Towards a Shared European Logistics Intelligent Information Space

The SELIS Community Node

Ioannis Konstantinou, CSLab, ICCS

Antonis Mygiakis, CLMS

Nikodimos Provatas, CSLab, ICCS

Evdokia Kassela, CSLab, ICCS

Tasos Bakogiannis, CSLab, ICCS

IPIC 2019

10th July 2019, London, UK

The SELIS case

A framework to promote collaboration

- Data sharing
- Infrastructure and technology sharing
- State-of-the art innovations

Streaming/Static Data in SELIS

- Logistics documents
- Sensor data (AIS, GPS)
- External sources of information (weather conditions, etc.)
- Business operation workflows

The objectives

- Streamline logistics workflows
- Introduce collaborative tools to improve planning
- Optimize resource management
- Increase supply chain visibility
- Measure quality and efficiency of services provided
- Optimize in terms of environmental impact – Green logistics

The challenges

- Significant discrepancies in technological adoption between stakeholders
- Cloud solution adoption barriers
- **Long list of use case vs Need to develop a single solution**

The SELIS case

Presenting the SELIS living labs and use cases

SELIS Business partners, use cases and datasets

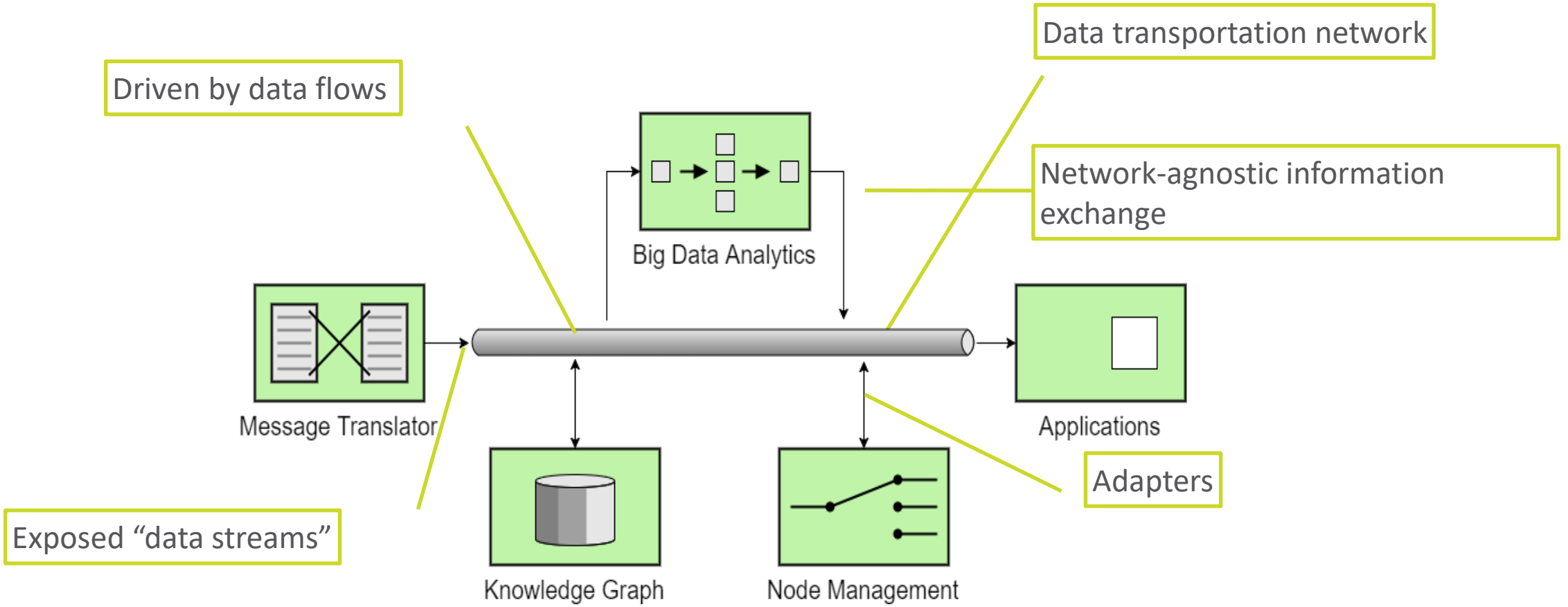
	DHL	Port of Rotterdam	SARMED	SUMMY	North Germany Hinterland Hub	Adria Kombi	European Customs	ELGEKA	SONAE
Business location									
Main modality									

SELIS Data Sets Overview

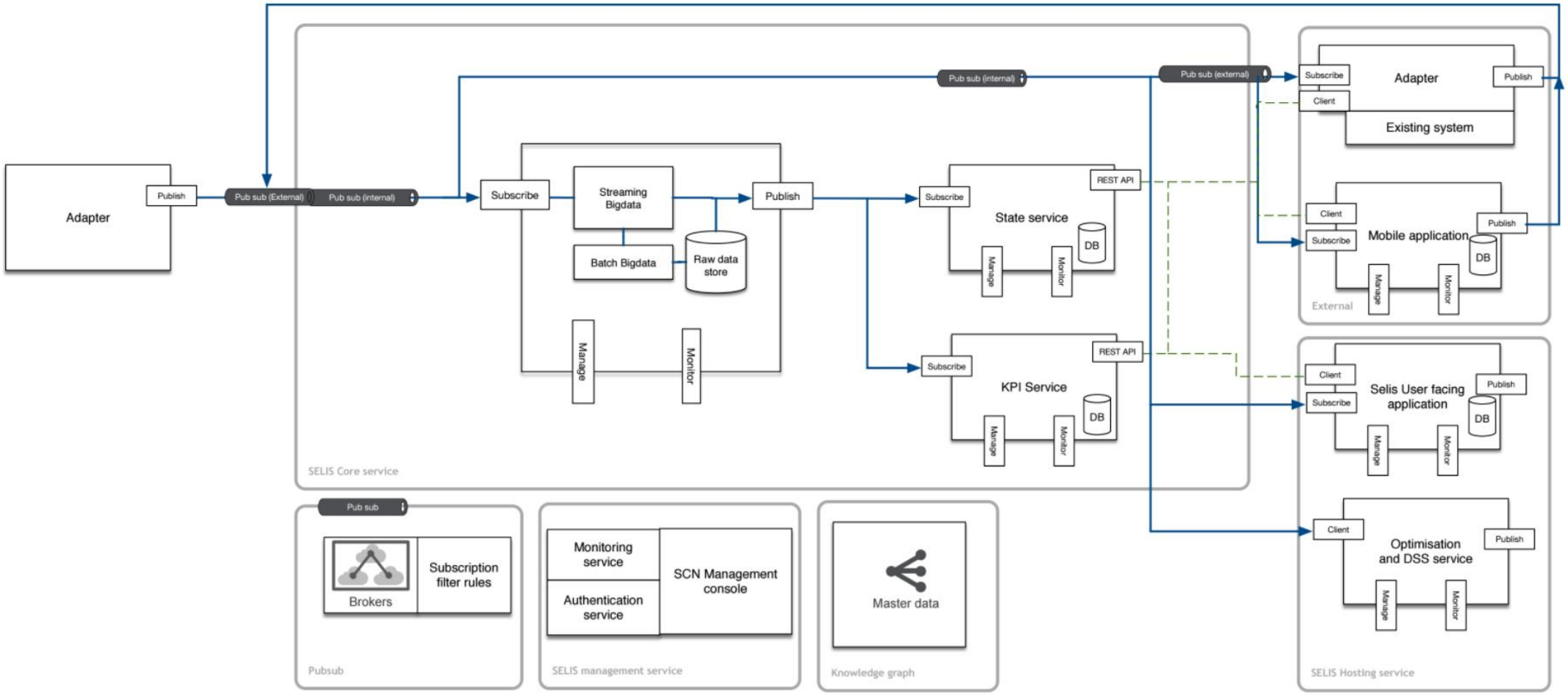
Description	Orders details and status, including Customer, Dispatcher Vehicle, Route, Warehouses and the Suppliers, Region Agencies, Warehouses, Suppliers, Items (SKUs), Orders, Warehouses Stock per SKU, Sales Forecast, Minimum Order Quantity per SKU, Lead Time Delivery Days, Past Sales, Barge information and GPS, Terminal geolocation information, Barge inland terminal visits, Barge deep sea terminal visits , Wind speed and direction data, scheduled train itineraries as a timetable, wagon state, train state, Container Information, customer orders, P.O.D status, Barge voyage, Terminal geolocation information, Fuel consumption data, ++
Origin	LL3 (3), LL8_1(9), LL8_2(6), LL2(6), LL5(5), LL4(4), ++
5Vs of Big Data for Logistics	
Volume	Ranging from few records to tens of GB
Velocity	Up to ~M records/year
Variety	30+ Datasets
Veracity	Empty/missing/wrong values
Value	Increased prediction in ETA (LL5), PoD quality (LL8), +++

