



## Towards a Shared European Logistics Intelligent Information Space

### D3.7 SELIS Nodes Connectors



This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No 690588.

## Document Summary Information

<b>Grant Agreement No</b>	690558	<b>Acronym</b>	SELIS
<b>Full Title</b>	Towards a Shared European Logistics Intelligent Information Space		
<b>Start Date</b>	01/09/2016	<b>Duration</b>	36 months
<b>Project URL</b>	www.selisproject.eu		
<b>Deliverable</b>	D3.7 SELIS Nodes Connectors		
<b>Work Package</b>	WP3 Logistics Information Communications and Navigation Connectivity Sub-system		
<b>Contractual due date</b>	31/08/2019	<b>Actual submission date</b>	05/09/2019
<b>Nature</b>	ORDP: Open Research Data Pilot	<b>Dissemination Level</b>	Public
<b>Lead Beneficiary</b>	CLMS UK LIMITED		
<b>Responsible Author</b>	Dr Yannis Zorgios		
<b>Contributions from</b>	AVT, DU, EBOS, IBM, ICCS, SON, VLTN, WAYZ		

### Revision history (including peer reviewing & quality control)

Version	Issue Date	Stage	Changes	Contributor(s)
0.1	18/07/2019	Structure the outline	Basic structure of document	Dr Yannis Zorgios Antonis Mygiakis
0.2	25/07/2019	First Draft	Added sections 2 and 3	Dr Yannis Zorgios Antonis Mygiakis Kyriakos Filippakos Orfeas Panagou
0.3	05/08/2019	Writing	Added description for RailData adapter	Dr Yannis Zorgios Kyriakos Filippakos
0.4	13/08/2019	Candidate version for review	Enriched section 3 and added conclusions and executive summary	Dr Yannis Zorgios Antonis Mygiakis Kyriakos Filippakos Orfeas Panagou
0.5	29/08/2019	Version ready for review	Added links to the open source repository and a description of the template. Enriched conclusions.	Dr Yannis Zorgios Antonis Mygiakis
0.6	04/09/2019	Addressing Reviewer Comments	Added figure with usage example of SELIS Node Connector, added contributors, added table	Dr Yannis Zorgios Antonis Mygiakis Orfeas Panagou

Version	Issue Date	Stage	Changes	Contributor(s)
			addressing SELIS DoA, changed language to equate SELIS Node Connectors to Adapters	

---

## Executive Summary

The SELIS Project aims to deliver a cloud-based “platform for pan-European logistics applications” which will work as “a network of logistic communities' specific shared intelligent information spaces termed SELIS Community Nodes (SCNs)”.

T3.3 aims to develop interfaces for connecting data sources to the content-based routing infrastructure. In particular the deliverable at hand, D3.7 aims at including a set of SELIS Nodes connectors (as per D3.5 and D3.6), integration libraries and guidelines (contribution to the Open Research Data Pilot).

More specifically, the connectivity interfaces utilized within this Open Research scheme include the following core categories: (a) Generic client library, (b) Connectors to SC Visibility, (c) Connectors for IoT devices, (d) Mobile connections to external navigational systems, and (e) Navigation and routing information as data providers - Navigation and routing information as a service. The content-based routing infrastructure is a core component of the SCN architecture that enables the sharing of information with the usage of pub/sub technologies.

The importance of interoperability among SELIS systems is amplified by the number and variability of the involved systems, using different types of data, different formats and messaging templates, causing syntactic, semantic and pragmatic heterogeneity. The role of the connectivity interfaces in SELIS, is to allow all involved systems to communicate (exchange information) in a seamless and agnostic manner. Therefore, this deliverable describes these generic mechanisms that are needed in order to facilitate this communication, and offers them as open access to research data, following the FAIR data principles.

### *Disclaimer*

The content of the publication herein is the sole responsibility of the publishers and it does not necessarily represent the views expressed by the European Commission or its services.

While the information contained in the documents is believed to be accurate, the authors(s) or any other participant in the SELIS consortium make no warranty of any kind with regard to this material including, but not limited to the implied warranties of merchantability and fitness for a particular purpose.

Neither the SELIS Consortium nor any of its members, their officers, employees or agents shall be responsible or liable in negligence or otherwise howsoever in respect of any inaccuracy or omission herein.

Without derogating from the generality of the foregoing neither the SELIS Consortium nor any of its members, their officers, employees or agents shall be liable for any direct or indirect or consequential loss or damage caused by or arising from any information advice or inaccuracy or omission herein.

### *Copyright message*

© SELIS Consortium, 2016-2019. This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both. Reproduction is authorised provided the source is acknowledged.



This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Grant Agreement No 690588.

## Table of Contents

1	Introduction .....	8
1.1	Addressing the SELIS Description of Action .....	9
1.2	Document outline .....	9
2	Open Data Repositories .....	10
2.1	Overview .....	10
2.2	Community Curation.....	10
2.3	Zenodo .....	11
3	SELIS Node Connectors .....	12
3.1	Overview .....	12
3.2	Depositor Order Adapter .....	14
3.2.1	Import Customers .....	14
3.2.2	Import Products .....	16
3.2.3	Import Depositor Orders.....	18
3.2.4	Import Invoices .....	20
3.3	ThreePL Adapter .....	23
3.3.1	Import Answers.....	23
3.4	PoD Adapter.....	25
3.4.1	Import Proofs of Delivery.....	25
3.5	WMS Adapter.....	26
3.5.1	Import WMS Orders.....	26
3.6	RailData Adapter.....	30
3.6.1	Import Train Schedules .....	30
3.6.2	Import Container Status Events .....	34
3.6.3	Import Wagon Events .....	36
3.7	SELIS Node Connectors in Zenodo .....	40
4	Conclusions .....	41
5	References .....	42

## List of Tables

Table 1: Import Customers Method .....	14
Table 2: Import Products Method .....	16
Table 3: Import Depositor Orders Method .....	18
Table 4: Import Invoices Method.....	20
Table 5: Import Answers Method .....	23
Table 6: Import Proofs of Delivery Method .....	25
Table 7: Import WMS Orders Method .....	26
Table 8: Import Train Schedules Method .....	30
Table 9: Import Container Status Events Method .....	34
Table 10: Import Wagon Events Method.....	36



## Glossary of terms and abbreviations used

Abbreviation / Term	Description
<b>ACCSEAS</b>	Accessibility for Shipping, Efficiency Advantages and Sustainability
<b>AIS</b>	Automatic Identification System
<b>API</b>	Application Programming Interface
<b>ASM</b>	Application Specific Messages
<b>AVT</b>	Avanti Communications Plc
<b>CAPITALS</b>	Collaborative and Adaptive Integrated Transport Across Land And Sea
<b>COTS</b>	Common Off the Shelf
<b>CSV</b>	Comma Separated Value
<b>DB</b>	Database
<b>EGLS</b>	European Green Logistic Strategy
<b>EOL</b>	End of Line
<b>EPD</b>	E-Navigation Prototype Display
<b>ESA</b>	European Space Agency
<b>ETA</b>	Estimated Time of Arrival
<b>EU</b>	European Union
<b>GA</b>	Grant Agreement
<b>GNSS</b>	Global Navigation Satellite System
<b>GUI</b>	Graphical User Interface
<b>HTTP</b>	HyperText Transfer Protocol
<b>IATA</b>	International Air Transport Association
<b>IMO</b>	International Maritime Organization
<b>JMS</b>	Java Messaging Service
<b>JPA</b>	Java Persistence API
<b>JSON</b>	Java Script Object Notation
<b>LLs</b>	Living Labs
<b>PNT</b>	Position Navigation and Timing
<b>POM</b>	Project Object Model
<b>PUBSUB</b>	Publish/Subscribe
<b>REST</b>	Representational State Transfer
<b>RTZ</b>	Route Exchange Format
<b>SC</b>	Supply Chain
<b>SC</b>	Supply Chain
<b>SCN</b>	SELIS Community Node
<b>SCV</b>	Semantic Conflict Values
<b>SIP</b>	Strategy Implementation Plan
<b>SMTP</b>	Simple Message Transfer Protocol
<b>SSO</b>	Single-Sign-On
<b>TCP</b>	Transmission Control Protocol
<b>TLS</b>	Transport Layer Security

<b>UI</b>	User Interface
<b>UK</b>	United Kingdom
<b>URL</b>	Uniform Resource Locator
<b>UTF</b>	Unicode Transformation Format
<b>VHF</b>	Very High Frequency
<b>VIS</b>	Voyage Information Service
<b>VPN</b>	Virtual Private Network
<b>VTS</b>	Vessel Traffic System
<b>WP</b>	Work Package
<b>XML</b>	eXtensible Markup Language

## 1 Introduction

T3.3 aimed to develop interfaces for connecting data sources to the content-based routing infrastructure. Various SELIS Node Connectors and Connectivity Interfaces were developed and are fully described in the two main deliverables of the task, namely D3.5<sup>1</sup> and D3.6<sup>2</sup>. The deliverable at hand, D3.7 briefly describes the SELIS Node Connectors that were chosen to be published as a contribution to the Open Research Data Pilot (ORDP).

More specifically, the connectivity interfaces utilized within this Open Research scheme include the following core categories: (a) Generic client library, (b) Connectors to SC Visibility, (c) Connectors for IoT devices, (d) Mobile connections to external navigational systems, and (e) Navigation and routing information as data providers - Navigation and routing information as a service. The content-based routing infrastructure is a core component of the SCN architecture that enables the sharing of information with the usage of pub/sub technologies.

This deliverable builds on top of and utilizes the outcomes from deliverables D3.5 and D3.6 with respect to the connectivity interfaces of SCNs, in order to provide an Open Research Data Pilot framework for the node connectors of SELIS.

