

SEMANTICS FOR PERFORMANT AND SCALABLE INTEROPERABILITY OF MULTIMODAL TRANSPORT

D5.4 –Requirements, Scenarios and Use cases for The Proof-of-Concept (F-REL)

Due date of deliverable: 30/06/2020

Actual submission date: 26/11/2020

Leader/Responsible of this Deliverable: POLIMI

Reviewed: Y

Document status		
Revision	Date	Description
1.0	26/08/2020	Version for TMC approval
2.0	02/09/2020	Final version after TMC approval and Quality check
3.0	19/11/2020	Revised version following JU comments for TMC approval
4.0	26/11/2020	Final version after 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 describes the consolidated scenarios that will be implemented in the F-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.

The scenarios presented in this deliverable update and extend those described in Deliverable D5.1 – “Requirements, scenarios and use cases for the proof of concept (C-REL)”, based on the consolidated user stories reported in Deliverable D.3.4 – “Requirements Analysis and Design of Architecture, Testing Infrastructure, Test Cases and Benchmarks of the IF (F-REL)”.

As in D5.1, The F-REL demonstrators are presented as Use Case Scenarios that describe the interaction between users and one or more components of the Interoperability Framework, to illustrate the relevant features/functions of the latter. More precisely, each scenario description includes:

- An informal description of the scenario, including involved actors and IF components and some hints about the user’s requirements and motivations to use such features.
- A UML activity diagram detailing the steps of the scenario.
- A description of the deployment platform required to carry out the scenario.

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

Executive Summary	2
Abbreviations and Acronyms	3
Table of Contents.....	4
List of Figures	5
List of Tables	6
1. Introduction	7
2. Proof of concept for IF use cases.....	8
2.1 Updated Scenarios for f-rel	8
2.1.1 Scenario S1	8
2.1.2 Scenario S2.....	9
2.1.3 Scenario S3.....	11
2.1.4 Scenario S4.....	12
2.1.5 Scenario S5.....	14
2.1.6 Scenario S6.....	15
2.1.7 Scenario S7.....	18
2.1.8 Scenario S8.....	20
2.1.9 Scenario S9.....	22
2.1.10 Scenario S10.....	24
2.2 New Scenarios.....	27
2.2.1 Scenario S11	27
2.2.2 Scenario S12.....	28
2.2.3 Scenario S13.....	30
2.2.4 Scenario S14.....	31
3. References	34

LIST OF FIGURES

Figure 1 - User Registration Activity Diagram.....	9
Figure 2 - Asset Discovery Activity Diagram.....	11
Figure 3 - Activity Diagram of Direct Download use case for Batch Data Conversion.....	15
Figure 4 - Activity Diagram for Direct Access Use Case for Runtime Data/Message Conversion..	17
Figure 5 - Automated Mapping Process Activity Diagram.....	19
Figure 6 - Automatic converter building.....	21
Figure 7 - Fast Adaptation to Peaks.....	23
Figure 8 - Activity Diagram for Special Purpose Asset Discovery Package: Resolver	26
Figure 9 - Activity Diagram for (Collaborative) Ontology Manager.....	28
Figure 10 - Activity Diagram for Ontology Creation using non-ontological sources.....	29
Figure 11 - Activity diagram for Asset Manager as National Access Points aggregator use case..	30
Figure 12 - Activity diagram for Asset Manager as a tool for contributing to a National Access Point use case.....	32

LIST OF TABLES

Table 1 Scenario S1: Joining the IF Use case (User Registration)	8
Table 2 Scenario S2: Joining the IF Use case (Provider Registration)	10
Table 3 Scenario S3: Service/Asset Discovery (Simple Discovery)	11
Table 4 Scenario S4: Distributed service/asset discovery	12
Table 5 Scenario S5: Direct Download use case for Batch Data Conversion	14
Table 6 Scenario S6: Direct Access Use Case for Runtime Data/Message Conversion.....	16
Table 7 Scenario S7: Automated Mapping Process for the Conversion use case	18
Table 8 Scenario S8: Automatic converter building Use case	20
Table 9 Scenario S9: Fast Adaptation to Peaks Use case	22
Table 10 Scenario S10: Special Purpose Asset Discovery Package : Resolver	24
Table 11 Scenario S11: (Collaborative) Ontology Manager.....	27
Table 12 Scenario S12: Ontology Creation using non-ontological sources	28
Table 13 Scenario S13: Asset Manager as National Access Points aggregator	30
Table 14 Scenario S14: Asset Manager as a tool for contributing to a National Access Point	31

1. INTRODUCTION

This deliverable describes the final SPRINT demonstrator's scenarios that have been derived through a continual process starting from the early phase of the project. In particular, we have defined an initial set of User Stories in Deliverable D3.2 - "Performance and scalability requirements for the IF (C-REL)", that steered the main IF design objectives and showed the challenges and requirements for the development of the IF. Those user stories then constituted the foundation of the proof-of-concept demonstration of the SPRINT project firstly defined in Deliverable D5.1 – "Requirements, scenarios and use cases for the proof of concept". The actual implementation and validation of those demonstrators have been reported in deliverables D5.2 – "Software release of the proof-of-concept in its technical environment (C-REL)" and D5.3 – "Validation of pilot implementation (C-REL)", respectively.

The validation results of the pilot implementations, as well as the advancement of the IF architecture (with respect to the early design presented in D3.1), led us to introduce in Deliverable D.3.4–"Requirements Analysis and Design of Architecture, Testing Infrastructure, Test Cases and Benchmarks of the IF (F-REL)" a few new User Stories with respect to the set described in D3.2, and to enhance some of the already defined ones.

Accordingly, the current deliverable presents the scenarios that complement and extend those in D5.1 based on the modifications of their corresponding User Stories in D3.4, and it introduces additional scenarios for the User Stories newly defined in D3.4. Similar to the proof-of-concept demonstrators, the F-REL demonstrators are presented in the form of Use Case Scenarios that define a particular course of action and interaction between users and one or more IF components, so as to illustrate some features/functionality of involved component(s). More precisely, each scenario section is composed of:

- An informal description of the scenario, including involved actors and IF components and some hints about the user's requirements and motivations to use such features.
- A UML activity diagram detailing the steps of the scenario.
- A description of the deployment platform required to carry out the scenario.