Abstract Data Flow Architecture

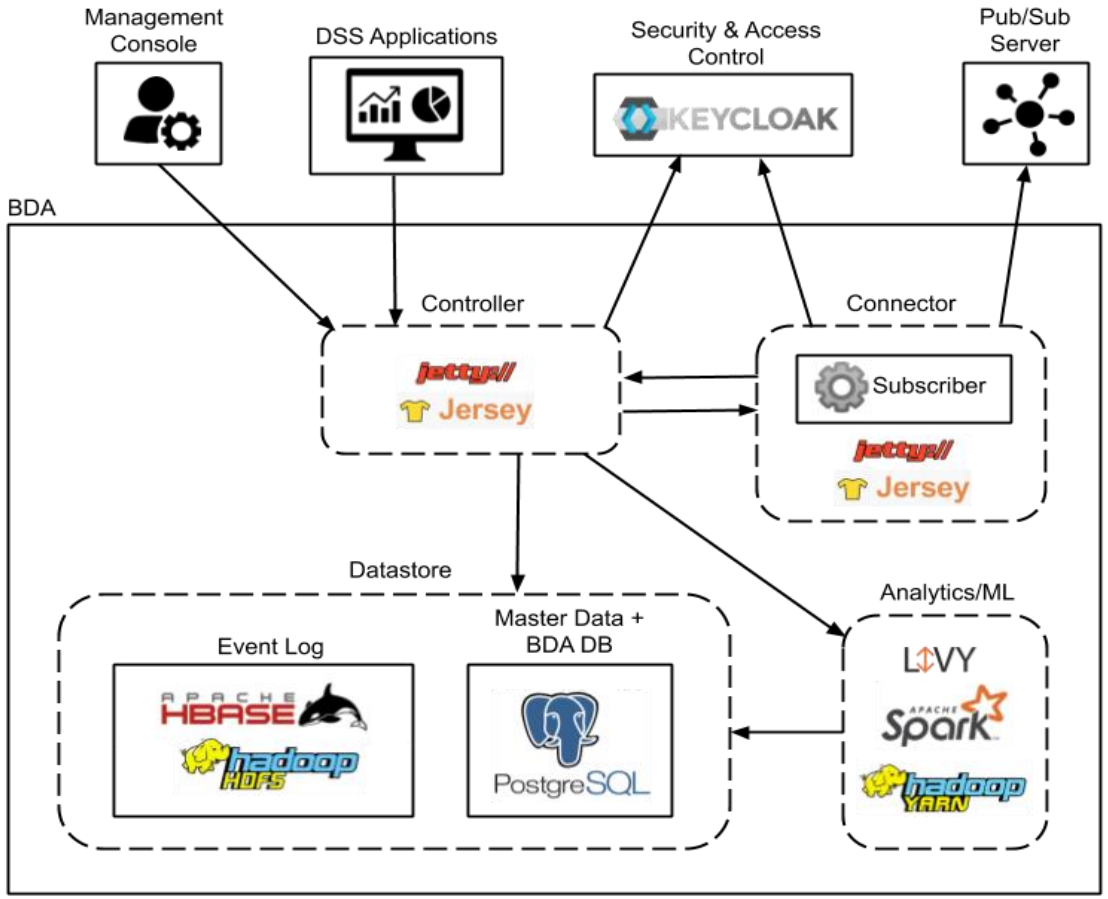
- Abstract Architecture



Detailed SELIS Community Node Architecture



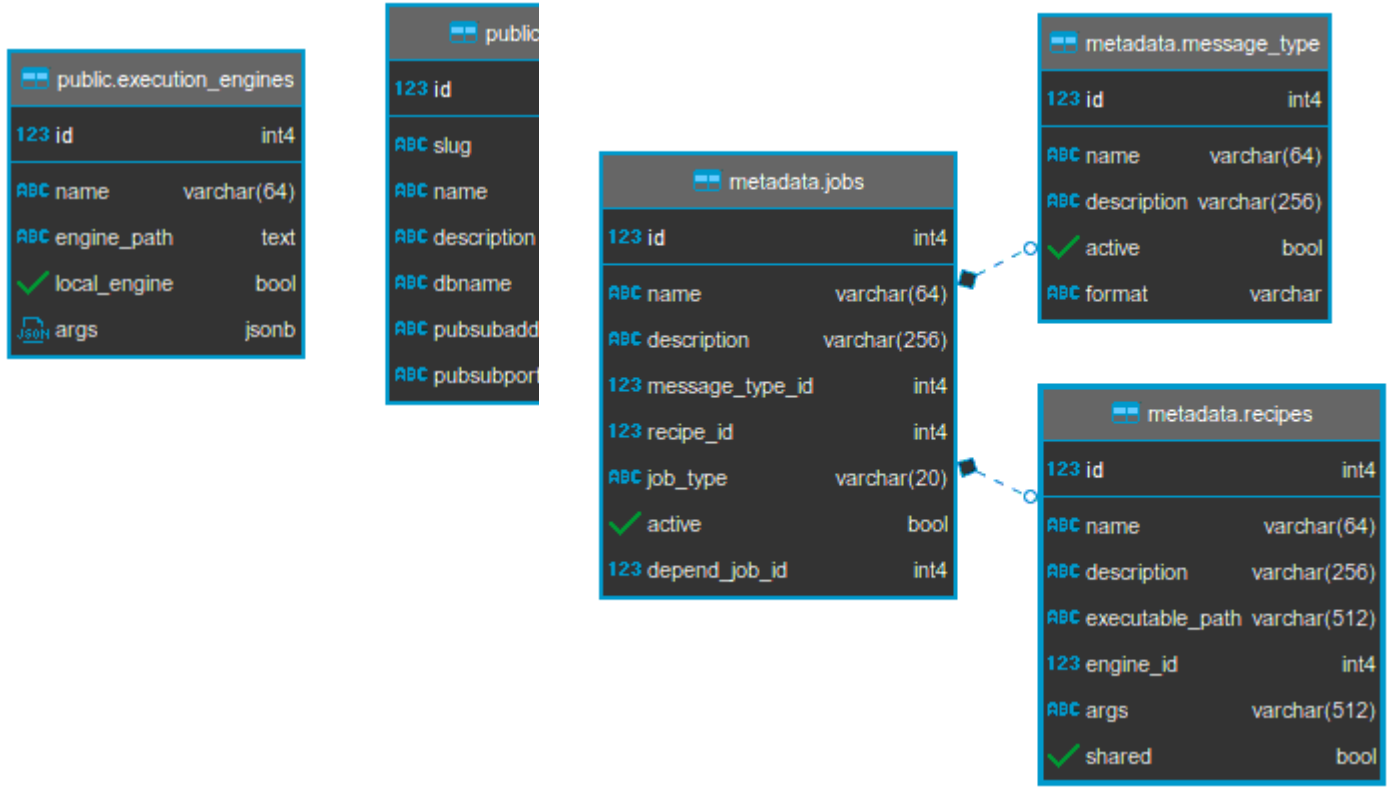
Big Data Enabled Architecture



- BDA (Big Data Analytics) Module
- Includes various subcomponents