Data Governance deals with data usage, consumption and policies. Complying to the H2020 overall strategy and principles for open access to research data, SELIS follows the FAIR data principles where all research data should be Findable, Accessible, Interoperable and Reusable (FAIR). Towards this end, this approach should be adhered regarding the final SELIS Node Connectors as well.

Appropriate repositories have been established for storing the results of the project with respect to the SELIS node connectors and providing access to the scientific community, such as OpenAIRE<sup>3</sup> and Zenodo<sup>4</sup>.

This deliverable describes the overall methodological approach, an overview of the open data repositories and the Pilot on Open Research Data in Horizon 2020, and how SELIS is facilitating actively towards the Pilot on Open Research Data by opt in regarding providing the relevant connectors and interfaces for connecting data sources to the content-based routing infrastructure.

Figure 1 shows a basic example of how a Node Connector links with an external system for data input and then transforms the aforementioned data into the SELIS Common Exchange Model. The resulting transformed dataset is then published through the PubSub to the SELIS Community Node.

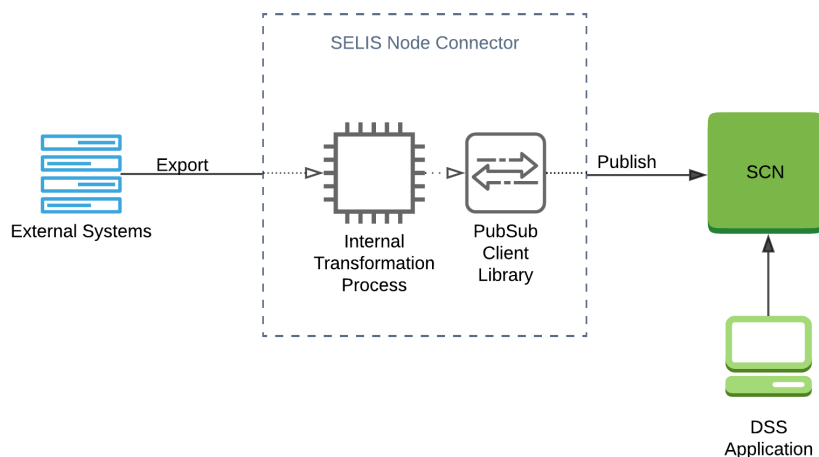


Figure 1 Example usage of a Node connector

<sup>1</sup> D3.5 - Connectivity Interfaces (version 1)

<sup>2</sup> D3.6 - Connectivity Interfaces (final version)

<sup>3</sup> <https://www.openaire.eu/>

<sup>4</sup> <https://zenodo.org/>



## 1.1 Addressing the SELIS Description of Action

SELIS GA requirements	Section(s) of present deliverable addressing SELIS GA	Description
<p><b>D3.7 SELIS Node Connectors</b></p> <p>This deliverable will include a set of SELIS Nodes connectors (as per D3.5 and D3.6), integration libraries and guidelines (contribution to the Open Research Data Pilot).</p>	<p>Section 2, 3</p>	<p>The relevant sub-tasks (ST) featured in Task T3.3: Connectivity Interfaces have been addressed by previous deliverables, namely D3.5 and D3.6, whilst the deliverable at hand aims to cover the contributions of SELIS in the Pilot on Open Research Data in Horizon 2020.</p>

## 1.2 Document outline

**Section 2** provides an overview of the Open Data Repositories concepts, the community curation and a brief summary of Zenodo.

**Section 3** provides a detailed description of the available SELIS adapters.

**Section 4** describes the available online links of SELIS in Zenodo.

**Section 5** provides some concluding remarks with reference to the deliverable.

**Section 6** lists all the references used throughout the document.

## 2 Open Data Repositories

### 2.1 Overview

Valuable information produced by researchers in many projects funded by the European Union (EU) are shared freely as a result of a Pilot on Open Research Data in Horizon 2020, the EU framework program for research and innovation. Researchers in projects participating in the pilot are asked to make the underlying data needed to validate the results presented in scientific publications and other scientific information available for use by other researchers, industries, and citizens. According to the European Commission (EC), this will lead to better and more efficient science and improved transparency for citizens and society. It will also contribute to economic growth through open innovation, the EC said.

The Pilot on Open Research Data in Horizon 2020 aims to improve and maximize access to, and reuse of research data generated by projects for the benefit of society and the economy. To achieve those goals, projects participating in the Pilot shall open their research data on an open access basis. The Open Research Data Pilot applies to two types of data:

- data, including associated metadata, needed to validate the results presented in scientific publications as soon as possible;
- other data, including associated metadata, as specified and within the dead- lines laid down in a data management plan.

Projects participating in the Pilot are:

- required to deposit the research data, preferably in a research data repository, and
- as far as possible, take measures to enable third parties to access, mine, exploit, reproduce and disseminate this research data.

OpenAIRE provides a repository called Zenodo that can be used for depositing data. OpenAIRE provides researcher support and services for the Open Research Data Pilot. Legal barriers to data sharing in the context of the Open Research Data Pilot are analysed here. In the following part of the study, the focus is on the practical implications of European data protection legislation for running the Open Research Data Pilot. The aim is to identify problem areas where data protection law conflicts with the open access obligation of the Open Research Data Pilot. The results are intended to give guidance to projects participating in the Pilot on the question of whether they should opt out of the Pilot for data protection reasons.

Projects may opt out of the pilot to allow for the protection of intellectual property or personal data, in view of security concerns, or should the main objective of their research be compromised by making data openly accessible.

The Pilot will give the EC a better understanding of what supporting infrastructure is needed and of the impact of limiting factors such as data protection. It will also contribute insights into how best to create incentives for researchers to manage and share their research data.

SELIS is facilitating actively towards the Pilot on Open Research Data by opt in regarding providing the relevant connectors and interfaces for connecting data sources to the content-based routing infrastructure.

### 2.2 Community Curation

The preservation and availability of research data is a major concern as it affects core principles of scientific practice, including repeating and contrasting experiment and sharing findings. This concern has resulted in a number of initiatives and services that offer long-term preservation of research data in a broad sense, many of them exposing data and its description in open access form. The key feature of those initiatives is offering mechanisms for the persistent identification and archiving of datasets together with services for sharing them and making them citeable. This later feature allows for using datasets as a complementary source for research output evaluation as suggested elsewhere [1].

Unsurprisingly, the concern for quality arises as in the publication of papers. There exist data journals or journals that require archival of data as associated to articles. However, repositories support a larger variety of use cases, and many of them implement minimal control on the data uploaded by users. This is due to the fact that repositories are intended for a wide or even global user community, and it is not economically feasible to implement any kind of centralized quality control. One of the approaches to non-centralized quality control is that of relying on user communities that organize around collection of resources picked from the repository using in many cases topical or discipline-specific criteria. These communities either explicitly or implicitly carry out some form of quality control, thus becoming *de facto* the quality delegates of the overall repository. The approach has the interesting attribute of being scalable, as it grows with the community of users of the repository and externalizes the work of applying selection criteria. Yoon [2] found that these user communities are one of the factors influencing user *trust* in digital repositories.

The approach of removing controls on update, and then using a social or community approach as a quality control mechanism has been used in other online repositories in the past. A good example is *Connexions*, a learning material repository that implemented a similar approach by means of the so-called *lenses*. The concept of lens is, in the Connexions repository, a mechanism that facilitates to focus on the content of the repository that is good or useful to a given user or community. Therefore, lenses enable both organizations and individuals to give their “stamps of approval” to content in the repository, allowing for user-driven quality control of modules and collections. Kelty, Burrus & Baraniuk [3] describe that approach as a post-publication process. In addition to pointing out to scalability as a property of the approach, they identify additional added values, as “[...] reveal relationships, new contexts of use, and possibilities for reuse that would not be possible if the objects in the repository had a single evaluation by a single reputable source”.

### 2.3 Zenodo

Zenodo (<https://zenodo.org/>) is an online repository hosted at CERN which allows sharing publications and supporting data. Launched in May 2013, the Zenodo repository was specifically designed to help ‘the long tail’ of researchers based at smaller institutions to share results in a wide variety of formats across all fields of science. Some communities are already using Zenodo [4] in their archival workflows, taking benefits also from their integration with the Github platform (<https://github.com/>).

As Zenodo is intended to support individual researchers, it features no mechanisms to control for the data uploaded. In words of one of its creators, Lars Holm Nielsen, “Researchers can upload files to Zenodo and there’s minimal validation of what goes in there, but these community collections essentially allow everyone to create and curate the content and this solves the issue of us otherwise having to validate everything that’s uploaded”. In its few years of existence, a good number of communities have appeared in Zenodo. As Zenodo does not restrict the creation of communities by registered users, their creation and functioning respond only to the will of individuals and communities engaged with the repository. This makes the repository an interesting exemplar of a data curation repository in which researcher behaviour manifests both in the growth and actual use of the repository and also in the selection made by communities.

## 3 SELIS Node Connectors

### 3.1 Overview

The purpose of the SELIS Node Connectors (or adapters for short), is to transform data from various sources (either exported from software or extracted from proprietary data) into a common SELIS model. The main structures used in these adapters are:

1. **Customers:** Static customer data which will be referenced by other structs following the flow of the supply chain
2. **Products:** Static product data which will also be referenced by other structs.
3. **Depositor Orders:** The orders to be fulfilled by the Logistics Service Provider.
4. **Answer to Depositor Order:** The detailed analysis of the make-up of the order (the amount available, which warehouse will it be shipped from etc.) to be fulfilled by a Third-Party Logistics Service Provider.
5. **Invoices:** The invoice of the order, which also can double as a shipping document.
6. **Proof of Delivery:** The final proof of delivery detailing the condition of the shipped order and the confirmation of arrival.
7. **WMS Order:** Orders generated by an external WMS (Warehouse Management System) describing all relevant details such as Origin/Destination, Cargo to be Transported, Customer and Logistics Service Provider contact details, vehicle data etc.
8. **Train Schedules:** Timetables and schedules of train cargo transports with detailed descriptions about stations, cargo etc.
9. **Containers:** Cargo information that references trains, schedules, stations and related events that occur during the transportation
10. **Wagon Events:** Status events that report every change in the transportation process. These events describe both the normal procedure flow as well as any possible disruptions and/or changes to the initial schedule.