2. PROOF OF CONCEPT FOR IF USE CASES

To avoid repetition, in this chapter we only present the updated or newly defined scenarios in section 2.1 and 2.2 respectively.

2.1 UPDATED SCENARIOS FOR F-REL

2.1.1 Scenario S1

With the reference to the Scenario S1 in Deliverable 5.1, we present the updated scenario for F-REL implementation.

Description

Table 1 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	<p>Bill is an employee in HT-train which is responsible to register this operator to IF.</p> <p>He goes to IF website of Italy [IF].it and selects to register to IF as Service Provider Role. He is redirected to the IF Identity Provider, which manages identity across all IF components.</p> <p>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.</p> <p>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.</p>

Activity Diagram

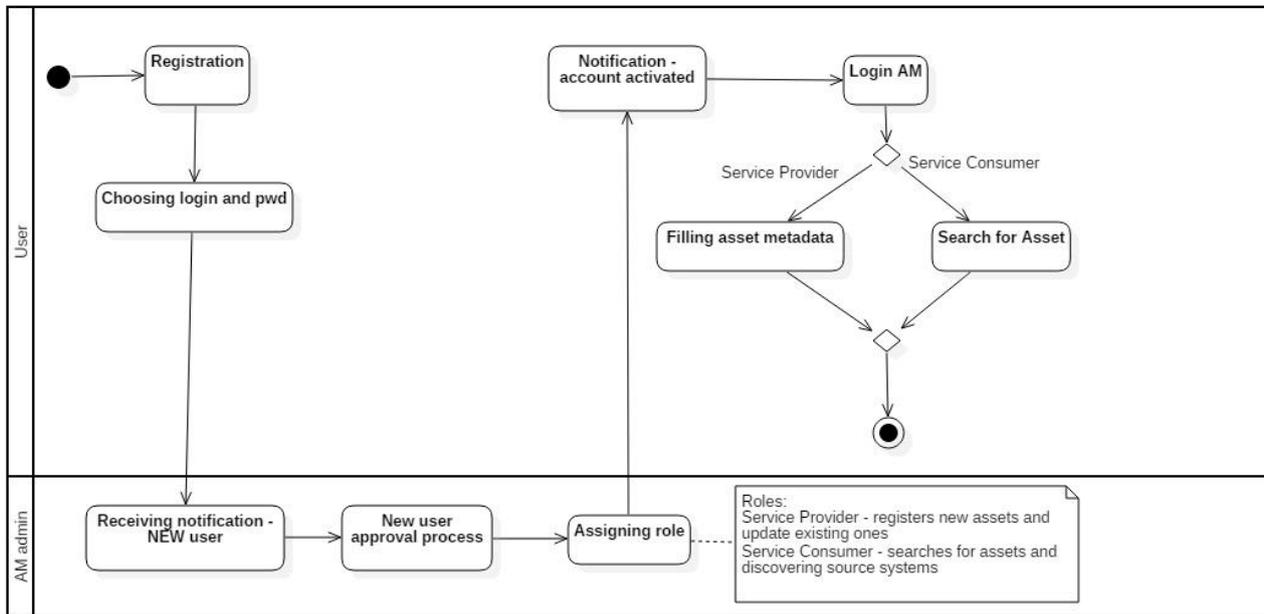


Figure 1 - User Registration Activity Diagram

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 registration process will be managed through the Identity Provider deployed by CONNECTIVE on their platform. Once the user will be registered, he will be redirected to the Asset Manager, where he will continue his journey with asset search and asset metadata publishing.

2.1.2 Scenario S2

With the reference to the Scenario S2 in Deliverable 5.1, we present the updated scenario for F-REL implementation.

Description

Table 2 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	<p>Alice is an employee in SafeTravel company that develops a smartphone application for ticket search and booking.</p> <p>Alice is responsible to register this transport application to IF. She goes to IF website and she is redirected to the IF Identity Provider, which manages identity across all IF components.</p> <p>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. She selects to register to IF as Service Consumer Role.</p> <p>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.</p>

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 together with the Identity Provider. To demonstrate such scenario, we will use the same Asset Manager instance which will be used to demonstrate Scenario 1, and the same Identity Provider deployed by CONNECTIVE on their own platform. As in Scenario 1, the user will register using the Identity Provider. 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.

2.1.3 Scenario S3

With the reference to the Scenario S3 in Deliverable 5.1, we present the updated scenario for F-REL implementation.

Description

Table 3 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

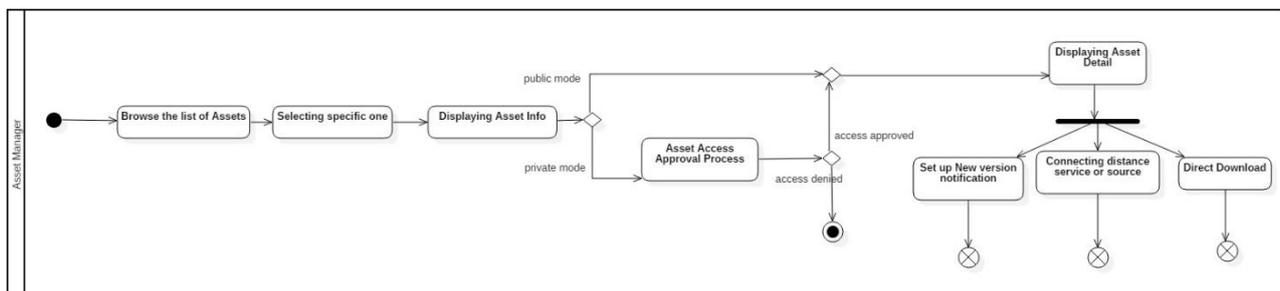


Figure 2 - Asset Discovery Activity Diagram

Demo/Deployment platform description

This scenario is unchanged with respect to the one described in D5.1 and validated in D5.3. We report here the scenario description from D.5.1 for completeness.

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 authorization 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 authorization 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.

2.1.4 Scenario S4

With the reference to the Scenario 4 in Deliverable 5.1, we present the updated scenario for F-REL implementation in which the Distributed SPARQL endpoint evaluates a set of SPARQL queries over multiple SPARQL endpoints.

