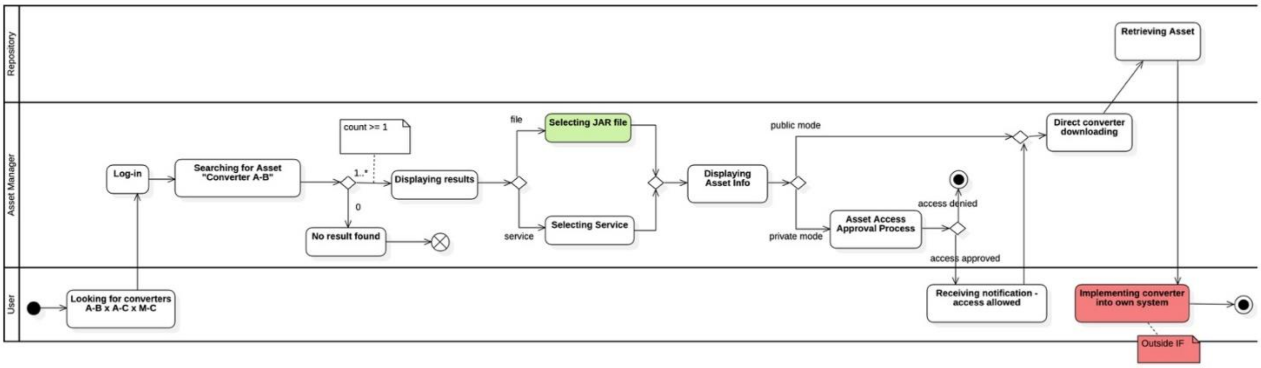

Figure 6 Activity Diagram for Direct Access Use Case for Runtime Data/Message Conversion

Demo/Deployment platform description

This scenario shows usage of the Asset Manager. To demonstrate such scenario, we will use the same Asset Manager instance which will be used to demonstrate previous scenarios. The user will login Web interface provided by the Asset Manager as Service Consumer (= role in AM).

Users search for specific Converter in the asset list. It allows to user to focus on assets of interest by specifying a set of search filters (derived from the available metadata).

If the authorisation policy of the selected asset will be “open access” (= public mode), then the user will view all the metadata information (complete detail) about the asset and will also have access to all its links/data attachments.

If otherwise (= private mode asset) the authorisation policy of the asset will be based on user request and on access approval. After approving of access user is notified and can display detail information about an asset. The user can download file which has been previously uploaded to asset manager by the asset provider.

This part of the scenario is outside the IF: The user implements the downloaded converter inside their company system. The conversion of data is done on the side of Service Consumer.

In case that these new services should be published as the Asset in AM, then the user has to log in as Service Provider (= role in AM) and insert this "service" asset into the AM.

3.2.3 Automated Mapping Process for the Conversion use case

Scenario S7

Description

Table 9 Scenario S7: Automated Mapping Process for the Conversion use case

Actor	**Best_Travel (ISA):** A service/application provider
Target Component/Sub-system/Entity	Converter: Mapping IDE
Description	This scenario describes utilisation of Mapping IDE.
Story	According to **Best_Travel** analysis, the Standard-K is becoming more and more popular and widely used that can substitute the other famous by legacy Standard-P. So, they decide to develop a K-P converter and publish it in the market for potential users. To this end, they need to the stablish the "Mapping" between the concepts and terms in both standards which are used to create the annotations as part of conversion mechanism. John is a specialist in the transport domain standardisation in. **Best_Travel** who is part of the team for developing the converter. His role to create the mapping. Mary is another member of the team who is IT engineer and knows about the Mapping utilities of the IF. She has already run the mapping utilities in some local docker container by utilising its docker image from the IF. By initialising the program, it asks the directory address to a structured representation of both source and target standards. After successful uploading standards, it starts the process. When the process is terminated, the output represents a list of the concepts in source format and the suggested equivalent concepts the target standards.

	John then goes through those suggestions and either confirm or reject them. The program produces the mapping between all the confirmed concepts as final output.