The sequencing above is also the recommended order to run the respective functions of the adapters mentioned below.

The adapters are used as command line applications, executing the sum of the methods described with arguments used to specify the file input directories and PubSub metadata such as the SCN (SELIS Community Node) Name and the PubSub URL. All the adapters utilize the open source pub/sub client libraries<sup>5</sup> (included in the SELIS Open Source offering) in order to connect to the relevant pub/sub channels. For each adapter mentioned below, a description of its methods is presented in detail featuring a specific structure (template). More specifically the template presents the following:

- **Method:** The title of the method described.
- **Arguments:** These are the required arguments for the method, the arguments that are requested to be parsed.
- **Synopsis:** This is a brief description of the method, giving an extended reference to its purpose, utilization and need within the system.
- **Input file structure:** These define the input parameters required by the method. Each input parameter is described though a specific format, the name of the parameter, the type and finally the SCV semantic of the parameter.
- **Output Message:** These define the output parameters produced by the method. Each output parameter is described though a specific format, the name of the parameter, the type and finally the SCV semantic of the parameter.

---

<sup>5</sup> As they were described in *D3.6 - Connectivity Interfaces (final version)*

- **Sample Output Message:** This is an example of the final message generated by the adapter. The message contains the output parameters mentioned above encapsulated in the required Pub/Sub semantics.

## 3.2 Depositor Order Adapter

The depositor order adapter is the main adapter used in the Supply Chain Visibility interfaces. It handles the transformation of Customers, Products, Depositor Orders, as well as the final Invoices from a specified directory. After the parsing is complete, the items are transformed to the SELIS SCV data models and then published to the PubSub in JSON format as a .csv message. The following sections describe the individual functions of the adapter.

### 3.2.1 Import Customers

Table 1: Import Customers Method

<b>Method</b>	ImportCustomers			
<b>Arguments</b>	PubsubUrl (string), csvDirectory (string)			
<b>Synopsis (Method description)</b>	The method is responsible for parsing static Customer data from a .csv file contained in the specified csv Directory with specific fields (see input below), transforming them to the SELIS SCV (Supply Chain Visibility) models and publishing them to the requested PubSub url. The method supports parsing multiple files at once.			
<b>Input file structure</b>	<b>Format</b>	<b>Name</b>	<b>Type</b>	<b>SCV Semantic</b>
	<i>Comma Delimited</i>	<i>Kunnr</i>	<i>String</i>	<i>CustomerCode</i>
		<i>Name1</i>	<i>String</i>	<i>CompanyName</i>
		<i>Stcd2</i>	<i>String</i>	<i>VATNo</i>
		<i>Name2</i>	<i>String</i>	<i>Occupation</i>
		<i>Telf1</i>	<i>String</i>	<i>Telephone</i>
		<i>Bztxt</i>	<i>String</i>	<i>Region</i>
		<i>Ort01</i>	<i>String</i>	<i>City</i>
		<i>Stras</i>	<i>String</i>	<i>Address</i>
		<i>Pstlz</i>	<i>String</i>	<i>PostalCode</i>
		<i>Bezel</i>	<i>String</i>	<i>DOY</i>
		<i>Bztx2</i>	<i>String</i>	<i>Region_2</i>
		<i>Ort02</i>	<i>String</i>	<i>City_2</i>
<i>Stra2</i>	<i>String</i>	<i>Address_2</i>		
<i>Pstl2</i>	<i>String</i>	<i>PostalCode_2</i>		

	Format	Name	Type	Comments
<b>Output Message</b>	JSON	CustomerCode	String	Code of Customer
		CompanyName	String	Name of customer's Company
		VATNo	String	Value Added Tax Number
		Occupation	String	Occupation of Customer/Company
		Telephone	String	Telephone of Customer/Company
		Region	String	Region of Customer/Company
		City	String	City of Customer/Company
		Address	String	Address of Customer/Company
		PostalCode	String	Postal Code of Customer/Company
		DOY	String	Identifier of responsible Tax Office for Customer/Company
		Region_2	String	Secondary Region of customer/company
		City_2	String	Secondary City of customer/company
		Address_2	String	Secondary Address of customer/company
		PostalCode_2	String	Secondary Postal Code of Customer/Company
<b>Sample Output Message</b>	<pre> {"message_type": "Customer", "scn_slug": "{{node_name}}",  "payload": "Id;CustomerCode;CompanyName;Occupation;TaxOfficeCode; TaxOfficeName;VATNo;Region;City;Address;PostalCode;DOY;Region_ 2; City_2;Address_2;PostalCode_2;Telephone;Document0;0000200233; *****;*****;*****;*****;***** *****; *****;*.******;*****;***** *****; 13561;*****1159;*****;*****;***** *****;*****;*****;*****;210 2*****;"}" </pre>			

### 3.2.2 Import Products

Table 2: Import Products Method

<b>Method</b>	ImportProducts			
<b>Request Arguments</b>	PubsubUrl (string), csvDirectory (string)			
<b>Synopsis (Method description)</b>	The method is responsible for parsing static Product data from a .csv file contained in the specified csv Directory with specific fields (see input below), transforming them to the SELIS SCV(Supply Chain Visibility) models and publishing them to the requested PubSub url. The method supports parsing multiple files at once.			
<b>Input parameters</b>	<b>Format</b>	<b>Name</b>	<b>Type</b>	<b>SCV Semantic</b>
	<i>Comma Delimited</i>	<i>Matnr</i>	<i>String</i>	<i>Code</i>
		<i>Maktx</i>	<i>String</i>	<i>Description</i>
		<i>Meins</i>	<i>String</i>	<i>BasicUnit</i>
		<i>O</i>	<i>String</i>	<i>ConversionFactor</i>
		<i>Meint</i>	<i>String</i>	<i>SaleUnit</i>
		<i>Brgew</i>	<i>String</i>	<i>GrossWeight</i>
		<i>Ltgew</i>	<i>Decimal</i>	<i>Volume</i>
		<i>O</i>	-	<i>Not Used</i>
		<i>Eantm</i>	<i>String</i>	<i>EAN_Tem</i>
		<i>Eankb</i>	<i>String</i>	<i>Ean_Kib</i>
		<i>T</i>	<i>String</i>	<i>Type</i>
		<i>B</i>	-	<i>Not Used</i>
		<i>Tlif</i>	<i>String</i>	<i>LifeDuration</i>
<i>Warn</i>		<i>String</i>	<i>Warning</i>	
<i>Stop</i>	-	<i>Not Used</i>		
<b>Output</b>	<b>Format</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>



<b>parameters</b>	<i>JSON</i>	<i>Code</i>	<i>String</i>	<i>Code of Product</i>
		<i>Description</i>	<i>String</i>	<i>Description of Product</i>
		<i>BasicUnit</i>	<i>String</i>	<i>Basic Unit of Product</i>
		<i>ConversionFactor</i>	<i>String</i>	<i>The conversion factor of the product</i>
		<i>SaleUnit</i>	<i>String</i>	<i>Unit in which product is sold</i>
		<i>GrossWeight</i>	<i>Decimal</i>	<i>GrossWeight of 1 unit of product</i>
		<i>Volume</i>	<i>Decimal</i>	<i>Volume of 1 unit of product</i>
		<i>Ean_Tem</i>	<i>String</i>	<i>Ean_Tem</i>
		<i>Ean_Kib</i>	<i>String</i>	<i>Ean_Kib</i>
		<i>Type</i>	<i>String</i>	<i>Type of product</i>
		<i>LifeDuration</i>	<i>String</i>	<i>Life Duration of product</i>
		<i>Warning</i>	<i>String</i>	<i>Special warning for product</i>
<b>Response JSON Schema</b>	<pre>                 {"message_type": "Product", "scn_slug": "{{node_name}}",                 "payload": "Code;Description;Price;BasicUnit;ConversionFactor;                 SaleUnit;Volume;EAN_Tem;EAN_Kib;Type;LifeDuration;Warning;                 GrossWeight;StopSelling;Quantity;BillingQuantity;OrderQuantity                 ;                 GrossValue;NetValue;VatValue;TotalValue0000103089                 ;*****;;*****;                 *****;2810 *****;71201;A' ΗΠΑΚΛΕΙΟΥ -                 8111;N.ΗΠΑΚΛΕΙΟΥ;*****;71201;;;0;0;0;0;                 0;0;0"}"             </pre>			