Description

Table 4 Scenario S4: Distributed service/asset discovery

Actor	<p>B-Com: A Belgian-based TSP which it is hosted by the (National Access Point) NAP of Belgium</p> <p>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.</p> <p>V-com₁, V-com₂, ..., V-com₁₀: TSPs which it is hosted by 10 (National Access Point) NAPs</p> <p>MyMobility: An Italian company providing transport services in many regions of Europe</p>
Target Component/Sub-system/Entity	Distributed SPARQL endpoint
Description	This scenario illustrates querying the SPARQL engine on several endpoints
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, Spain and 10 other countries to expand its routes services, we treat S-Com, B-Com, V-com1, V-com2, ..., V-com10 as twelve different data sources, whose resources can be combined to find metadata catalogues of the NAP datasets. For this, IF needs to make use of the services provided by B-Com, S-Com, V-com1, V-com2, ..., and V-com10 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.

Also, MyMobility wants to further filter its results by using preferences. For example, MyMobility can specify preferences on update frequency and creation date of the datasets belonging to the catalogues in order to find those datasets that were initially updated frequently but are now stable because they are no longer updated, i.e., the most frequently updated datasets (qual_freq) with the oldest last modified time (metadata_date).

Activity Diagram

This diagram corresponds to the scenario S3 considering asset discovering is over a distributed environment (reference to D5.1).

Demo/Deployment platform description

For the F-REL demonstration, a set of federated SPARQL queries (with and without preferences) can be executed by a distributed SPARQL endpoint like Ontario. As a proof of concept, the distributed SPARQL endpoint will access twelve different RDF repositories which host different datasets and whose resources will be combined. Ontario will execute the federated query on RDF repositories and combine resources from 12 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.

2.1.5 Scenario S5

With the reference to the Scenario S5 in Deliverable 5.1, we present the updated scenario for F-REL implementation.

Description

Table 5 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	<p>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.</p> <p>Sara is an IT engineer in S_BusTravel. She searches within the IF to find out an A-B converter.</p> <p>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.</p> <p>Based on the reputation of TO, Sara has more trust in the effectiveness of their Converter service versus the Converter of SC.</p> <p>She collects the URL to call the TO converter service from the discovery result.</p> <p>The rest of the process is outside of the boundary of IF</p>

Activity Diagram

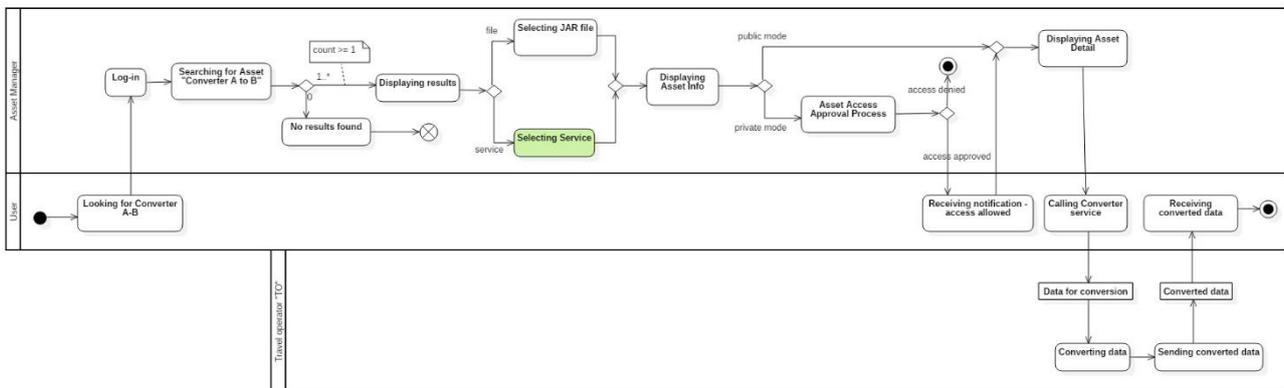


Figure 3 - Activity Diagram of Direct Download use case for Batch Data Conversion

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 using the Identity Provider deployed and configured by CONNECTIVE and will then be redirected to the Asset Manager.

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.

2.1.6 Scenario S6

With the reference to the Scenario S6 in Deliverable 5.1, we present the updated scenario for F-REL implementation.

Description

Table 6 Scenario S6: Direct Access Use Case for Runtime Data/Message Conversion

<p>Actor</p>	<p>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.</p>
<p>Target Component/Sub-system/Entity</p>	<p>Converter</p>
<p>Description</p>	<p>This scenario describes Converter discovery and deployment (Direct download and then local deployment by consumer) for Runtime Data/Message Conversion</p>
<p>Story</p>	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>He starts by searching A-B converter.</p> <p>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.</p> <p>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</p>

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

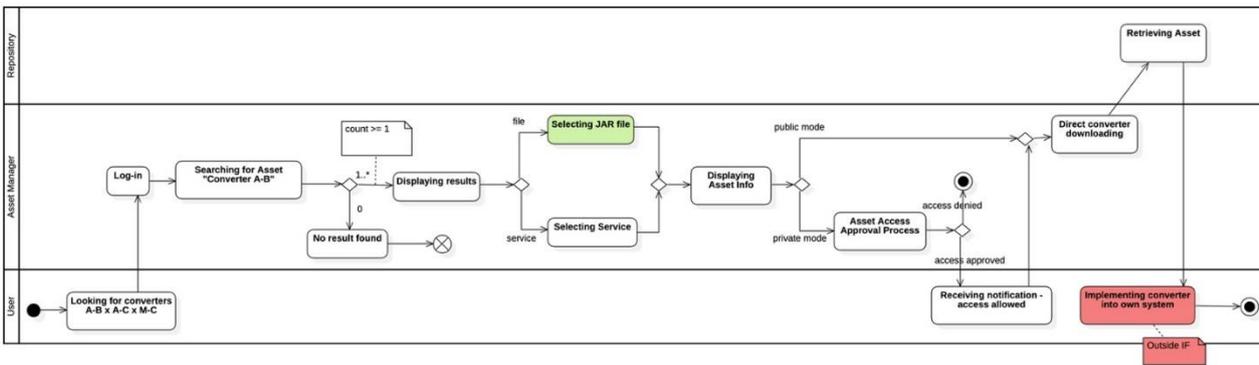


Figure 4 - Activity Diagram for Direct Access Use Case for Runtime Data/Message Conversion

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 using the Identity Provider deployed and configured by CONNECTIVE and will then be redirected to the Asset Manager.