- Communicates with external systems
- Docker Image for each sub-component

Data Storage (I)

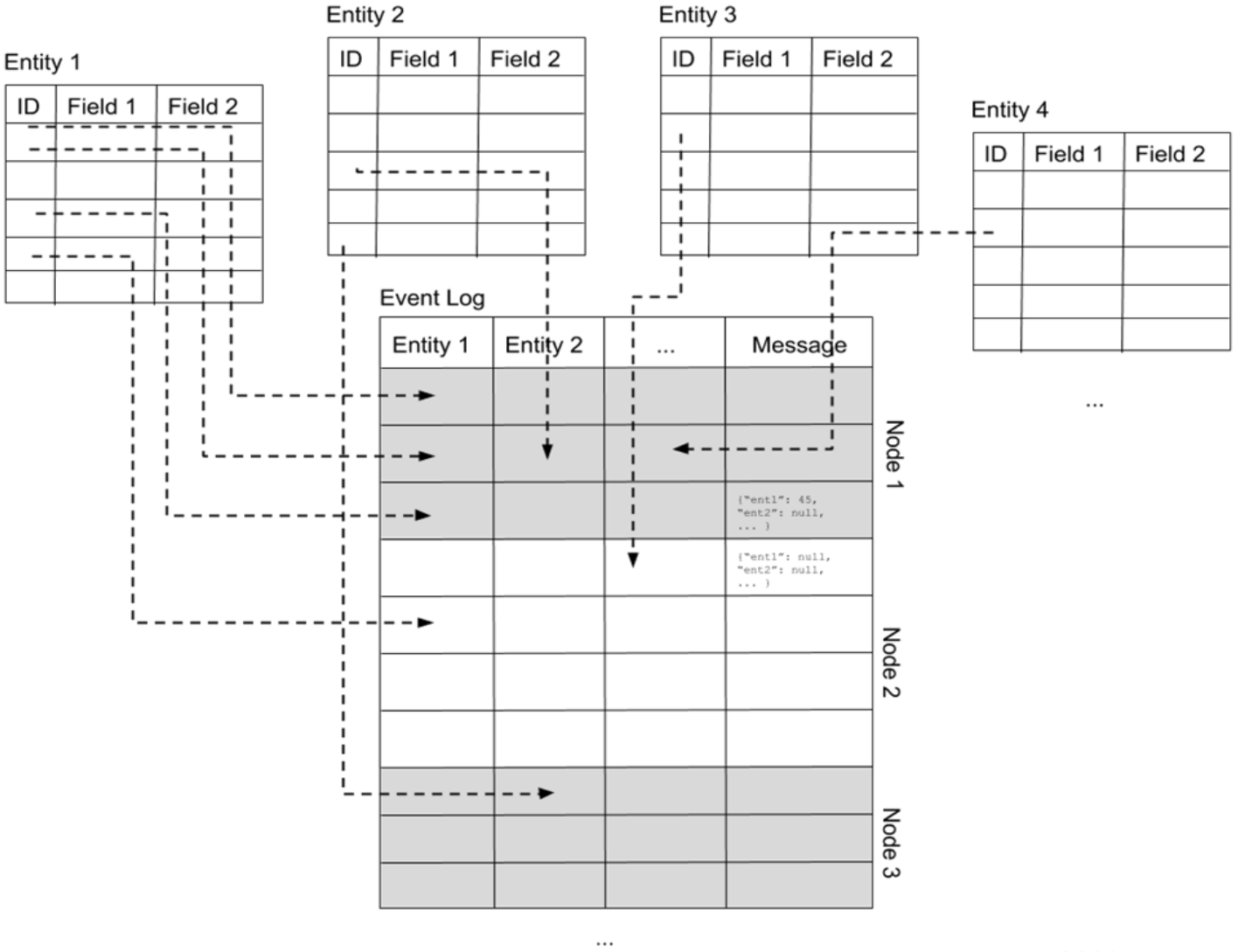
- PostgreSql Database
  - BDA DB
    - Execution engines
    - Shared recipes
    - SCN info
  - Metadata schema for each LL DB
    - Message Types
    - Jobs
    - Recipes