### 3.2.3 Import Depositor Orders

Table 3: Import Depositor Orders Method

<b>Method</b>	ImportDepositorOrders			
<b>Request Arguments</b>	PubsubUrl (string), csvDirectory (string)			
<b>Synopsis (Method description)</b>	The method is responsible for parsing Depositor Order data and the corresponding products from a .csv file contained in the specified directory ( <i>csvDirectory</i> ) with specific fields (see input below), transforming them to the SELIS SCV(Supply Chain Visibility) models and publishing them to the requested PubSub url. The method supports parsing multiple files at once.			
<b>Input parameters</b>	<b>Format</b>	<b>Name</b>	<b>Type</b>	<b>SCV Semantic</b>
	<i>Comma Delimited</i>	<i>Erdat</i>	<i>DateTime("yyyyMMdd")</i>	<i>CreationDate</i>
		<i>Vbeln</i>	<i>String</i>	<i>DeliveryNo</i>
		<i>Wadaf</i>	<i>DateTime("yyyyMMdd")</i>	<i>DeliveryDateFrom</i>
		<i>Wadat</i>	<i>DateTime("yyyyMMdd")</i>	<i>DeliveryDateTo</i>
		<i>TimF</i>	-	<i>Not Used</i>
		<i>TimT</i>	-	<i>Not Used</i>
		<i>Werk</i>	<i>String</i>	<i>Warehouse</i>
		<i>Matnr</i>	<i>String</i>	<i>Product.Code</i>
		<i>Empt</i>	-	<i>Not Used</i>
		<i>Lgmng</i>	<i>Integer</i>	<i>Product.Quantity</i>
		<i>Kunnr</i>	<i>String</i>	<i>CustomerCode</i>
		<i>Name1</i>	-	<i>Not Used</i>
		<i>Stcd2</i>	-	<i>Not Used</i>
		<i>Name3</i>	-	<i>Not Used</i>
<i>Telf1</i>		-	<i>Not Used</i>	
<i>Empt1</i>	-	<i>Not Used</i>		

		<i>Sxo1</i>	-	<i>Not Used</i>
		<i>Empt2</i>	-	<i>Not Used</i>
		<i>Ort02</i>	-	<i>Not Used</i>
		<i>Ort01</i>	<i>String</i>	<i>Comments</i>
		<i>Stras</i>	-	<i>Not Used</i>
		<i>Pstlz</i>	-	<i>Not Used</i>
		<i>Text1</i>	-	<i>Not Used</i>
		<i>Posnr</i>	<i>String</i>	<i>PayerCode</i>
		<i>Text2</i>	-	<i>Not Used</i>
		<i>Kunrg</i>	-	<i>Not Used</i>
		<i>Kunpy</i>	-	<i>Not Used</i>
<b>Output parameters</b>	<b>Format</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>
	<i>JSON</i>	<i>CreationDate</i>	<i>DateTime</i>	<i>Creation date of order</i>
		<i>DeliveryNo</i>	<i>String</i>	<i>Order Identifier</i>
		<i>DeliveryDateFrom</i>	<i>DateTime</i>	<i>Delivery date range (from)</i>
		<i>DeliveryDateTo</i>	<i>DateTime</i>	<i>Delivery date range (to)</i>
		<i>Warehouse</i>	<i>String</i>	<i>Warehouse of Order</i>
		<i>Product.Code</i>	<i>String</i>	<i>Code of products in order</i>
		<i>Product.Quantity</i>	<i>Integer</i>	<i>Quantity of products in order</i>
		<i>CustomerCode</i>	<i>String</i>	<i>Code of customer</i>
		<i>Comments</i>	<i>String</i>	<i>Comments for order</i>
		<i>Document</i>	<i>String</i>	<i>Name of file parsed</i>
		<i>PoyerCode</i>	<i>String</i>	<i>Code of order payer</i>
<b>Response Schema</b>	<b>JSON</b>	<pre> {"message_type": "DepositorOrder", "scn_slug": "{{node_name}}", "payload": "CreationDate;DeliveryNo;DeliveryDateFrom; DeliveryDateTo;WareHouse;Comments;PayerCode; </pre>		

```

Document20-Dec-18 12:00:00 AM;2105536254;
20-Dec-18 12:00:00 AM;
20-Dec-1812:00:00 AM;2490;*****;000001;
1_DLVRV_1670_20181221_003036.CSV
"}
    
```

### 3.2.4 Import Invoices

Table 4: Import Invoices Method

<b>Method</b>	ImportInvoices			
<b>Request Arguments</b>	PubsubUrl (string) , csvDirectory (string)			
<b>Synopsis (Method description)</b>	The method is responsible for parsing Invoice data and the corresponding products from a .csv file contained in the specified csv Directory with specific fields(see input below), transforming them to the SELIS SCV(Supply Chain Visibility) models and publishing them to the requested PubSub url. The method supports parsing multiple files at once.			
<b>Input parameters</b>	<b>Format</b>	<b>Name</b>	<b>Type</b>	<b>SCV Semantic</b>
	<i>Comma Delimited</i>	<i>Kunpy</i>	-	<i>Not Used</i>
		<i>Kunnr</i>	<i>String</i>	<i>CustomerCode</i>
		<i>Name1</i>	-	<i>Not Used</i>
		<i>Stras</i>	-	<i>Not Used</i>
		<i>Wadat_D</i>	<i>DateTime("yyyyMMdd")</i>	<i>DeliveryDate</i>
		<i>Wadat_IS</i>	-	<i>Not Used</i>
		<i>Vbeln_D</i>	<i>String</i>	<i>DeliveryNo</i>
		<i>LineD</i>	<i>String</i>	<i>DeliveryLine_No</i>
		<i>Legaldoc_D</i>	<i>String</i>	<i>DeliveryLegalDoc</i>
		<i>Wadat_B</i>	<i>DateTime("yyyyMMdd")</i>	<i>BillingDate</i>
		<i>Vbeln_B</i>	<i>String</i>	<i>BillingNo</i>
<i>LineB</i>		-	<i>Not Used</i>	

		<i>Legaldoc_B</i>	<i>String</i>	<i>BillingLegalDoc</i>
		<i>Wadat_O</i>	-	<i>Not Used</i>
		<i>Erzet_</i>	-	<i>Not Used</i>
		<i>Vbeln_O</i>	<i>String</i>	<i>Product.Code</i>
		<i>LineO</i>	<i>String</i>	<i>OrderLine</i>
		<i>Matnr</i>	-	<i>Not Used</i>
		<i>Maktx</i>	-	<i>Not Used</i>
		<i>Qty_O</i>	<i>Long</i>	<i>OrderQuantity</i>
		<i>Qty_B</i>	<i>Decimal</i>	<i>BillingQuantity</i>
		<i>Unt</i>	<i>String</i>	<i>SaleUnit</i>
		<i>Kzwi1</i>	<i>Decimal</i>	<i>GrossValue</i>
		<i>Kzwi2</i>	<i>Decimal</i>	<i>NetValue</i>
		<i>Mwsbp</i>	<i>Decimal</i>	<i>VatValue</i>
		<i>Totalval</i>	<i>Decimal</i>	<i>TotalValue</i>
<b>Output parameters</b>	<b>Format</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>
	<i>JSON</i>	<i>CustomerCode</i>	<i>String</i>	<i>Code of Customer</i>
		<i>DeliveryDate</i>	<i>DateTime</i>	<i>Date of Delivery</i>
		<i>DeliveryNo</i>	<i>String</i>	<i>Delivery Identifier</i>
		<i>DeliveryLine_No</i>	<i>String</i>	<i>Line of Delivery in document</i>
		<i>DeliveryLegalDoc</i>	<i>String</i>	<i>Legal document of delivery</i>
		<i>Product.Code</i>	<i>String</i>	<i>Code of products in order</i>
		<i>BillingDate</i>	<i>DateTime</i>	<i>Date of Billing</i>
		<i>BillingNo</i>	<i>String</i>	<i>Number of billing</i>
		<i>BillingLegalDoc</i>	<i>String</i>	<i>Legal document of billing</i>
	<i>Document</i>	<i>String</i>	<i>Name of file parsed</i>	

		<i>Product.Code</i>	<i>String</i>	<i>Code of product referenced by Invoice</i>
		<i>LineO</i>	<i>String</i>	<i>Line of order in document</i>
		<i>OrderQuantity</i>	<i>Long</i>	<i>Quantity of items ordered</i>
		<i>BillingQuanntity</i>	<i>Decimal</i>	<i>Quantity of items billed</i>
		<i>SaleUnit</i>	<i>String</i>	<i>Unit of sale for products</i>
		<i>GrossValue</i>	<i>Decimal</i>	<i>Gross Value of Invoiced Products</i>
		<i>NetValue</i>	<i>Decimal</i>	<i>Net Value of Invoiced Products</i>
		<i>VatValue</i>	<i>Decimal</i>	<i>Vat Value of Invoiced Products</i>
		<i>TotalValue</i>	<i>Decimal</i>	<i>Total Value of Invoiced Products</i>

<b>Response JSON Schema</b>	<pre> "{\"message_type\":\"Customer\", \"scn_slug\":\"{{node_name}}\", \"payload\": \"SupplierVATNo;RetailerEANLocNo;InvoiceCodeNo;RetailerVATNo; SupplierEANLocNo;AgentEANLocNo;IssuerEANLocNo;AgentVATNo; IssuerVATNo;StoreEANLocNo;DocumentLine;DocumentNo; DocumentIssuedDate;DocumentDeliveredDate;OrderLine; OrderDate;DeliveryLocation;MeansOfPayment ;CorrespondingDocumentLine;CorrespondingDocumentNo; RelatedDocumentDate;PaymentDescription;PaymentDayNo; NetValue;DiscountValue;OtherDiscountPercentage; OtherDiscountValue;TotalDiscounts;ValueOfCharges; OtherChargesPercentage;OtherChargesValue;TotalCharges; VATPercentage;TaxedAmount;VATValue;VATDescription; TotalUnderlyingValue;TotalTaxedAmount;TotalVAT;MixedValue; TransportationValue;TransportationVAT;TransporterName; MeansofTransport;Guarantee;TotalPaymentAmount;TotalItems; TotalBoxes;TotalKilosGross;TotalKilosNet;Comments;LegalDoc_DA; DeliveryNo;BillingNo;BillingDate;BillingLine_No;LegalDoc_TIM; DeliveryLegalDoc;BillingLegalDoc;DeliveryDate;OrderNo; DeliveryLine_No;GrossValue;OrderMeasurementUnit ;InvoiceQuantity;InvoiceMeasurementUnit;Document; TotalValue;isTDA;;;;;;;01/01/0001 12:00:00 μ ;01/01/0001 12:00:00 μ;92146;01/01/0001 12:00:00 μ                 </pre>
-----------------------------	--

```

;;;;;01/01/0001 12:00:00
πμ;;;;;27,12;;;;;27,12;;;;;20180926;20180925;
01/01/0001 12:00:00 πμ;3300045071;;000001;000001;
25/09/2018 12:00:00
πμ;20180925;2105466342;131,093;;;3_INVOICE_1600_20180926_003045.C
SV;27,12;False"}"
    
```

### 3.3 ThreePL Adapter

The ThreePL adapter handles the transformation of answers to depositor orders sent after the depositor orders are received. After the parsing is complete, the items are transformed to the SELIS SCV data models and then published to the PubSub in JSON format as a .csv message. The following sections describe the individual functions of the adapter.