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 the software artifact which has been previously uploaded to Asset Manager by the asset provider.

The user then deploys and run the downloaded converter inside the system of his company (this part of the scenario is outside the IF).

2.1.7 Scenario S7

With the reference to the Scenario S7 in Deliverable 5.1, we present the updated scenario for F-REL implementation.

Description

Table 7 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	<p>According to Best_Travel analysis, the Standard-K is becoming more and more popular and widely used that can substitute the other famous but 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 generate the “Mapping” between the concepts and terms in both standards, as well as, the Java annotations.</p> <p>John is an ontology engineer and specialist in the transport domain standardization in Best_Travel who is part of the team for developing the converter.</p> <p>Mary is a member of the team in Best_Travel organization, who is a software engineer and knows the Mapping Tool of the IF. She has already run the tool utilizing its docker image.</p> <p>By initializing the tool, she goes through a wizard to provide the program with the required resources to start the process. To this end, she selects the format of source and target file (XML, OWL, ttl, etc) and afterward, she can select the files.</p> <p>Default outputs are the Java annotated files (which are then the required inputs for Converter components) but a user can configure the tool to receive the actual mappings (one-to-one translation of terms from source to target standard) as well.</p> <p>After the successful uploading of the standards, the tool starts the mapping process and notifies user upon the termination of the job. At this point, Mary views a list of the concepts in source format and the suggested equivalent concepts to the target standards which are ranked based on the probability of being similar.</p>

So, Mary, with the help of John, can go through suggestions, where the top-ranked suggestion is considered as the confirmed mapping by default. Nevertheless, they then can select another suggestion as the confirmed one, or, manually add a term if the desired term is not among the suggestions.

When they are done with reviewing all the mappings, the tool proceeds with the generation of the final outputs which are the mappings and annotated java files. The final outputs, then, would be stored in the desired directory specified by the user.

Activity Diagram

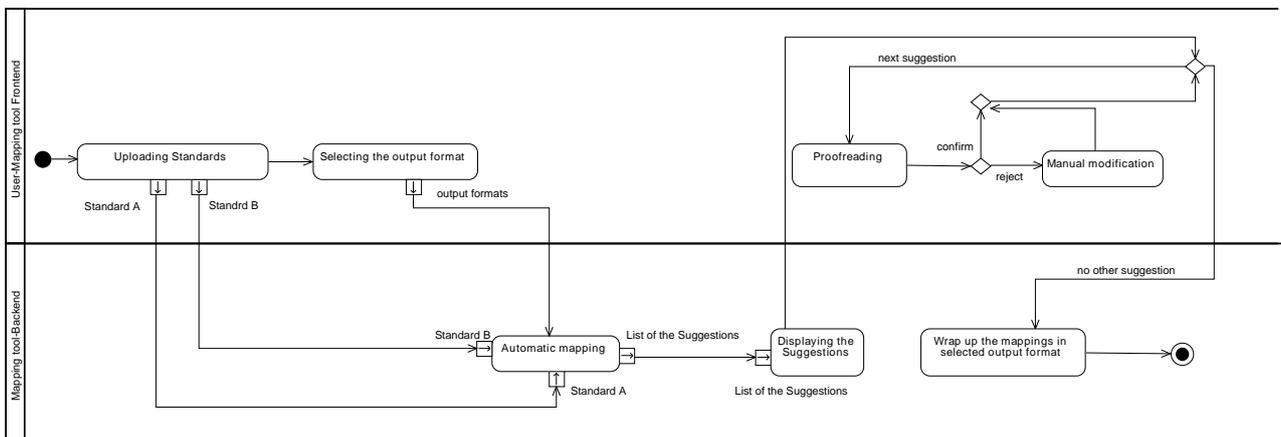


Figure 5 - Automated Mapping Process Activity Diagram

Demo/Deployment platform description

The F-REL demonstration for Mapping Tool that offers an automated mechanism that utilizes both the syntactical structures of two standards and the linguistic semantics behind the terms/concepts of each standard to drive a set of mappings. In particular, there are three possible outputs that users can opt to get one or any of them: **I)** a raw set of mapping, called “Mappings”, which is a set of one-to-one translation of concepts in source and target standards (For the complete explanation of the algorithm of Mapping generation please refer to SPRINT deliverables D.4.2 and D.4.3). **II)** The “Java annotations” to be used for Annotation-based Conversion in the Converter Component. **III)** The “RML Mappings” to be used for RML-based Conversion in the Converter Component. The last two outputs are various materializations of lifting and lowering block in Converter component, where the former follows the java annotation technology to represent the equivalent concepts in two standards, and latter uses the RML language for this purpose (Please See SPRINT deliverable D.3.3 and D.5.2 for more information about Converters and materialization of the conversion blocks).

The F-REL version of the Mapping Tool is containerized and distributed as a stand-alone docker image. The tool is a web-based application that is intended to be run on the local server of the user system. The server management, however, is hidden from the user and such configurations are encapsulated within the docker image. Hence, users can follow the common standard docker approach/commands to pull the docker image from the docker hub, install and run it. Upon running the docker image, the tool is accessible in a predefined port in the localhost, and the user can engage with the provided GUIs and go through the steps indicated in the above scenario via any browser.

2.1.8 Scenario S8

With the reference to the Scenario S8 in Deliverable 5.1, we present the updated scenario for F-REL implementation.

Description

Table 8 Scenario S8: Automatic converter building Use case

Actor	N-rail: a rail <u>TSP</u> which just joined the Shift2Rail ecosystem. Y-bus and X-bus: bus <u>TSPs</u> 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. Whenever one of the assets required by the Converter is updated, Y-Bus is notified. This allows the publisher TSP to evaluate whether a new Converter release is required to adapt to the new situation.