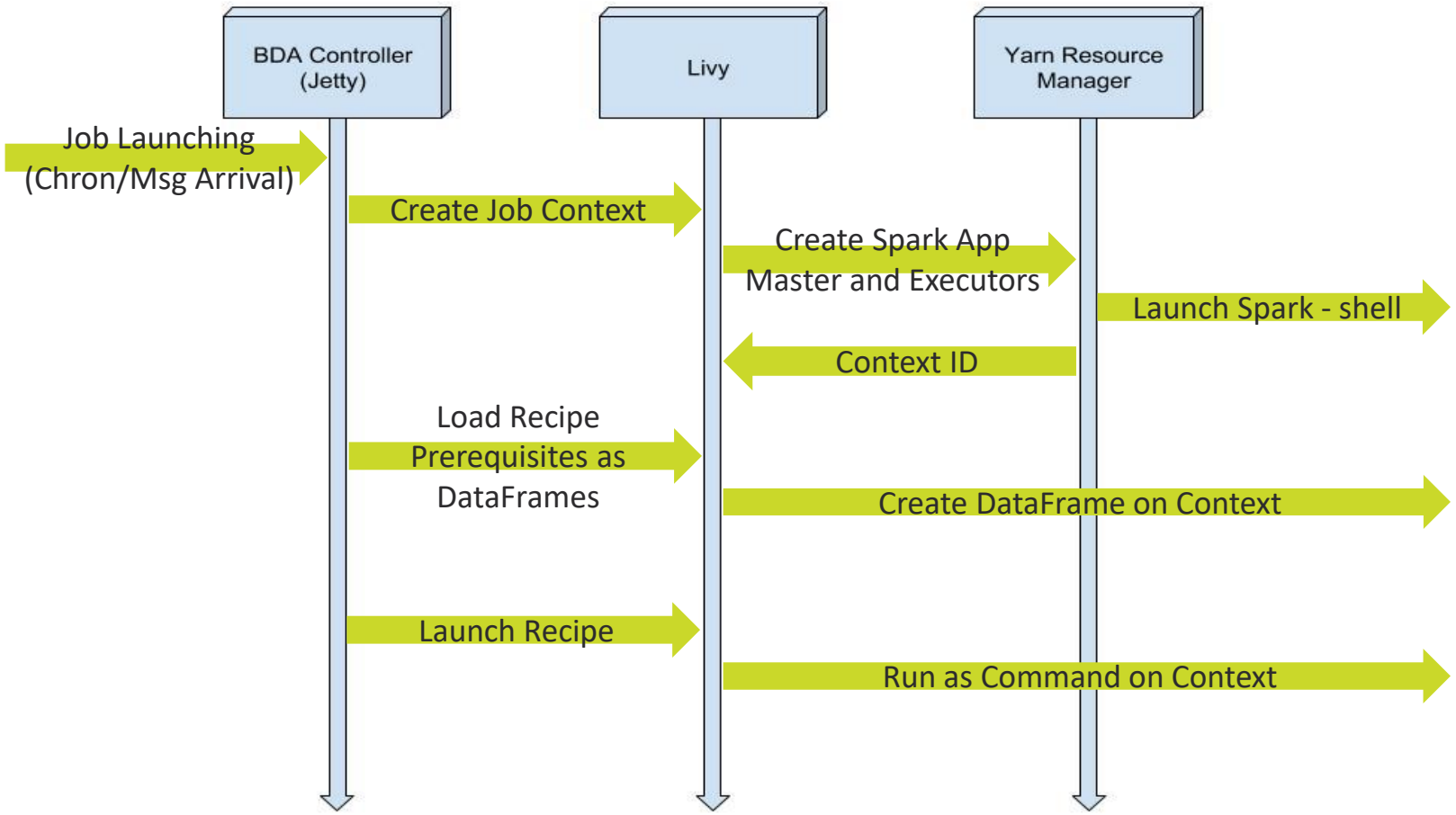
Data Storage (II)

- Generic Approach Supporting many LLs
- Different DB for each LL . Contains:
  - Metadata Schema
  - Dimension Tables (Generally static LL Data)
- Messages Stored on HBase Table (Event Log)
  - Different Namespace for each LL
- Can be created dynamically using BDA REST API at LL Bootstrapping
  - Internally using Java Connectors to create HBase/Postgres Tables
  - Developers can use custom Java connectors for other Systems (e.g. Cassandra, MySql, …)
  - REST Calls for Message Type, Recipe, Job creation

SELIS Star Schema



Recipe Execution



BDA Code Location

- Java code organized in 2 Gitlab projects with various submodules

https://selis-gw.cslab.ece.ntua.gr/gitlab/selis/bda

Big Data Analytics and Machine Learning System

SELIS Big Data Analytics and Machine Learning System

Star 0 | Fork 0 | SSH | git@selis-gw.cslab.ece.ntua.gr:se | Global

Files (5 MB) | Commits (356) | Branches (14) | Tags (0) | Readme | Add Changelog | Add License | Add Contribution guide | Set up CI

Name	Last commit	Last update
bda-analytics-ml	fix compilation errors	2 weeks ago
bda-controller	fix subscription info for external connectors	a day ago
bda-datastore	added new rest call to create message service and added m...	2 weeks ago
common	set external properly in the bean.	2 weeks ago
conf	use external subscriber for messages that need to be sent to...	2 weeks ago
docker	use external subscriber for messages that need to be sent to...	2 weeks ago
examples	Merge branch 'bda_external_connectors' of selis-gw.cslab.ec...	2 weeks ago
kpi-db	update kpi table schema to use correct message columns	a month ago
.gitignore	ignore bin folders	9 months ago
README.md	move properties file and create template	9 months ago
compile.sh	added mvn clean command to avoid errors appearing on st...	5 months ago
pom.xml	refactor analytics and ml modules with one runner class	4 months ago
run.sh	Separate hadoop, spark, hbase containers working.	3 months ago

https://selis-gw.cslab.ece.ntua.gr/gitlab/selis/bda-subscriber

BDA Subscriber

Star 0 | Fork 0 | SSH | git@selis-gw.cslab.ece.ntua.gr:se | Global

Files (1.9 MB) | Commits (6) | Branches (4) | Tags (0) | Readme | Add Changelog | Add License | Add Contribution guide | Set up CI

Name	Last commit	Last update
bda-subscriber	work with public and secure pubsub server	a day ago
conf	initial on aeolix forwarder.	a month ago
docker	work with public and secure pubsub server	a day ago
.gitignore	first commit of bda subscriber	4 months ago
README.md	first commit of bda subscriber	4 months ago
compile.sh	first commit of bda subscriber	4 months ago
pom.xml	first commit of bda subscriber	4 months ago
run.sh	first commit of bda subscriber	4 months ago

Different connectors developed in different branches!

BDA Code Execution (I)

- Steps for a complete local containerized BDA deployment:

1. Clone the code

> git clone git@selis-gw.cslab.ece.ntua.gr:selis/bda.git

> cd bda

> git clone git@selis-gw.cslab.ece.ntua.gr:selis/bda-subscriber.git

> cd bda-subscriber

2. Create the configuration based on the provided templates

> cp conf/bda.properties.template conf/bda.properties

> cp conf/bda-subscriber.properties.template conf/bda-subscriber.properties

3. Launch the containers

> cd docker

> ./sls.sh run all

> cd docker

> ./sls.sh run subscriber

After this we are logged in inside the selis-controller and the selis-subscriber containers that contain the code.