Activity Diagram

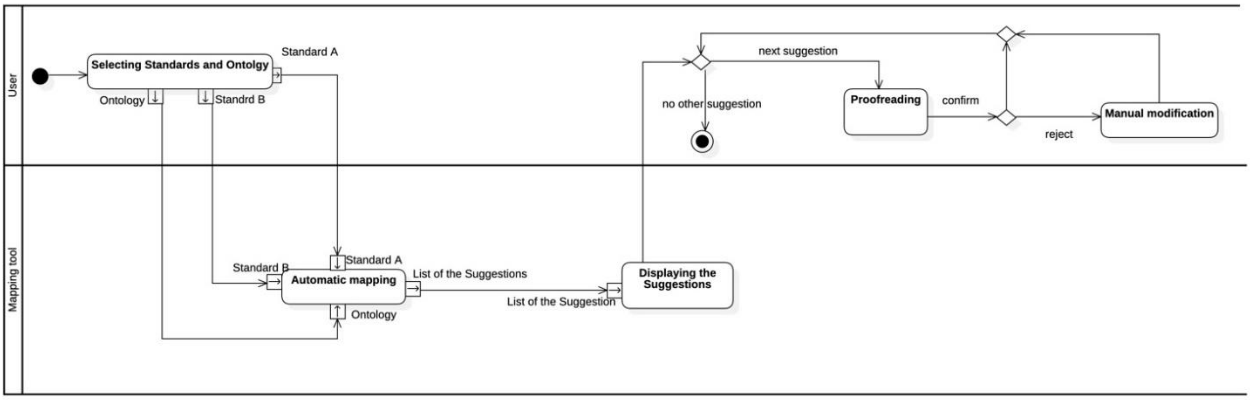

Figure 7 Automated Mapping Process Activity Diagram

Demo/Deployment platform description

The C-REL demonstration for automated mapping is mainly concentrated on the proof of concept for the general idea behind this technique, which is an automated mapping process that is achieved through a machine-learning approach to extract the semantically similar concepts in different standardisations. The C-REL prototype of the system is built based on a Word-2-Vec algorithm and a generic pre-trained model based on the Google News Data Set. Furthermore, the experiment is concentrated on two specific standards (IT2Rail and Transmodel) and it excludes the graphical user interfaces for viewing, manipulation and acceptance/rejection of the mapping suggestions; And leaves these advanced user-friendliness features for the F-REL demonstration. Hence, C-REL version of the mapping IDE will demonstrate the process of taking IT2Rail and Transmodel specifications as the input source and target model, all the essential pre-processing and analysis of the text to prepare them for the actual learning and similarity calculation process, and a text-based output for the suggested mapping between these two standards.

3.2.4 Automatic converter building Use case

Scenario S8

Description

Table 10 Scenario S8: Automatic converter building Use case

Actor	**N-rail:** a rail *TSP* which just joined the Shift2Rail ecosystem.

	Y-bus and X-bus: bus *TSPs* already part of the Shift2Rail ecosystem.
Target Component/Sub-system/Entity	Asset Manager, Converter
Description	A new operator is interested in establishing a new business by communicating with other operators who already joined the IF ecosystem. Since those operators are compliant with the Shift2Rail Ontology, he just needs to provide the mapping between the messages used by his IT systems and the reference ontology. The Asset Manager will then be able to assemble a Converter, composing the different mapping and the required ontologies and data sets. Such Converter will be then used by the operator to effectively connect his system to the ones provided by the other operators.
Story	Y-bus services joined the S2R ecosystem and contributed a Converter to let its clients interact with X-bus, an allied bus operator. It does so by providing a mapping which "lifts" its own data model to the Shift2Rail ontology, and also a mapping which "lowers" instances of the S2R ontology to the X-bus data model.