Activity Diagram

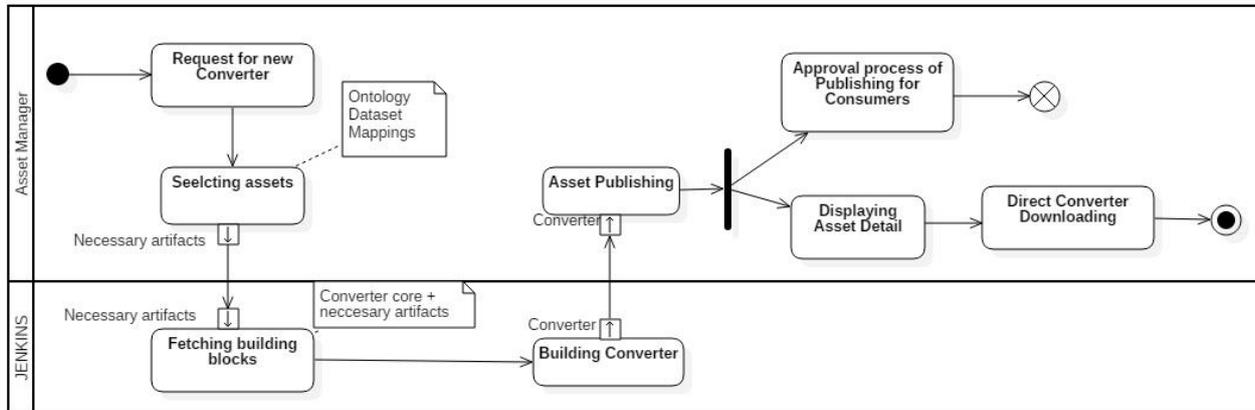


Figure 6 - Automatic converter building

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 Asset Manager will then package the JAR file together with other relevant configuration files, and will also offer the user the possibility to download a Docker Compose archive and a Kubernetes configuration archive. The attachments 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 them.

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. Whenever an Ontology, an RDF Dataset or a Mapping will be updated, a notification will be triggered in the lifecycle management process to warn the Converter owner about the change. This will allow the Converter owner to evaluate whether he will need to re-release his Converter including the updated dependencies.

2.1.9 Scenario S9

With the reference to the Scenario S9 in Deliverable 5.1, we present the updated scenario for F-REL implementation.

Description

Table 9 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

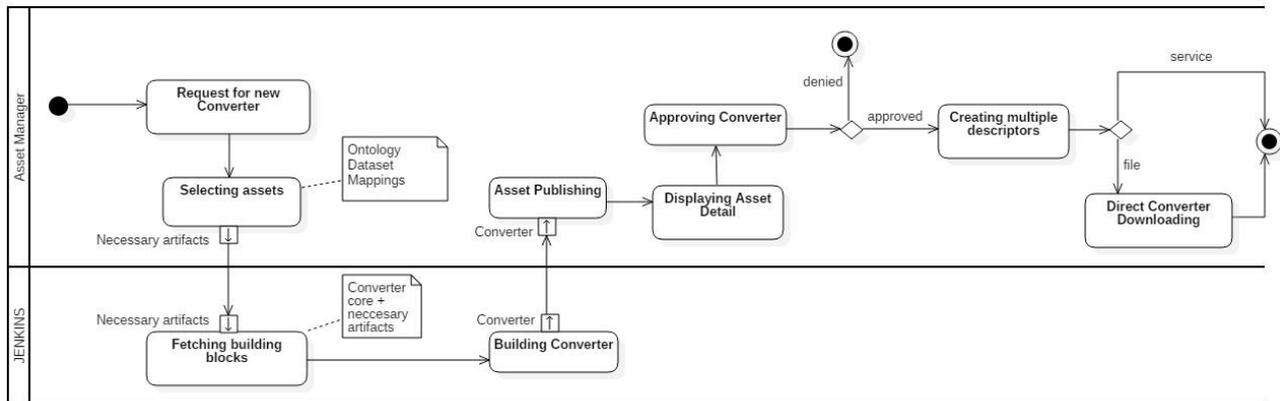


Figure 7 - Fast Adaptation to Peaks

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.

This scenario relies on the automatic creation of Converter artifacts described in Scenario 8. The Asset Manager will create multiple archives and attach them to the Converter asset. The available attachments will be:

- A JAR file with no external dependencies to be used to test the Converter from the command line interface. This artifact will not offer scalability features.
- A Docker Compose package, which will enable obtaining multiple instances of the Converter will contain:
 - the previously described JAR archive;
 - a Dockerfile to create the Docker image of the Converter;
 - an Nginx configuration to let it act as a load balancer;
 - a Docker Compose configuration to enable deploying the Converter and the Load Balancer in a way that the number of Converter instances can change according to the administrator request.

- A Kubernetes package that enables the definition of a Converter Service distributing requests over multiple Pod instances. In this case the Kubernetes platform will take care of providing a unique endpoint for the Converter Service load balancing requests among the different instances. Therefore, the downloadable artifact will contain the JAR file, the Dockerfile and the Kubernetes Service configuration files (manifests).

Triggering the up/down scaling of the Docker Compose containers will be user-driven. The provided Kubernetes configuration will enable the possibility of controlling manually the number of replicas, but also to define a Horizontal Pod Autoscaler within the cluster to automatically control the number of running instances. The Horizontal Pod Autoscaler (HPA), relies on the Kubernetes *metrics-server*, that needs to be deployed in the cluster, and by default can be configured considering the CPU utilization of Pods. The demonstration of this scenario will showcase an example of an auto-scaling policy configured for the Converter Service taking into account the CPU utilization of instances running on a local Kubernetes cluster. Moreover, it will showcase how the Converter can be configured to expose custom metrics on a particular endpoint, which is the basis for the definition of custom autoscaling policies. The implementation of the custom autoscaling solution will be left to the user since it heavily depends on the specific application metrics collection solution deployed on the specific Kubernetes cluster¹.

2.1.10 Scenario S10

With the reference to the Scenario S10 in Deliverable 5.1, we present the updated scenario for F-REL implementation.