4. Compile the code and run the servers (inside the containers)

> ./compile.sh && ./run.sh

> ./compile.sh && ./run.sh

The BDA controller API is available on localhost:9999

BDA Code Execution (II)

- Curl examples exist in the bda/examples folder for using the BDA REST API:

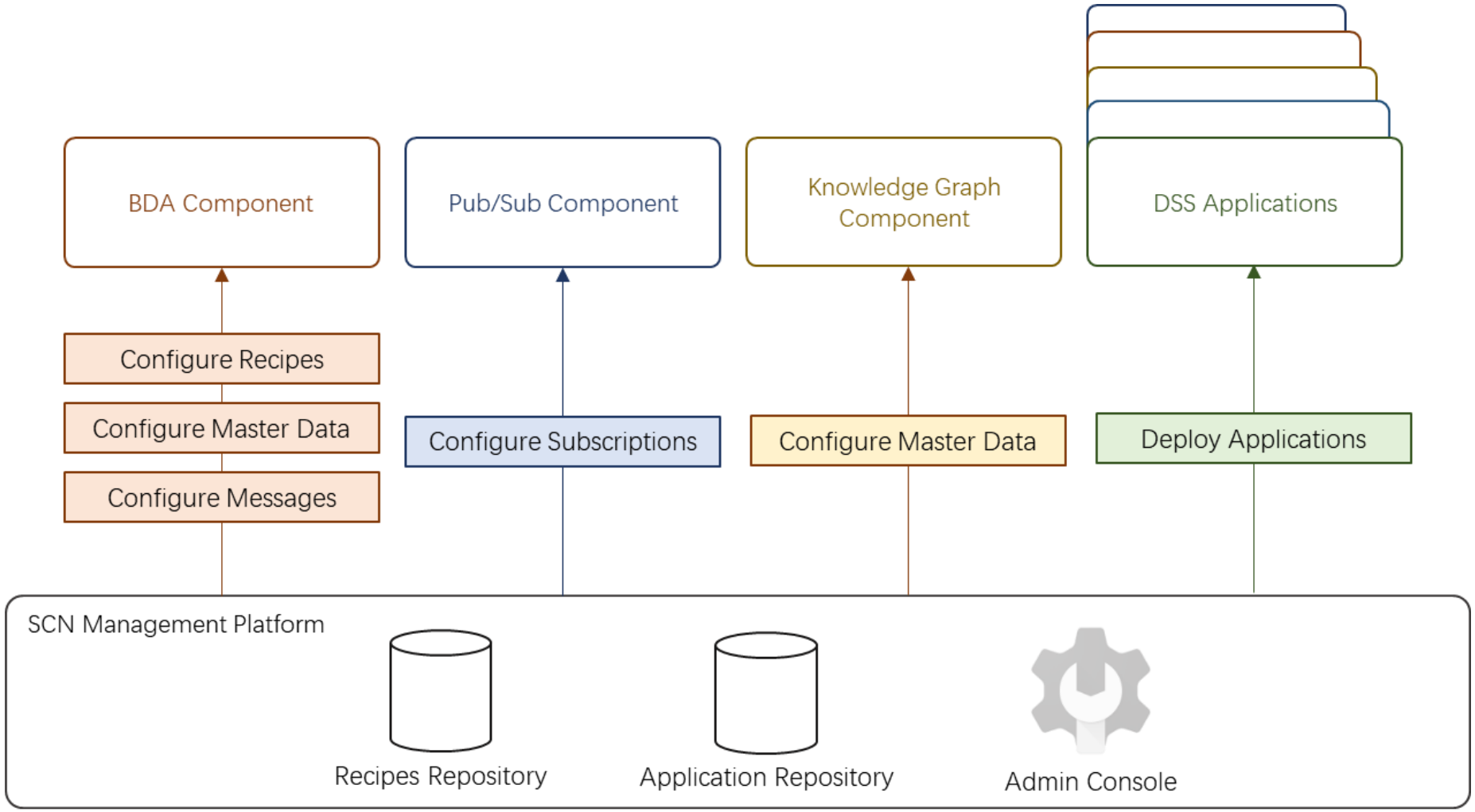
1. Create a connector: *curl -ik -X POST -H "Content-type:application/json" -H "Accept:application/json" --data @newconnector.json http://localhost:9999/api/connector/create && echo*
2. Create the databases for a new SCN: *curl -ik -X POST -H "Content-type:application/json" -H "Accept:application/json" --data @newscn.json http://localhost:9999/api/datastore/create && echo*
3. Create a message type to subscribe to: *curl -ik -X PUT -H "Content-type:application/json" -H "Accept:application/json" --data @msgtype.json http://localhost:9999/api/message/{scn_slug} && echo*
4. Create a recipe:

   curl -ik -X PUT -H "Content-type:application/json" -d @recipe.json http://localhost:9999/api/recipe/{scn_slug}/

   curl -ik -X PUT -H "Content-type:application/octet-stream" --data-binary @recipe.py http://localhost:9999/api/recipe/{scn_slug}/upload/{recipe_id}/recipe.py
5. Create a job: *curl -ik -X PUT -H "Content-type:application/json" -H "Accept:application/json" --data @job.json http://localhost:9999/api/job/{scn_slug} && echo*