#### 3.3.1 Import Answers

Table 5: Import Answers Method

<b>Method</b>	ImportAnswers			
<b>Request Arguments</b>	PubsubUrl (string) , csvDirectory (string)			
<b>Synopsis (Method description)</b>	The method is responsible for parsing the fulfillment answers from a 3PL provider to the original depositor orders. The method parses a .csv file contained in the specified csv Directory with specific fields (see input below), transforming them to the SELIS SCV(Supply Chain Visibility) models and publishing them to the requested PubSub url. The method supports parsing multiple files at once.			
<b>Input parameters</b>	<b>Format</b>	<b>Name</b>	<b>Type</b>	<b>SCV Semantic</b>
	<i>Comma Delimited</i>	<i>Wadat</i>	<i>DateTime("yyyyMMdd")</i>	<i>DeliveryDate</i>
		<i>Emp</i>	-	<i>Not Used</i>
		<i>Nbr</i>	-	<i>Not Used</i>
		<i>Empt2</i>	-	<i>Not Used</i>
		<i>Mat</i>	<i>String</i>	<i>Product.Code</i>
		<i>Wrk</i>	<i>String</i>	<i>Warehouse</i>
		<i>Empt3</i>	-	<i>Not Used</i>
<i>Qty</i>		<i>Int</i>	<i>Product.Quantity</i>	

		<i>Empt4</i>	-	<i>Not Used</i>
		<i>Kunnr</i>	<i>String</i>	<i>CustomerCode</i>
		<i>Mvtext</i>	-	<i>Not Used</i>
		<i>Vbeln</i>	<i>String</i>	<i>DeliveryNo</i>
		<i>Vbeln1</i>	<i>String</i>	<i>OrderNo</i>
		<i>Pcki</i>	<i>String</i>	<i>PickingListNo</i>
		<i>Nbrtu</i>	<i>String</i>	<i>VehiclePlateNo</i>
		<i>Empt5</i>	-	<i>Not used</i>
		<i>Mvtext1</i>	-	<i>Not Used</i>
		<i>Posnr</i>	-	<i>LineNo</i>
		<i>Empt6</i>	-	<i>Not Used</i>
		<i>Koli</i>	-	<i>NoOfColli</i>
<b>Output parameters</b>	<b>Format</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>
	<i>JSON</i>	<i>DeliveryDate</i>	<i>DateTime</i>	<i>Date of Delivery</i>
		<i>Product.Code</i>	<i>String</i>	<i>Codes of Products</i>
		<i>Warehouse</i>	<i>String</i>	<i>Warehouse for order loading</i>
		<i>Product.Quantity</i>	<i>int</i>	<i>The conversion factor of the product</i>
		<i>CustomerCode</i>	<i>String</i>	<i>Code of ordering Customer</i>
		<i>DeliveryNo</i>	<i>String</i>	<i>Referenced Delivery Identifier</i>
		<i>OrderNo</i>	<i>String</i>	<i>Referenced Order Identifier</i>
		<i>PickingListNo</i>	<i>String</i>	<i>Number of Picking List</i>
		<i>VehiclePlateNo</i>	<i>String</i>	<i>Plate No of vehicle transporting goods</i>
		<i>LineNo</i>	<i>String</i>	<i>Line number in document</i>
		<i>NoOfColli</i>	<i>String</i>	<i>Number of collo packages</i>



<b>Response Schema</b>	<b>JSON</b>	<pre>{ "message_type": "ThreePLAnswer", "scn_slug": "{node_name}" , "payload": "DeliveryDate;MaterialNumber;WareHouse;Quantity; DeliveryNo;PickingListNo;VehiclePlateNo;NoOfColli; Document;LineNo;CustomerCode;OrderNo;Products; 9254301;2360;;2105386572;;;0; 2_OUTB_1600_20180615_003033.CSV;000001;0000215748;; 123213,123213,123123;" }</pre>
------------------------	-------------	--

### 3.4 PoD Adapter

The PoD adapter handles the transformation of Proof of Delivery documents sent after the depositor orders are invoiced and delivered. After the parsing is complete, the items are transformed to the SELIS SCV data models and then published to the PubSub in JSON format as a .csv message. The following sections describe the individual functions of the adapter.

#### 3.4.1 Import Proofs of Delivery

Table 6: Import Proofs of Delivery Method

<b>Method</b>	ImportProofOfDeliveries			
<b>Request Arguments</b>	PubsubUrl (string) , csvDirectory (string)			
<b>Synopsis (Method description)</b>	The method is responsible for parsing Proof of Delivery data from a .csv file contained in the specified csvDirectory with specific fields(see input below), transforming them to the SELIS SCV(Supply Chain Visibility) models and publishing them to the requested PubSub url. The method supports parsing multiple files at once.			
<b>Input parameters</b>	<b>Format</b>	<b>Name</b>	<b>Type</b>	<b>SCV Semantic</b>
	<i>Comma Delimited</i>	<i>OrderConfirmDT</i>	<i>DateTime</i>	<i>DateIssued</i>
		<i>OrderId</i>	<i>String</i>	<i>PODNumber</i>
		<i>OrderStatus</i>	-	<i>Not Used</i>
<i>StatusDescription</i>		<i>String</i>	<i>DeliveryStatus</i>	
<b>Output parameters</b>	<b>Format</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>
	<i>JSON</i>	<i>DateIssued</i>	<i>DateTime</i>	<i>Issue date of proof of delivery</i>

		<i>PODNumber</i>	<i>String</i>	<i>Corresponding order to proof of delivery identifier</i>
		<i>DeliveryStatus</i>	<i>String</i>	<i>Status of delivery</i>
<b>Response Schema</b>	<b>JSON</b>	<pre> {"message_type": "ProofOfDelivery", "scn_slug": "{{node_name}}", "payload": "PODNumber;DateIssued;QualityStatus;QuantityStatus; DeliveryStatus;PODAttachment;Comments*****; 04-Oct-18 1:00:15 PM;;;Served Without signature;;"}                 </pre>		

### 3.5 WMS Adapter

The WMS adapter handles the transformation of Warehouse Order documents. After the parsing is complete, the items are transformed to the SELIS SCV data models and then published to the PubSub in JSON format as a .csv message. The following sections describe the individual functions of the adapter.

#### 3.5.1 Import WMS Orders

Table 7: Import WMS Orders Method

<b>Method</b>	ImportOrderWMS			
<b>Request Arguments</b>	PubsubUrl (string) , csvDirectory (string)			
<b>Synopsis (Method description)</b>	The method is responsible for parsing Orders from a WMS system in a form of a .csv file contained in the specified csvDirectory with specific fields (see input below), transforming them to the SELIS SCV (Supply Chain Visibility) models and publishing them to the requested PubSub url. The method supports parsing multiple files at once.			
<b>Input parameters</b>	<b>Format</b>	<b>Name</b>	<b>Type</b>	<b>SCV Semantic</b>
	<i>Comma Delimited</i>	<i>OrdID</i>	<i>Int</i>	<i>OrdID</i>
		<i>OrdOrderID_Remote</i>	<i>int</i>	<i>OrdOrderID_Remote</i>
		<i>OrdOrderExternalCode</i>	<i>String</i>	<i>OrdOrderExternalCode</i>
		<i>OrdOrderDescr</i>	<i>String</i>	<i>OrdOrderDescr</i>
		<i>OrdRegDateTime</i>	<i>DateTime</i>	<i>OrdRegDateTime</i>
		<i>OrdDateTime</i>	<i>DateTime</i>	<i>OrdDateTime</i>
		<i>OrdDeliveryDateTime</i>	<i>DateTime</i>	<i>OrdDeliveryDateTime</i>
<i>ordDepositorCode</i>		<i>String</i>	<i>ordDepositorCode</i>	