Activity Diagram

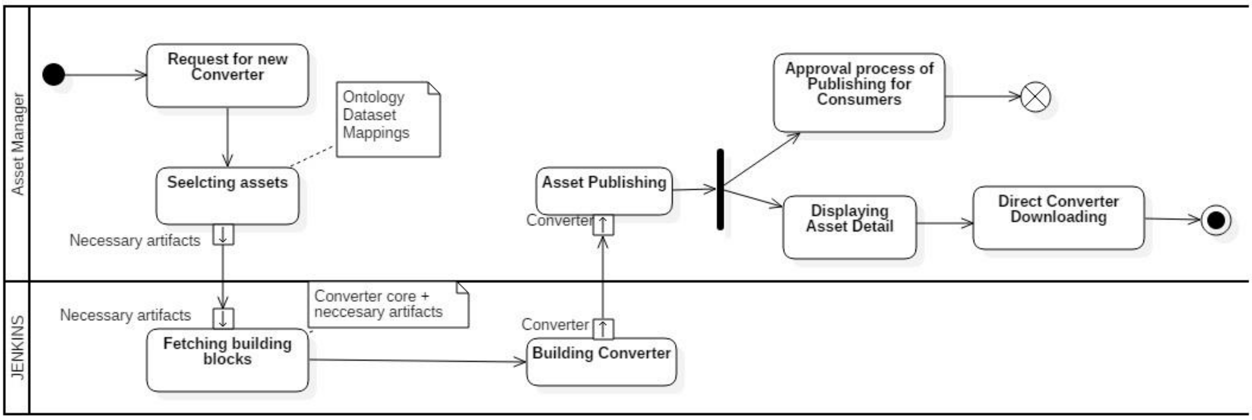

Figure 8 Automatic converter building

Demo/Deployment platform description

The demonstration of this scenario will be implemented using the Asset Manager. Via the AM Publisher Web interface, the user will start publishing the Converter. The user interface will let the user choose the reference ontology, the additional datasets and the mappings which will be required to perform the conversion. Once the "Converter asset" will then be approved using the lifecycle management feature, the Asset Manager will fetch all the required resources, plus a base library containing the SPRINT Converter building blocks, and generate a proper configuration for the Converter. All the “ingredients” will be then packaged as a single JAR archive, which will be added as an attachment to the asset description. The attachment will be then visible both in the Publisher and in the Store Web interfaces of the Asset Manager, and the user will be able to download it.

To prepare the demonstration, the Asset Manager will be configured to host several asset types:

- Ontology
- RDF Dataset
- Mappings
- Converter

The lifecycle management process will take into account that a series of post-publishing steps need to be performed to fetch the required resources, create the Converter configuration and package it.

3.2.5 Fast Adaptation to Peaks Use case

Scenario S9

Description

Table 11 Scenario S9: Fast Adaptation to Peaks Use case

Actor	**BE-Service:** Booking Engine for land (rail, bus, etc.) travels within central parts of Europe. Its front-end API is used by mobile and web applications (say T-A-1 to T-A-10) and its back-end has access to, and, engaged with many train/bus operators (say T- O-1 to T-O-20) in the covered zones.
Target Component/Sub-system/Entity	Converter, Asset Manager
Description	The infrastructure managing the converters deployed by BE-Service to interact with its partner operators need to dynamically adapt to the load. BE-Service needs to quickly replicate Converters, possibly in a cloud environment, to adapt the infrastructure and avoid denial of service.
Story	One of the cities covered by BE-Service is hosting a huge music event, and BE-Service expects a surge of booking request. BE-Service, therefore, needs to cope with two different scenarios: prepare for the first wave of requests to reach the city, and then to cope with mass requests to reach the music event before its start and to reach the homes and hotels after its end.