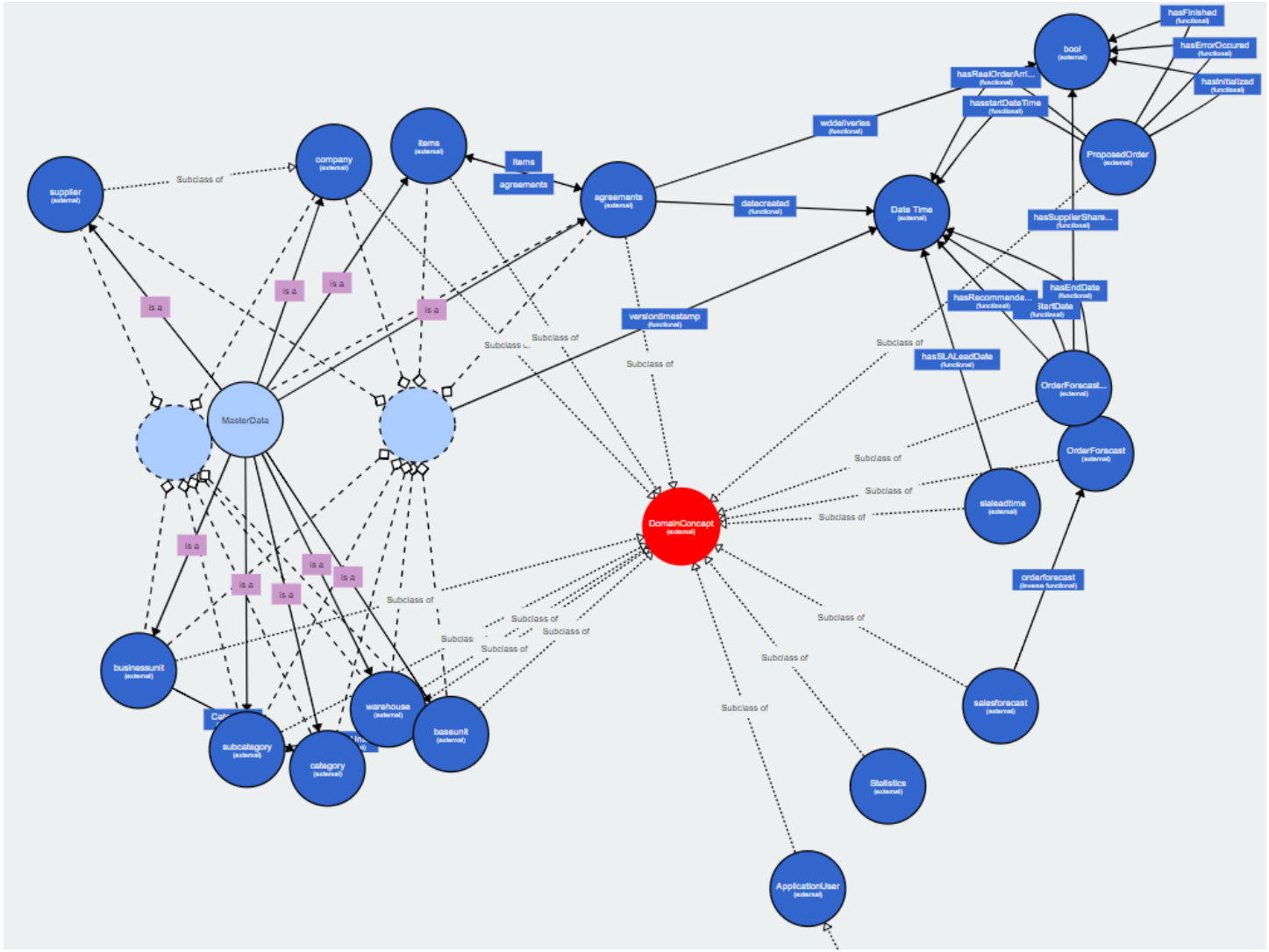
SCN Management – GUI

SELIS Community Node – Extensible Model



The project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under the Grant Agreement No 690588

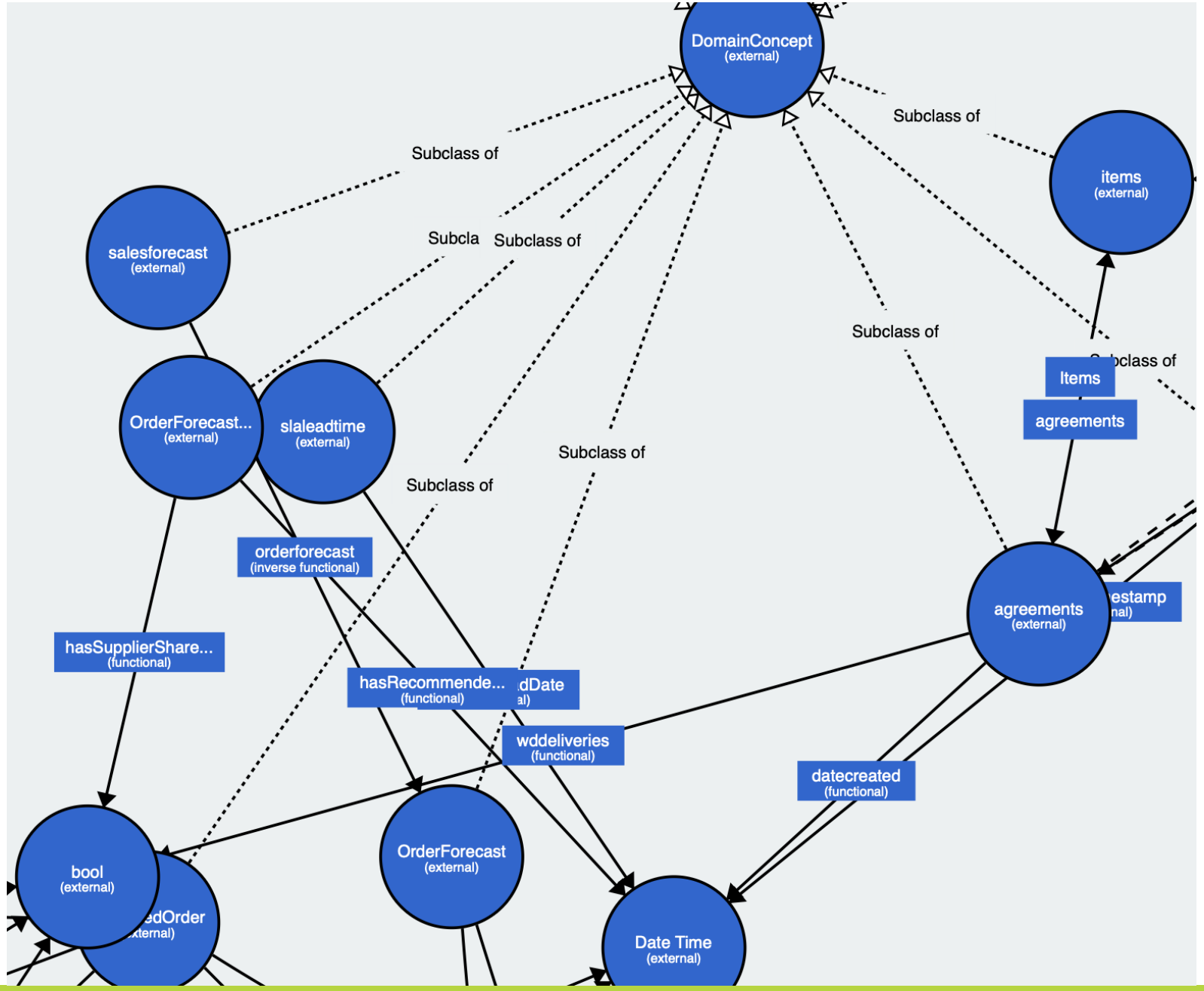
SELIS Community Node – Extensible Model



The project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under the Grant Agreement No 690588