<i>OrdDepositorFullName</i>	<i>String</i>	<i>OrdDepositorFullName</i>
<i>OrdCustomerCode</i>	<i>String</i>	<i>OrdCustomerCode</i>
<i>OrdCustomerFullName</i>	<i>String</i>	<i>OrdCustomerFullName</i>
<i>OrdStatus</i>	<i>Int</i>	<i>OrdStatus</i>
<i>OrdMemo</i>	<i>String</i>	<i>OrdMemo</i>
<i>OrdPurpose</i>	<i>String</i>	<i>OrdPurpose</i>
<i>OrdDeliveryValue</i>	<i>Float</i>	<i>OrdDeliveryValue</i>
<i>OrdDeliveryNotes</i>	<i>String</i>	<i>OrdDeliveryNotes</i>
<i>OrdIsReturn</i>	<i>Int</i>	<i>OrdIsReturn</i>
<i>OrdPrdTypeDescr</i>	<i>String</i>	<i>OrdPrdTypeDescr</i>
<i>OrdWrhRouteCode</i>	<i>String</i>	<i>OrdWrhRoutingCode</i>
<i>OrdDlsId</i>	<i>Int</i>	<i>OrdDlsId</i>
<i>OrdDLsDescr</i>	<i>String</i>	<i>OrdDlsDescr</i>
<i>OrdDlsZipCode</i>	<i>String</i>	<i>OrdDlsZipCode</i>
<i>OrdDlsCity</i>	<i>String</i>	<i>OrdDlsCity</i>
<i>OrdDlsArea</i>	<i>String</i>	<i>OrdDlsArea</i>
<i>OrdRoutingStatus</i>	<i>Int</i>	<i>OrdRoutingStatus</i>
<i>OrdLfdWeight</i>	<i>Float</i>	<i>OrdLfdWeight</i>
<i>OrdLfdVol</i>	<i>Float</i>	<i>OrdLfdVol</i>
<i>OrdLfdCtn</i>	<i>Int</i>	<i>OrdLftCtn</i>
<i>OrdLfdBar</i>	<i>Int</i>	<i>OrdLftBar</i>
<i>OrdLfdPalEuro</i>	<i>Int</i>	<i>OrdLfdPalEuro</i>
<i>OrdLfdPalInd</i>	<i>Int</i>	<i>OrdLfdPalInd</i>
<i>OrdLfdPalElsa</i>	<i>Int</i>	<i>OrdLfdPalElsa</i>
<i>OrdLfdParOU</i>	<i>Int</i>	<i>OrdLfdParOU</i>
<i>OrdLfdVar</i>	<i>Int</i>	<i>OrdLfdVar</i>

		<i>OrdVoucherDateTime</i>	<i>DateTime</i>	<i>OrdVoucherDateTime</i>
		<i>OrdVoucherNumber</i>	<i>String</i>	<i>OrdVoucherNumber</i>
		<i>OrdVoucherSeries</i>	<i>String</i>	<i>OrdVoucherSeries</i>
		<i>OrdVoucherId</i>	<i>Int</i>	<i>OrdVoucherId</i>
		<i>OrdAgencyCode</i>	<i>String</i>	<i>OrdAgencyCode</i>
		<i>OrdAgencyDescription</i>	<i>String</i>	<i>OrdAgencyDescription</i>
		<i>OrdAgencyAdress</i>	<i>String</i>	<i>OrdAgencyAdress</i>
		<i>OrdTelRouteId</i>	<i>Long</i>	<i>OrdTelRouteId</i>
		<i>OrdVchRouteNr</i>	<i>Int</i>	<i>OrdVchRouteNr</i>
		<i>OrdVhrDate</i>	<i>DatTime</i>	<i>OrdVhrDate</i>
		<i>OrdVhrID</i>	<i>Long</i>	<i>OrdVhrID</i>
		<i>OrdLastUpdateDateTime</i>	<i>DateTime</i>	<i>OrdLastUpdateDateTime</i>
		<i>OrdReadyForLoading</i>	<i>Int</i>	<i>OrdReadyForLoading</i>
<b>Output parameters</b>	<b>Format</b>	<b>Name</b>	<b>Type</b>	<b>Comment</b>
	<i>JSON</i>	<i>OrdID</i>	<i>Int</i>	<i>Order Id</i>
		<i>OrdOrderID_Remote</i>	<i>int</i>	<i>Order Id Remote</i>
		<i>OrdOrderExternalCode</i>	<i>String</i>	<i>External Order Code</i>
		<i>OrdOrderDescr</i>	<i>String</i>	<i>Order Description</i>
		<i>OrdRegDateTime</i>	<i>DateTime</i>	<i>Order Registration Date</i>
		<i>OrdDateTime</i>	<i>DateTime</i>	<i>Order Date</i>
		<i>OrdDeliveryDateTime</i>	<i>DateTime</i>	<i>Order Delivery Date</i>
		<i>ordDepositorCode</i>	<i>String</i>	<i>Order Depositor Code</i>
		<i>OrdDepositorFullName</i>	<i>String</i>	<i>Order Depositor Full Name</i>
		<i>OrdCustomerCode</i>	<i>String</i>	<i>Order Customer Code</i>
		<i>OrdCustomerFullName</i>	<i>String</i>	<i>Order Customer Full Name</i>
		<i>OrdStatus</i>	<i>Int</i>	<i>Order Status</i>

<i>OrdMemo</i>	<i>String</i>	<i>Order Memo</i>
<i>OrdPurpose</i>	<i>String</i>	<i>Order Purpose</i>
<i>OrdDeliveryValue</i>	<i>Float</i>	<i>Order Delivery Value</i>
<i>OrdDeliveryNotes</i>	<i>String</i>	<i>Order Delivery Notes</i>
<i>OrdIsReturn</i>	<i>Int</i>	<i>Order Is returned</i>
<i>OrdPrdTypeDescr</i>	<i>String</i>	<i>Order Product Type Description</i>
<i>OrdWrhRouteCode</i>	<i>String</i>	<i>Order Warehouse Route Code</i>
<i>OrdDlslid</i>	<i>Int</i>	<i>Order DL Id</i>
<i>OrdDLsDescr</i>	<i>String</i>	<i>Order DL Description</i>
<i>OrdDlsZipCode</i>	<i>String</i>	<i>Order DLs Zip Code</i>
<i>OrdDlsCity</i>	<i>String</i>	<i>Order Dls City</i>
<i>OrdDlsArea</i>	<i>String</i>	<i>Order Dls Area</i>
<i>OrdRoutingStatus</i>	<i>Int</i>	<i>Order Routing Status</i>
<i>OrdLfdWeight</i>	<i>Float</i>	<i>OrdLfdWeight</i>
<i>OrdLfdVol</i>	<i>Float</i>	<i>OrdLfdVol</i>
<i>OrdLfdCtn</i>	<i>Int</i>	<i>OrdLftCtn</i>
<i>OrdLfdBar</i>	<i>Int</i>	<i>OrdLftBar</i>
<i>OrdLfdPalEuro</i>	<i>Int</i>	<i>OrdLfdPalEuro</i>
<i>OrdLfdPalInd</i>	<i>Int</i>	<i>OrdLfdPalInd</i>
<i>OrdLfdPalElsa</i>	<i>Int</i>	<i>OrdLfdPalElsa</i>
<i>OrdLfdParOU</i>	<i>Int</i>	<i>OrdLfdParOU</i>
<i>OrdLfdVar</i>	<i>Int</i>	<i>OrdLfdVar</i>
<i>OrdVoucherDateTime</i>	<i>DateTime</i>	<i>Order Voucher Date</i>
<i>OrdVoucherNumber</i>	<i>String</i>	<i>Order Voucher Number</i>
<i>OrdVoucherSeries</i>	<i>String</i>	<i>Order Voucher Series</i>
<i>OrdVoucherId</i>	<i>Int</i>	<i>Order Voucher Id</i>

		<i>OrdAgencyCode</i>	<i>String</i>	<i>Order agency Code</i>
		<i>OrdAgencyDescription</i>	<i>String</i>	<i>Order Agency Description</i>
		<i>OrdAgencyAdress</i>	<i>String</i>	<i>Order Agency Address</i>
		<i>OrdTelRouteId</i>	<i>Long</i>	<i>Order Tel Route Id</i>
		<i>OrdVchRouteNr</i>	<i>Int</i>	<i>Order Voucher Route No</i>
		<i>OrdVhrDate</i>	<i>DateTime</i>	<i>Order Vehicle Date</i>
		<i>OrdVhrID</i>	<i>Long</i>	<i>Order Vehicle Id</i>
		<i>OrdLastUpdateDateTime</i>	<i>DateTime</i>	<i>Order Last Updated Date</i>
		<i>OrdReadyForLoading</i>	<i>Int</i>	<i>Order Ready for Loading</i>
<b>Response JSON Schema</b>	<pre> {"message_type": "ProofOfDelivery", "scn_slug": "{node_name}" , "payload": "PODNumber;DateIssued;QualityStatus;QuantityStatus ; DeliveryStatus;PODAttachment;Comments*****; 04-Oct-18 1:00:15 PM;;;Served Without signature;;"}                 </pre>			

### 3.6 RailData Adapter

The RailData adapter handles the transformation of train data of an intermodal operator and particularly:

- Train Schedules,
- Container Status events and
- Wagon events

After the parsing is complete, the items are transformed to the SELIS SCV data models and then published to the PubSub in JSON format as a .csv message. The following sections describe the individual functions of the adapter.