Description

Table 10 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

¹ Support for custom metrics is currently provided by “adapter” API servers offered by metrics solution vendors. There are some known solutions (<https://github.com/kubernetes/metrics/blob/master/IMPLEMENTATIONS.md#custom-metrics-api>), but none of those implementations are officially part of Kubernetes.

	<p>Interoperability Framework, or in any external runtime environment, e.g. at the Travel Service Provider.</p> <p>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.</p>
<p>Story</p>	<p>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:</p> <ol style="list-style-type: none"> 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 <ol style="list-style-type: none"> 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 call a dynamic Converter, which will locate the relevant ontologies, datasets and mappings inside the Asset Manager, 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.

	<p>This link may be used by the requestor to access the convertor to be used with the returned data asset or service descriptor</p> <p>6. The response thus build is returned to the requestor Actor</p>
--	--

Activity Diagram

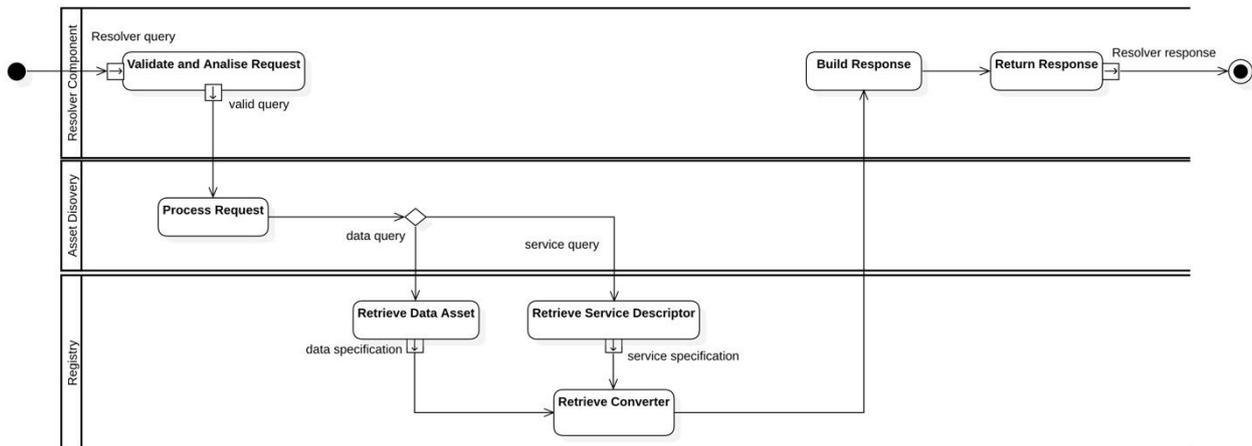


Figure 8 - Activity Diagram for Special Purpose Asset Discovery Package: Resolver

Demo/Deployment platform description

The S10 Scenario which will be demonstrated in F-Rel will focus on the dynamic retrieval of assets by a Converter. The dynamic Converter will contain the logic to query the Asset Discovery according to its input and output parameters, which will dictate the input format/standard/specification and the desired output format/standard/specification. Using such parameters, the Converter will perform an Asset Discovery request, looking for mappings and their related ontologies and RDF datasets. Once obtained from the AM, the Converter will perform an RML-based lifting, and a template-based Apache Velocity lowering, returning the obtained results to the requester.

In the demo, the dynamic Converter will have its own credentials to access the Asset Manager, and we will assume that the Mappings will have the granularity of whole standard/specifications. It will be therefore possible to ask for a “GTFS to LinkedGTFS” mapping without worrying whether such mapping will contain the specific part which will be required to perform lifting or lowering on the specific input message. Another assumption will be that the input message will not require any “preprocessing”, and that RML will be enough to obtain a proper lifting.

The demo will demonstrate that the SPRINT solutions are flexible enough to cover many scenarios. Whereas the automatic building of the Converter covers the “static compilation”

of an optimized and specific Converter, this case will illustrate that leveraging on SPRINT Chimera framework we will be able to create a generic, format agnostic Converter. Such Converter which will use the Asset Manager as a registry to dynamically discover how to solve a specific conversion process.

2.2 NEW SCENARIOS

2.2.1 Scenario S11

With the reference to the User Story SU-9, in D.3.4 here we define the respected use case scenario for F-REL implementation.

Description

Table 11 Scenario S11: (Collaborative) Ontology Manager

Actor	EU-Tram: A European TSP, in specific a tram service provider that aims to represent its data semantically.
Target Component/Sub-system/Entity	OnToology
Description	This scenario illustrates validation and documentation during the collaborative ontology development lifecycle
Story	EU-Tram is interested in generating, extending and reviewing a set of ontologies that have developed to semantically represent its data. As part of the feedback process during the ontology development lifecycle, EU-Tram proposes its ontology developers to use OnToology in the generation and revision of its ontologies. To build each ontology, developers must interact with domain experts that are not familiar with ontologies. Thus, these experts need to easily understand vocabulary involved in the transport domain by means of diagrams in order to elicit their requirements. After that, they can provide feedback to the ontology developers which will extend and improve the already created ontologies and re-start the documentation process. In addition, the ontology developers must evaluate the quality of their generated ontologies by means of OnToology, and identify and fix potential errors before publishing the ontology.

Activity Diagram

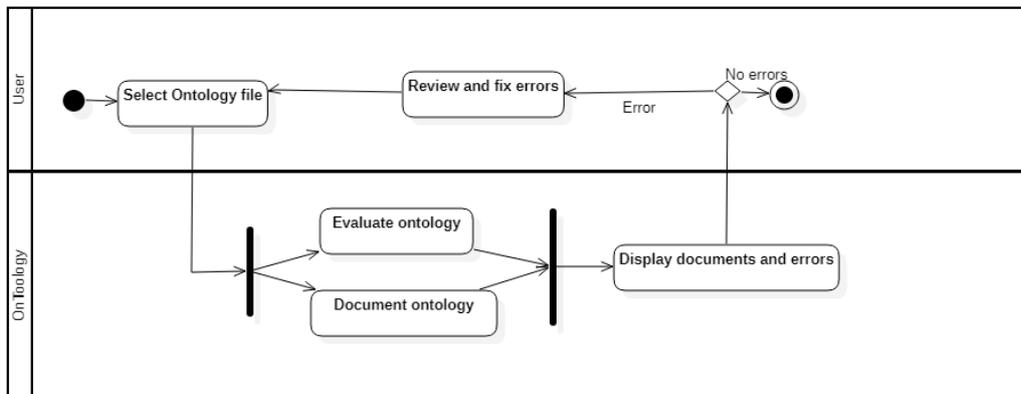


Figure 9 - Activity Diagram for (Collaborative) Ontology Manager

Demo/Deployment platform description

For the F-REL demonstration of the Collaborative Ontology Manager, we will use test ontologies where we will be able to visualize human-readable documentation that will allow users to understand the ontological file produced after the specification of the ontology requirements and the design of the ontology properties and classes. Also, the ontologies will be evaluated by checking if the requirements are answered properly and if they follow design patterns and well-established practices for their implementation.