SELIS Community Node

Extensible Model

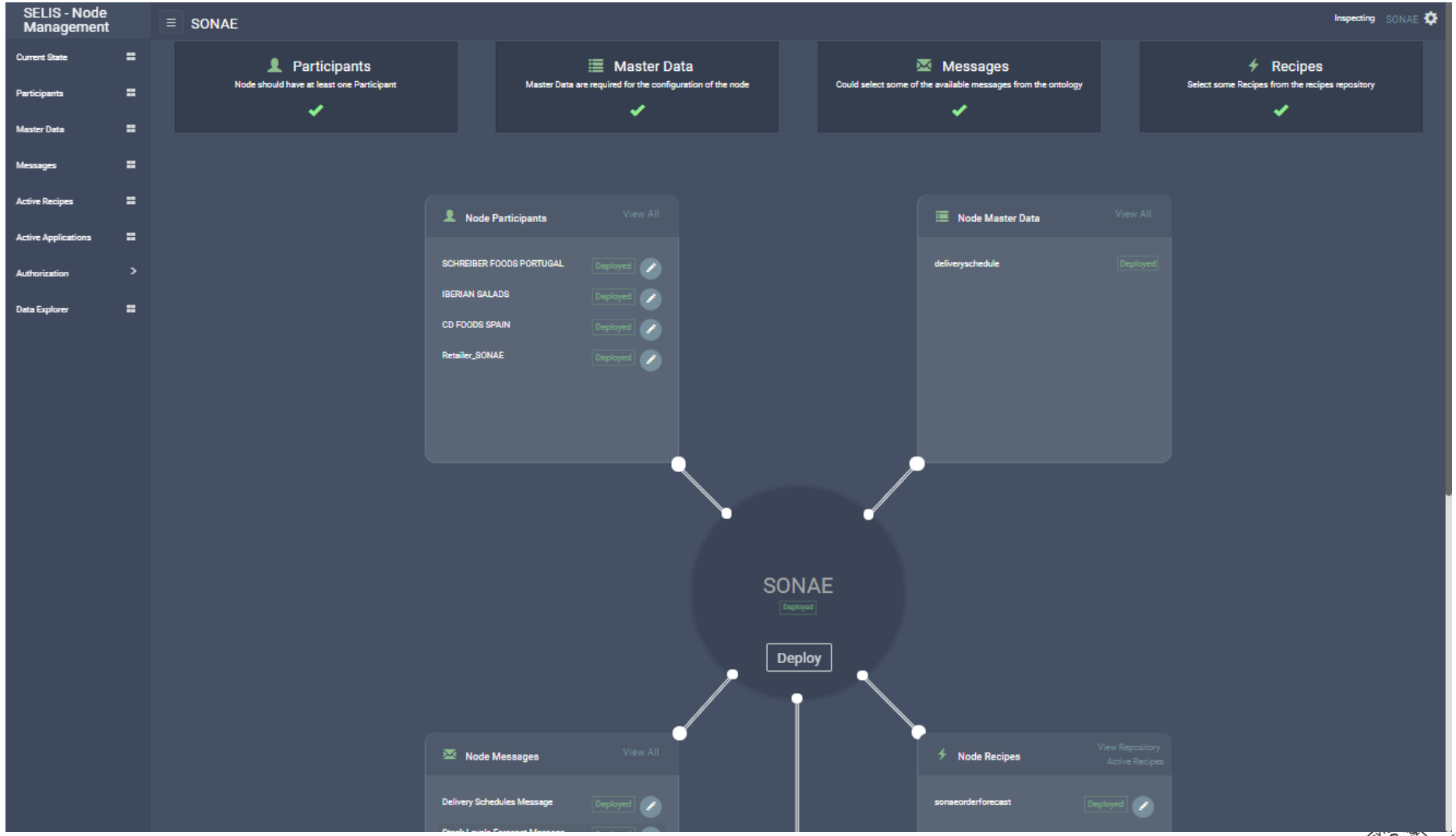


The project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under the Grant Agreement No 690588

SCN Management Platform – Main Control Panel

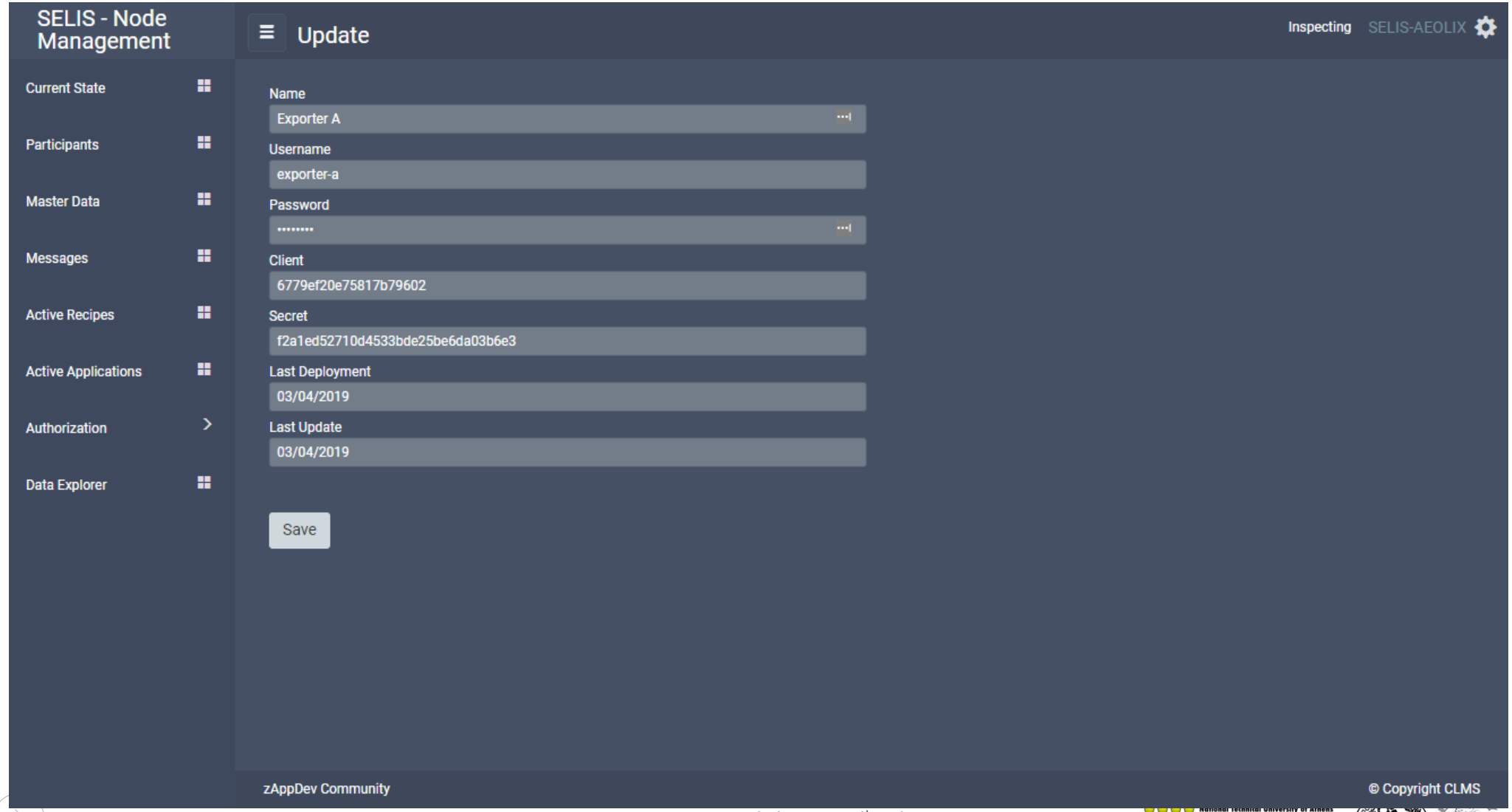
The project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under the Grant Agreement No 690588