#### 3.6.1 Import Train Schedules

Table 8: Import Train Schedules Method

<b>Method</b>	ImportTrainScheduleCalendar
<b>Request Arguments</b>	PubsubUrl (string), csvFile(string), messageType(string)

<b>Synopsis (Method description)</b>	This method parses calendar data of train schedules that are exported in CSV format. The parsed data is converted to the appropriate SELIS models that unify schedules from various cargo transport sources into a universal data schema. Finally, the transformed data is published to the Pub/Sub component to reach possible subscribers and/or trigger Big Data Analytics recipes for related analysis.			
<b>Input parameters</b>	<b>Format</b>	<b>Name</b>	<b>Type</b>	<b>SCV Semantic</b>
	<i>Semi-Colon Delimited</i>	<i>CalendarId</i>	<i>int</i>	<i>CalendarId</i>
	<i>TimetableId</i>	<i>int</i>	<i>TimetableId</i>	
	<i>Announced</i>	<i>DateTime</i>	<i>Announced</i>	
	<i>Cancelled</i>	<i>DateTime</i>	<i>Cancelled</i>	
	<i>Private</i>	<i>bool</i>	<i>Private</i>	
	<i>Departure</i>	<i>DateTime</i>	<i>Departure</i>	
	<i>Arrival</i>	<i>DateTime</i>	<i>Arrival</i>	
	<i>TimeTableId</i>	<i>int</i>	<i>TimeTableId</i>	
	<i>Weekdays</i>	<i>string</i>	<i>Weekdays</i>	
	<i>ValidFrom</i>	<i>DateTime</i>	<i>ValidFrom</i>	
	<i>ValidTill</i>	<i>DateTime</i>	<i>ValidTill</i>	
	<i>DepartureSecondOfTheDay</i>	<i>int</i>	<i>DepartureSecondOfTheDay</i>	
	<i>TrainId</i>	<i>int</i>	<i>TrainId</i>	
	<i>TrainNumber</i>	<i>int</i>	<i>TrainNumber</i>	
	<i>HasReachedDestination</i>	<i>bool</i>	<i>HasReachedDestination</i>	
	<i>ReachedDestinationDateTime</i>	<i>DateTime</i>	<i>ReachedDestinationDateTime</i>	
	<i>StartUIC</i>	<i>int</i>	<i>StartUIC</i>	
	<i>StartUicCountryId</i>	<i>int</i>	<i>StartUicCountryId</i>	
	<i>EndUIC</i>	<i>int</i>	<i>EndUIC</i>	
<i>EndUicCountryId</i>	<i>int</i>	<i>EndUicCountryId</i>		
<i>ExternalTrainId</i>	<i>long</i>	<i>ExternalTrainId</i>		
<i>TrainCommercialName</i>	<i>string</i>	<i>TrainCommercialName</i>		

		<i>LineId</i>	<i>int</i>	<i>LineId</i>
<b>Output parameters</b>	<b>Format</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>
	<i>JSON</i>	<i>CalendarId</i>	<i>String</i>	<i>Calendar Id</i>
		<i>TimetableId</i>	<i>String</i>	<i>Timetable Id</i>
		<i>Announced</i>	<i>String</i>	<i>Announced Date</i>
		<i>Cancelled</i>	<i>String</i>	<i>Is Cancelled</i>
		<i>Private</i>	<i>String</i>	<i>Is Private</i>
		<i>Departure</i>	<i>String</i>	<i>Departure Date</i>
		<i>Arrival</i>	<i>String</i>	<i>Arrival Date</i>
		<i>TimeTableId</i>	<i>String</i>	<i>Timetable Id</i>
		<i>Weekdays</i>	<i>String</i>	<i>Valid Weekdays</i>
		<i>ValidFrom</i>	<i>String</i>	<i>Valid from</i>
		<i>ValidTill</i>	<i>String</i>	<i>Valid until</i>
		<i>DepartureSecondOfTheDay</i>	<i>String</i>	<i>Departure second of the day</i>
		<i>TrainId</i>	<i>String</i>	<i>Train Id</i>
		<i>TrainNumber</i>	<i>String</i>	<i>Train registered number</i>
		<i>HasReachedDestination</i>	<i>String</i>	<i>Has reached destination</i>
		<i>ReachedDestinationDateTime</i>	<i>String</i>	<i>Reached destination Datetime</i>
		<i>StartUIC</i>	<i>String</i>	<i>Start UIC</i>
		<i>StartUicCountryId</i>	<i>String</i>	<i>Start UIC Country Id</i>
		<i>EndUIC</i>	<i>String</i>	<i>End UIC</i>
		<i>EndUicCountryId</i>	<i>String</i>	<i>End UIC Country Id</i>



		<i>ExternalTrainId</i>	<i>String</i>	<i>External Train Id</i>
		<i>TrainCommercialName</i>	<i>String</i>	<i>Train Commercial Name</i>
		<i>LineId</i>	<i>String</i>	<i>Line Id</i>
<b>Response JSON Schema</b>	<pre> {   "message_type": "CalendarSchedules",   "payload":   "CalendarId;TimetableId;Announced;Cancelled;Private;D   eparture;Arrival;TimeTableId;Weekdays;ValidFrom;Valid   Till;DepartureSecondOfTheDay;TrainId;TrainNumber;HasR   eachedDestination;ReachedDestinationDateTime;StartUIC   ;StartUicCountryId;EndUIC;EndUicCountryId;ExternalTra   inId;TrainCommercialName;LineId\n43338;210;;;False;1/   1/2016          12:44:00          ;2/1/2016   10:06:00;210;MO,FR,SA,SU;2/2/2015   12:00:00;;;45420;33121;450287;True;4/1/2016   10:17:25;443515;79;332841;54;17125503000;RX001;201",   "scn_slug": "selis" } </pre>			

### 3.6.2 Import Container Status Events

Table 9: Import Container Status Events Method

<b>Method</b>	ImportTrainContainerStatuses			
<b>Request Arguments</b>	PubsubUrl (string), csvFile(string), messageType(string)			
<b>Synopsis (Method description)</b>	This method parses container transportation status data where the means of transport is rail transport. The original content is imported from a CSV format and then transformed to the appropriate SELIS models. The converted data is published to the Pub/Sub component to reach possible subscribers and/or trigger Big Data Analytics recipes for related analysis.			
<b>Input parameters</b>	<b>Format</b>	<b>Name</b>	<b>Type</b>	<b>SCV Semantic</b>
	<i>Semi-Colon Delimited</i>	<i>ContainerId</i>	<i>int</i>	<i>ContainerId</i>
		<i>ContainerNumber</i>	<i>int</i>	<i>ContainerNumber</i>
		<i>ContainerEndStationUIC</i>	<i>long</i>	<i>ContainerEndStationUIC</i>
		<i>WagonId</i>	<i>int</i>	<i>WagonId</i>
		<i>WagonNumber</i>	<i>int</i>	<i>WagonNumber</i>
		<i>WagonReachedDestinationDateTime</i>	<i>DateTime</i>	<i>WagonReachedDestinationDateTime</i>
		<i>WagonStartCountryCode</i>	<i>string</i>	<i>WagonStartCountryCode</i>
		<i>WagonStartStationName</i>	<i>string</i>	<i>WagonStartStationName</i>
		<i>WagonEndCountryCode</i>	<i>string</i>	<i>WagonEndCountryCode</i>
		<i>WagonEndStationName</i>	<i>string</i>	<i>WagonEndStationName</i>
		<i>TrainId</i>	<i>int</i>	<i>TrainId</i>
		<i>CalendarId</i>	<i>int</i>	<i>CalendarId</i>
		<i>TrainNumber</i>	<i>int</i>	<i>TrainNumber</i>
		<i>TrainScheduledDepartureDateTime</i>	<i>DateTime</i>	<i>TrainScheduledDepartureDateTime</i>
<i>TrainReachedDestinationDateTime</i>		<i>DateTime</i>	<i>TrainReachedDestinationDateTime</i>	
<i>ExternalTrainId</i>	<i>long</i>	<i>ExternalTrainId</i>		

		<i>TrainCommercialName</i>	<i>string</i>	<i>TrainCommercialName</i>
		<i>LineId</i>	<i>int</i>	<i>LineId</i>
		<i>TrainStartCountryCode</i>	<i>string</i>	<i>TrainStartCountryCode</i>
		<i>TrainStartStationName</i>	<i>string</i>	<i>TrainStartStationName</i>
		<i>TrainEndCountryCode</i>	<i>string</i>	<i>TrainEndCountryCode</i>
		<i>TrainEndStationName</i>	<i>string</i>	<i>TrainEndStationName</i>
<b>Output parameters</b>	<b>Format</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>
	JSON	<i>ContainerId</i>	<i>String</i>	<i>Container Id</i>
		<i>ContainerNumber</i>	<i>String</i>	<i>Container number</i>
		<i>ContainerEndStationUIC</i>	<i>String</i>	<i>Container end station UIC</i>
		<i>WagonId</i>	<i>String</i>	<i>Wagon Id</i>
		<i>WagonNumber</i>	<i>String</i>	<i>Wagon number</i>
		<i>WagonReachedDestinationDateTime</i>	<i>String</i>	<i>Wagon reached destination date</i>
		<i>WagonStartCountryCode</i>	<i>String</i>	<i>Wagon start country code</i>
		<i>WagonStartStationName</i>	<i>String</i>	<i>Wagon start station name</i>
		<i>WagonEndCountryCode</i>	<i>String</i>	<i>Wagon end country code</i>
		<i>WagonEndStationName</i>	<i>String</i>	<i>Wagon end station name</i>
		<i>TrainId</i>	<i>String</i>	<i>Train Id</i>
		<i>CalendarId</i>	<i>String</i>	<i>Calendar Id</i>
		<i>TrainNumber</i>	<i>String</i>	<i>Train number</i>
		<i>TrainScheduledDepartureDateTime</i>	<i>String</i>	<i>Train scheduled departure Date</i>
		<i>TrainReachedDestinationDateTime</i>	<i>String</i>	<i>Train reached destination date</i>
	<i>ExternalTrainId</i>	<i>String</i>	<i>External train Id</i>	