Activity Diagram

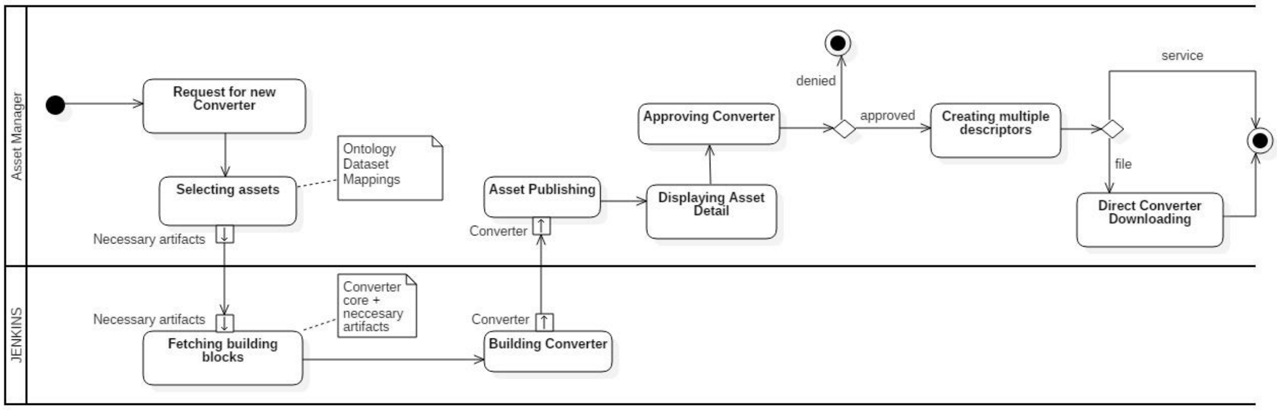

Figure 9 Fast Adaptation to Peaks

Demo/Deployment platform description

BE-Service is hosting a Converter to let other companies access its booking service. The Converter architecture must be conceived so that a Converter will be self-contained and with minimal dependencies. Scaling a Converter must not require configuring anything but the Load Balancer used to route the requests to the different service replicas. To ensure self-containment and fast replication of Converter instances, when possible the Converter should work as a stateless service. This way the Load Balancer will be free to efficiently route the requests to the least loaded instance.

The Converter package must be compatible with technical systems used to implement autoscaling, like Docker Compose, Docker Swarm or Kubernetes. Since the creation of the Converter package can be managed by the Asset Manager, the Asset Manager can create multiple "packaging descriptors" to support easy deployment and replication. In the C-REL release, the Converter will be packaged with a Docker Compose descriptor which will enable scaling it on a single machine. The descriptor will deploy a load balancer and a variable number of Converter containers. The user will then use standard Compose commands ("scale") to upscale the service deployment or to downscale it. Triggering the scale command in C-Rel will not be driven by automatic monitoring and will be completely user-driven.

F-Rel version of the Converter package will deliver a solution to exploit the capabilities of the underlying cloud orchestrator, thus providing recovery from crashes, and automatic up and down-scaling driven by cloud monitoring. In F-REL we will also explore the possibility to let the Asset Manager deploy the Converter on a dedicated cloud runtime environment. To implement this feature we will produce another deployment descriptor which will enable autoscaling and high availability on the selected cloud orchestration technology.

3.2.6 Special Purpose Asset Discovery Package: Resolver

Scenario S10

Description

Table 12 Scenario S10: Special Purpose Asset Discovery Package : Resolver

Actor	Travel Service Provider and/or Public Authority
Target Component/Sub-system/Entity	Resolver, Asset Discovery, Registry, Converter
Description	Special-purpose Asset Discovery components, or Resolvers, are packaged as deployable units and used to perform discovery/retrieve of specific categories of resources such a Locations or Travel Expert services. These Resolvers can be deployed equally internally to the Interoperability Framework, or in any external runtime environment, e.g. at the Travel Service Provider. In this scenario, Resolvers are deployed externally to the Interoperability Framework and are used by a Travel Service Provider or Public Authority in the execution of a shopping/booking or trip tracking process, where there is a need to identify and access resources that are unknown to the requesting application at runtime. These resources may be the geographical coordinates of some Point Of Interest, the Stop Places closest to these geographical coordinates, the web service interface specification of a remote system that can compute offers for an itinerary starting and ending at specified Stop Places, or of a remote system that can perform bookings for an offer.
Story	Requesting Actor, e.g. Travel Service Provider application needs access to resources that may be distributed over the network and unknown to it at runtime. To locate and get access to these resources: 1. The requesting Actor calls the service interface of special-purpose Resolver component with a specific query 2. The Resolver validates and analyses the query 3. If the query is valid, it is passed to the Asset Discovery component for processing. The Process Request activity may use the Distributed SPARQL endpoint to access assets semantic annotations meta-data to determine the nature of the assets being requested a. If data assets are being requested, such as Stop Places or geographical coordinates, the Asset Discovery initiates the Retrieve Data Assets activity in the Registry through a call to the registry's interface.

	b. If a web service interface is being requested, such as a Travel Expert or Booking Engine, the Asset Discovery initiates the Retrieve Service Descriptor activity in the Registry through a call to the registry's interface. 4. Data Assets or Service Descriptors obtained in steps 3a or 3b, respectively, are associated with their format specification. This specification is used to retrieve a specialised semantic converter from the Registry in the case where this is needed by the requestor Actor to map the data asset or service descriptor to a target different specification 5. The Resolver then builds a response to be returned to the requestor Actor. The response contains the requested data asset or the service descriptor, depending on the specific request, and the electronic link to the associated convertor where applicable. This link may be used by the requestor to access the convertor to be used with the returned data asset or service descriptor 6. The response thus build is returned to the requestor Actor