2.2.2 Scenario S12

With the reference to the User Story SU-8, in D.3.4 here we define the respected use case scenario for F-REL implementation.

Description

Table 12 Scenario S12: Ontology Creation using non-ontological sources

Actor	EU-Bus: A European TSP, in specific a bus service provider
Target Component/Sub-system/Entity	XSD2Ontology
Description	This scenario illustrates an ontology creation using non-ontological sources
Story	EU-Bus needs to register itself as a company with the IF node but its information follows the XML format designed for the efficient, updateable exchange of complex transport data among distributed systems. For not starting from scratch, EU-Bus proposed to transform its XML documents

	to ontologies by means of automatization of the ontology creation. The Ontology generation will be done using XML instances and validating XSD files. The generated ontology can be added to our transport ecosystem.
--	---

Activity Diagram

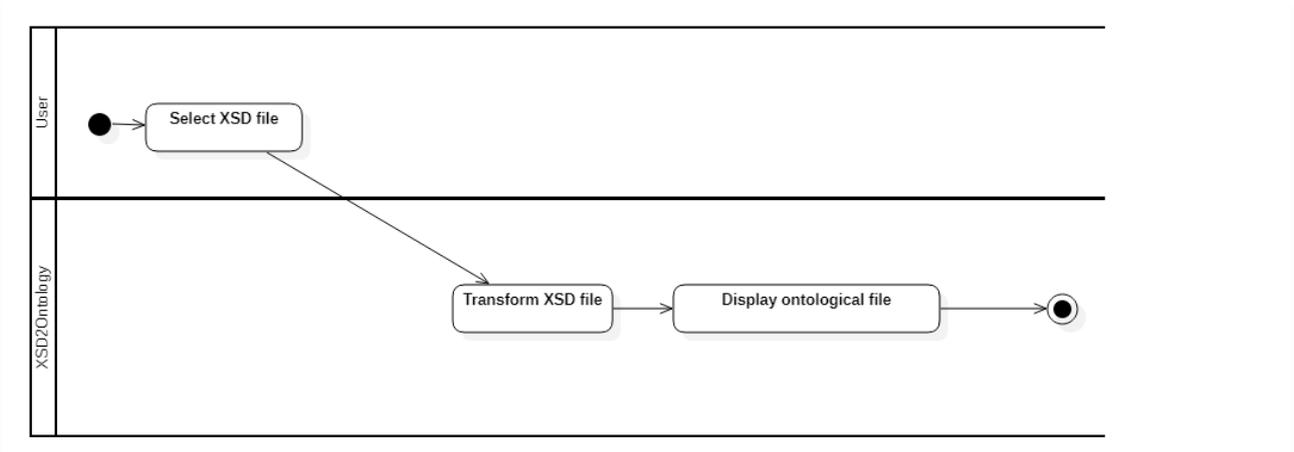


Figure 10 - Activity Diagram for Ontology Creation using non-ontological sources

Demo/Deployment platform description

For the F-Rel demonstration, we will automatically transform from NeTeX XSD files to RDF/OWL. We have selected Netex because it is a standard for the exchange of timetable data related to public transport. Since NeTeX -based ontologies have been generated manually, we will be able to compare the result of our transformation with the ontologies that are currently being created with respect to Transmodel Ontology.

2.2.3 Scenario S13

With the reference to the User Story SU-12, in D.3.4 here we define the corresponding use case scenario for F-REL implementation.

Description

Table 13 Scenario S13: Asset Manager as National Access Points aggregator

Actor	EuroTrain, a rail TSP focused on cross-border travels.
Target Component/Sub-system/Entity	Asset Manager, Converter
Description	This scenario demonstrates the usage of the Asset Manager as an aggregator of National Access Points.
Story	EuroTrain is planning a new cross-border service connecting France and Netherlands through Belgium. They need to access the Journey planning data to analyse the competitors' offers and create their offer. Since their planned service crosses three EU countries, they would need to access three different National Access Points. Instead, they deploy and configure an IF Asset Manager to let it aggregate the metadata coming from those countries, and get access to the desired assets using a single interface.

Activity Diagram

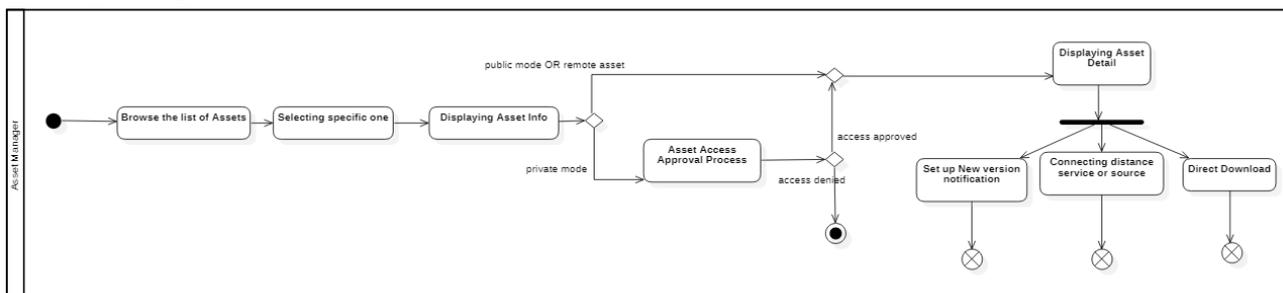


Figure 11 - Activity diagram for Asset Manager as National Access Points aggregator use case

Demo/Deployment platform description

This scenario demonstrates the usage of the Asset Manager as an aggregator of National Access Points. It will require performing many activities using the tools offered by the SPRINT IF:

- creation of mappings between the metadata used in different NAPs and a common RDF metadata structure (based on the Coordinated Metadata Catalogue metadata described in D2.3);
- creation of special Converters which will fetch NAP metadata and execute such mappings;
- extension of the Store and Publisher UI to be able to list both local and remote assets;
- creation of an Exploration API to allow users to perform Asset Discovery for both local and remote assets.