		<i>TrainCommercialName</i>	<i>String</i>	<i>Train commercial name</i>
		<i>LineId</i>	<i>String</i>	<i>Line Id</i>
		<i>TrainStartCountryCode</i>	<i>String</i>	<i>Train start country code</i>
		<i>TrainStartStationName</i>	<i>String</i>	<i>Train start station</i>
		<i>TrainEndCountryCode</i>	<i>String</i>	<i>Train end country code</i>
		<i>TrainEndStationName</i>	<i>String</i>	<i>Train end station</i>
<b>Response JSON Schema</b>	<pre> {   "message_type": "ContainerStatus",   "scn_slug": "selis",   "payload":   "ContainerId;ContainerNumber;UtiId;ContainerEndStationUIC;WagonId;WagonNumber;ReachedDestinationDateTime;WagonStartCountryCode;WagonStartStationName;WagonEndCountryCode;WagonEndStationName;TrainId;CalendarId;TrainNumber;ScheduledDepartureDateTime;ExternalTrainId;TrainCommercialName;LineId;TrainStartCountryCode;TrainStartStationName;TrainEndCountryCode;TrainEndStationName\n634407;2989;766192563000;;682815;233332;4/1/2016 10:17:25;SI;Koper Luka;CZ;Dobra u Frydku Mistku;33121;43338;450287;1/1/2016 12:44:00;17125503000;RX001;201;SI;Koper Luka;CZ;Dobra u Frydku Mistku" } </pre>			

### 3.6.3 Import Wagon Events

Table 10: Import Wagon Events Method

<b>Method</b>	ImportWagonEvents
<b>Request Arguments</b>	PubsubUrl (string), csvFile(string), messageType(string)
<b>Synopsis (Method description)</b>	This method parses various wagon events that are generated throughout the transportation of monitored containers. Those events are converted to the universal

SELIS model structure and the afterwards published to the Pub/Sub component of the SELIS node for further distribution and/or analysis.				
Input parameters	Format	Name	Type	SCV Semantic
	<i>Semi-Colon Delimited</i>	<i>WagonEventId</i>	<i>Int</i>	<i>WagonEventId</i>
		<i>WagonId</i>	<i>Int</i>	<i>WagonId</i>
		<i>TrainId</i>	<i>int</i>	<i>TrainId</i>
		<i>EventType</i>	<i>string</i>	<i>EventType</i>
		<i>LocationX</i>	<i>double</i>	<i>LocationX</i>
		<i>LocationY</i>	<i>double</i>	<i>LocationY</i>
		<i>WagonStatusId</i>	<i>int</i>	<i>WagonStatusId</i>
		<i>WagonStatusName</i>	<i>string</i>	<i>WagonStatusName</i>
		<i>WagonIrregularityId</i>	<i>int</i>	<i>WagonIrregularityId</i>
		<i>WagonIrregularityName</i>	<i>string</i>	<i>WagonIrregularityName</i>
		<i>EventDateTime</i>	<i>DateTime</i>	<i>EventDateTime</i>
		<i>Comment</i>	<i>string</i>	<i>Comment</i>
		<i>RisStatusId</i>	<i>Int</i>	<i>RisStatusId</i>
		<i>RisStatusName</i>	<i>string</i>	<i>RisStatusName</i>
		<i>LocationDescription</i>	<i>string</i>	<i>LocationDescription</i>
		<i>SourceId</i>	<i>int</i>	<i>SourceId</i>
		<i>SourceName</i>	<i>string</i>	<i>SourceName</i>
		<i>IsProcessed</i>	<i>bool</i>	<i>IsProcessed</i>
		<i>DelaySeconds</i>	<i>int</i>	<i>DelaySeconds</i>
<i>TrainEventId</i>		<i>int</i>	<i>TrainEventId</i>	
<i>Uic</i>	<i>string</i>	<i>Uic</i>		
<i>StationId</i>	<i>int</i>	<i>StationId</i>		
<i>TrainOperatingNumber</i>	<i>int</i>	<i>TrainOperatingNumber</i>		

		<i>IsDelayProcessed</i>	<i>bool</i>	<i>IsDelayProcessed</i>
		<i>WagonEventStationCountryCode</i>	<i>string</i>	<i>WagonEventStationCountryCode</i>
		<i>WagonEventStationName</i>	<i>string</i>	<i>WagonEventStationName</i>
<b>Output parameters</b>	<b>Format</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>
	<i>JSON</i>	<i>WagonEventId</i>	<i>String</i>	<i>Wagon event Id</i>
		<i>WagonId</i>	<i>String</i>	<i>Wagon Id</i>
		<i>TrainId</i>	<i>String</i>	<i>Train Id</i>
		<i>EventType</i>	<i>String</i>	<i>Event type</i>
		<i>LocationX</i>	<i>String</i>	<i>Location latitude</i>
		<i>LocationY</i>	<i>String</i>	<i>Location longitude</i>
		<i>WagonStatusId</i>	<i>String</i>	<i>Wagon status Id</i>
		<i>WagonStatusName</i>	<i>String</i>	<i>Wagon status name</i>
		<i>WagonIrregularityId</i>	<i>String</i>	<i>Wagon irregularity Id</i>
		<i>WagonIrregularityName</i>	<i>String</i>	<i>Wagon irregularity name</i>
		<i>EventDateTime</i>	<i>String</i>	<i>Event date and time</i>
		<i>Comment</i>	<i>String</i>	<i>Comment</i>
		<i>RisStatusId</i>	<i>String</i>	<i>Risk status Id</i>
		<i>RisStatusName</i>	<i>String</i>	<i>Risk status name</i>
		<i>LocationDescription</i>	<i>String</i>	<i>Location description</i>
		<i>SourceId</i>	<i>String</i>	<i>Source Id</i>
<i>SourceName</i>	<i>String</i>	<i>Source name</i>		
<i>IsProcessed</i>	<i>String</i>	<i>Is processed</i>		

		<i>DelaySeconds</i>	<i>String</i>	<i>Delay in seconds</i>
		<i>TrainEventId</i>	<i>String</i>	<i>Train event Id</i>
		<i>Uic</i>	<i>String</i>	<i>UIC</i>
		<i>StationId</i>	<i>String</i>	<i>Station Id</i>
		<i>TrainOperatingNumber</i>	<i>String</i>	<i>Train operating number</i>
		<i>IsDelayProcessed</i>	<i>String</i>	<i>Is delay processed</i>
		<i>WagonEventStationCountryCode</i>	<i>String</i>	<i>Wagon event station country code</i>
		<i>WagonEventStationName</i>	<i>String</i>	<i>Wagon event station name</i>
<b>Response Schema</b>	<b>JSON</b>	<pre> {   "message_type": "WagonEvent",   "scn_slug": "selis",   "payload":   "WagonEventId;WagonId;TrainId;EventType;LocationX;LocationY;WagonStatusId;WagonStatusName;WagonIrregularityId;WagonIrregularityName;EventDateTime;Comment;RisStatusId;RisStatusName;LocationDescription;SourceId;SourceName;IsProcessed;DelaySeconds;TrainEventId;Uic;StationId;TrainOperatingNumber;IsDelayProcessed;WagonEventStationCountryCode;WagonEventStationName\n5132236;345683;15622;Status;15,651237;46,539049;;;1/1/2014 8:43:00;41408 / ZBIRNIK;1;Arrival;Maribor Tezno;10;RIS-SI;True;;301910;433045;21;4350;False;SI;Maribor Tezno" } </pre>		

### 3.7 SELIS Node Connectors in Zenodo

In the interest of the Open Research Data Pilot, this publication( <https://github.com/selisproject/selis-node-connectors>) provides a basic implementation of connectors used in real world applications for supply chain participants to connect and funnel their data to the SELIS Community Node (SCN).

Along with the connectors, obfuscated/test data is also provided for each solution. The connectors all follow the same approach, which is parsing the data provided, transforming them into a SCN-compatible data exchange model and publishing them to the SCN under a specific configuration.

The samples provided parse data generated by WMS or ERP systems for monitoring order life-cycle (initial order to delivery), such as order fulfilment data, invoice and shipping document data, proof of delivery data, as well as static data such as Customer or Product data. Other datasets include schedules for various cargo transport modes provided by the according logistics service providers that participate in SELIS.



## 4 Conclusions

This deliverable reports on the specifications and configuration of the SELIS Node Connectors towards the Pilot on Open Research Data in regards with providing the relevant connectors and interfaces for connecting data sources to the content-based routing infrastructure. It describes the overall methodological approach, an overview of the engaged open data repositories and the Pilot on Open Research Data in Horizon 2020, and how SELIS is actively facilitating towards the Pilot on Open Research Data.

Appropriate repositories have been established for storing the results of the project with respect to the SELIS node connectors and providing access to the scientific community, such as OpenAIRE<sup>6</sup> and Zenodo<sup>7</sup>. OpenAIRE provides a repository called Zenodo that can be used for depositing data. OpenAIRE provides researcher support and services for the Open Research Data Pilot. The Zenodo link for the SELIS connectors can be found at the following location: <https://zenodo.org/record/3248918>.

The purpose of the SELIS Adapters is to transform data from various sources (either exported from software or extracted from proprietary data) into a common SELIS model. The content-based routing infrastructure is a core component of the SCN architecture that enables the sharing of information with the usage of pub/sub technologies.

---

<sup>6</sup> <https://www.openaire.eu/>

<sup>7</sup> <https://zenodo.org/>

## 5 References

- [1] Bonilla-Calero, A. (2014). Institutional repositories as complementary tools to evaluate the quantity and quality of research outputs. *Library Review*, 63(1/2), 46-59.
- [2] Yoon, A. (2014). End users trust in data repositories: definition and influences on trust development. *Archival Science*, 14(1), pp. 17-34.
- [3] Kelty, C. M., Burrus, C. S., & Baraniuk, R. G. (2008). Peer review anew: Three principles and a case study in postpublication quality assurance. *Proceedings of the IEEE*, 96(6), 1000-1011
- [4] Plant, R. (2004). Online communities. *Technology in society*, 26(1), 51-65.