Activity Diagram

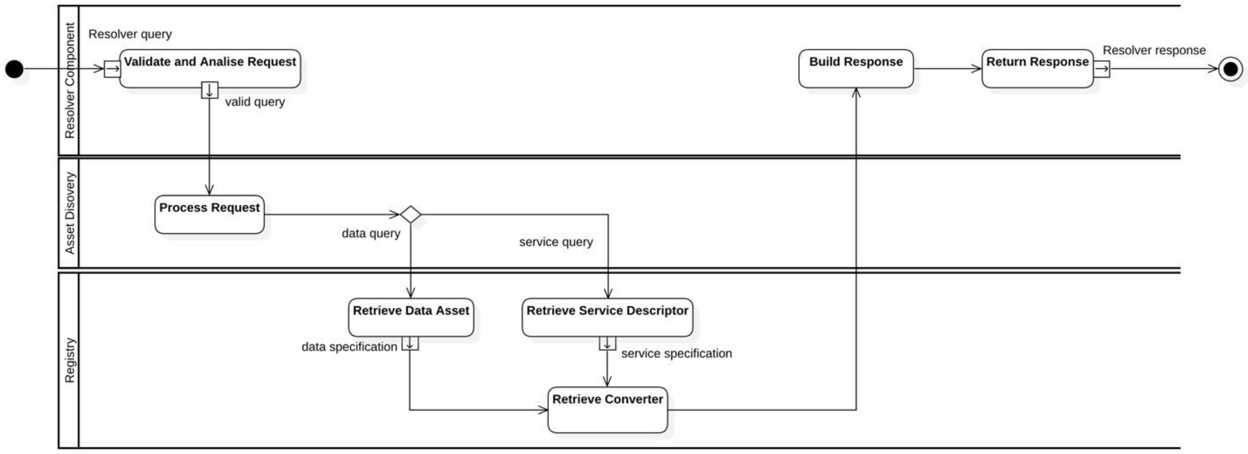

Figure 10 Activity Diagram for Special Purpose Asset Discovery Package: Resolver

Demo/Deployment platform description

The following Resolvers are available as packaged deployable units delivered by the IT2RAIL project[1]

1. The **Location Resolver** service provides data discovery, query and aggregation about transport "infrastructures" such as Airports, Bus Stops, etc.

[1] See http://www.it2rail.eu/ for IT2RAIL's project description and deliverables

2. The **Travel Expert Resolver** service provides discovery from the Service Registry of Travel Expert services exposed by Travel Service Providers that can generate Offer Items for a given Route during the Shopping process.
3. The **Locations Identification** service provides geographic coordinates of specified Points of Interest.
4. The **Resolve Events Source** service provides Stop Places and Transportation Services for use with the Navitia platform[2] to identify disruptions to booked itineraries.
5. The **NeTEx Producer** service implements a subset of the NeTEx 1.03 producer web service specification to return Stop Places in a bounding box in the NeTEx 1.03 format.

For the C-REL implementation of the SPRINT project each available resolver's configuration file will be updated to redirect the resolver's calls to the new SPRINT Asset Manager's SPARQL and Registry endpoints.

The reconfigured Resolvers will be redeployed to the still available IT2RAIL project's demonstration runtime environment, implementing the scenario whereby Resolvers execute externally to the Interoperability Framework while the assets are kept under the control of the Interoperability Framework's Asset Manager.

The deployed Resolvers will be registered to in the Asset Manager so that their deployment to the IT2RAIL demonstration scenario can itself be discovered by Users through the SPRINT Asset Manager's asset discovery mechanism.

[2] The Navitia platform (https://www.navitia.io/) provides application programming interfaces for access to Public Transport data sets and application services covering approximately 15.000 cities and 1.600 networks in 25 countries who participate in an "open data" approach to the implementation of Public Transport applications

4. REFERENCES

[1] SPRINT PROJECT, "D2.2 REQUIREMENTS FOR AN IF ARCHITECTURAL DESIGN (C-REL)," 2019. [Online]. Available: http://sprint-transport.eu/download.aspx?id=7b3a7122-16f5-43e6-9a4f-f749c24e1592.

[2] ST4RT Consortium, "www.st4rt.eu," [Online].