For this demonstration, we will access Journey planning metadata (describing mostly GTFS datasets).

2.2.4 Scenario S14

With the reference to the User Story SU-13, in D.3.4 here we define the corresponding use case scenario for F-REL implementation.

Description

Table 14 Scenario S14: Asset Manager as a tool for contributing to a National Access Point

Actor	EuroTrain, a rail TSP focused on cross-border travels.
Target Component/Sub-system/Entity	Asset Manager, Converter
Description	This scenario demonstrates the usage of the Asset Manager as a tool for contributing to National Access Points.
Story	Following the activities related to S13, EuroTrain has planned his new cross-border service connecting France and Netherlands through Belgium. According to the NAP regulations, they must contribute their Journey Planning data to the NAP of each country. They publish their GTFS dataset just once, informing the Asset Manager that the dataset is related to a cross-border operator and including the names of the countries the dataset is related to. The Asset Manager takes care of activating the publication processes required by each NAP, and in case a conversion is required performs data conversion using an existing Converter.

Activity Diagram

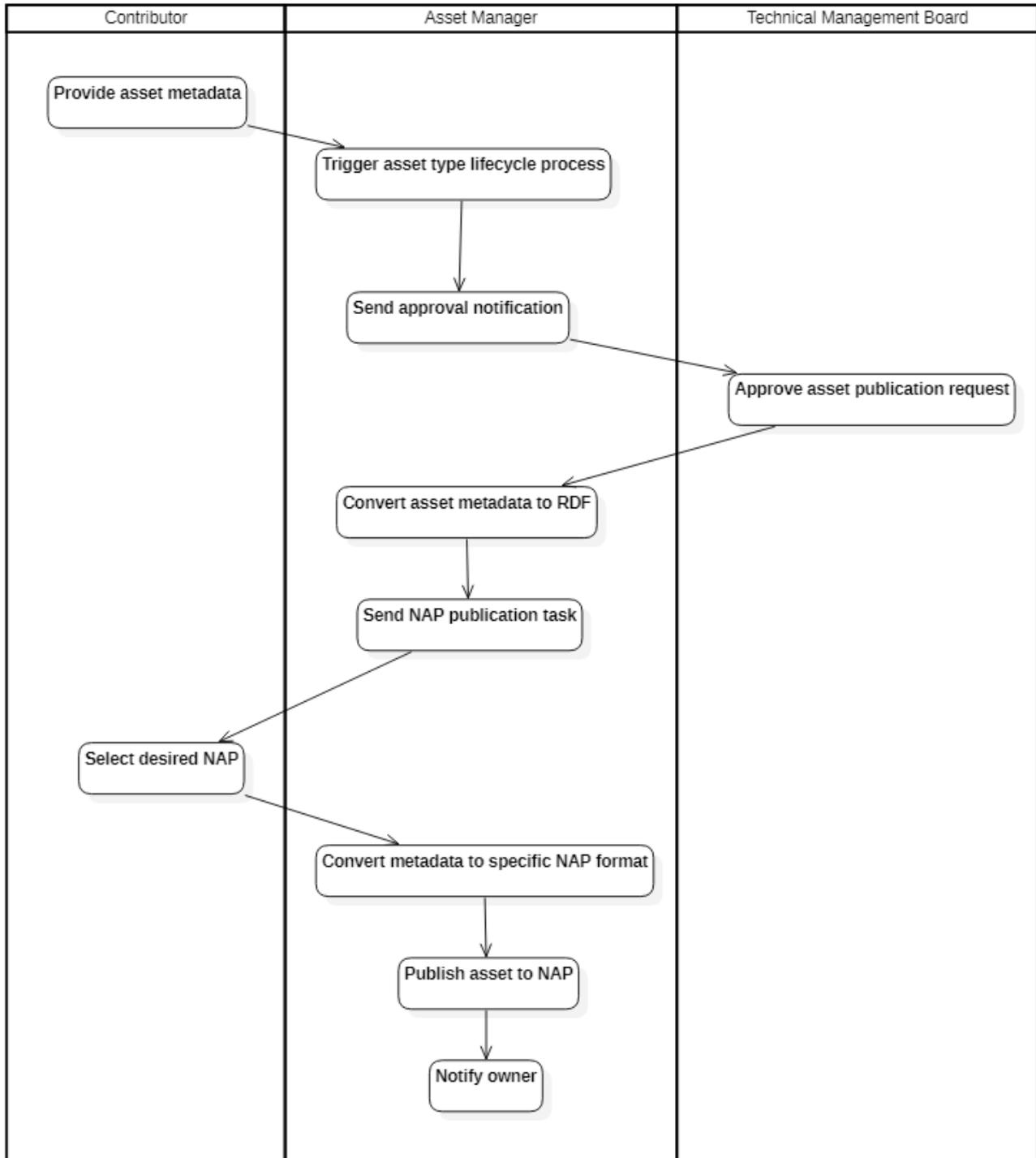


Figure 12 - Activity diagram for Asset Manager as a tool for contributing to a National Access Point use case

Demo/Deployment platform description

This scenario demonstrates the usage of the Asset Manager as a contributor to one or more National Access Points. It will require performing many activities using the tools offered by the SPRINT IF:

- capture the information relating the dataset to multiple EU countries with the asset publication form;
- find Converters which are able to convert GTFS datasets into the format mandated by the specific NAP;
- perform calls to external publication services. Since each NAP has its own publication process, dedicated publication routes must be established.

For this demonstration, we will access Journey planning metadata (describing mostly GTFS datasets).

3. 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].