SCN Management Platform – Node Overview



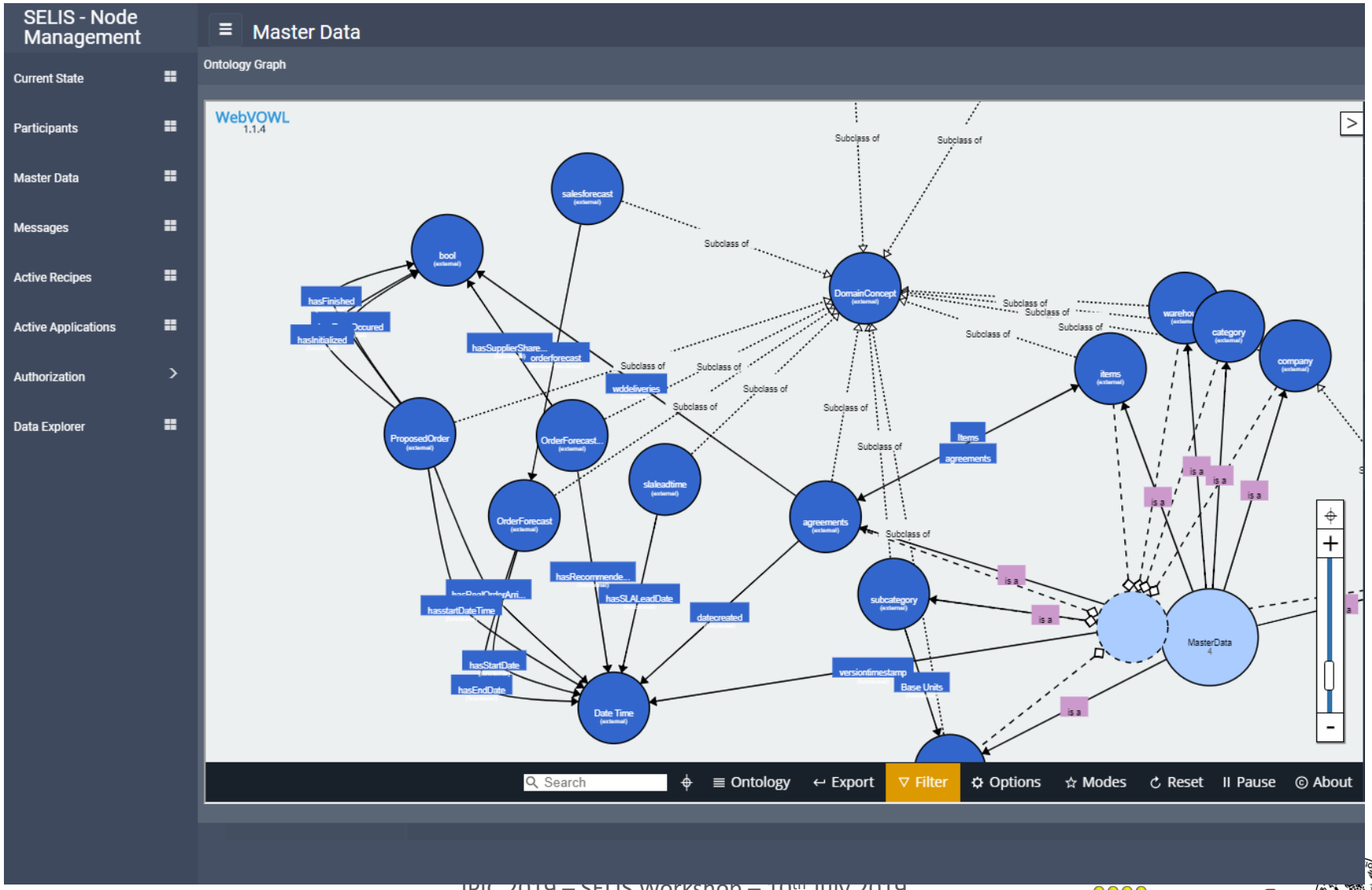
The project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under the Grant Agreement No 690588

SCN Management Platform – Participants



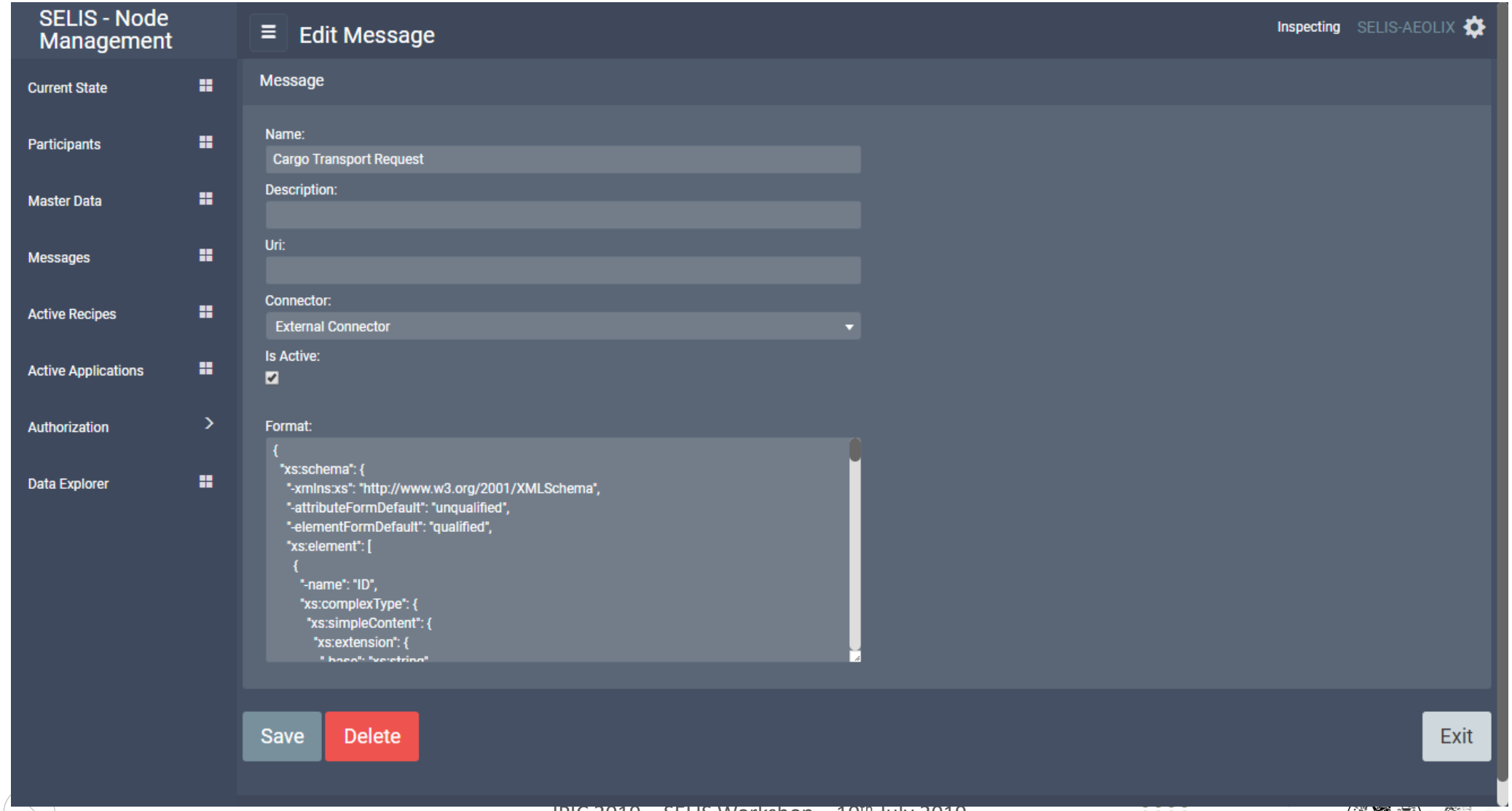
The project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under the Grant Agreement No 690588

SCN Management Platform – Master Data



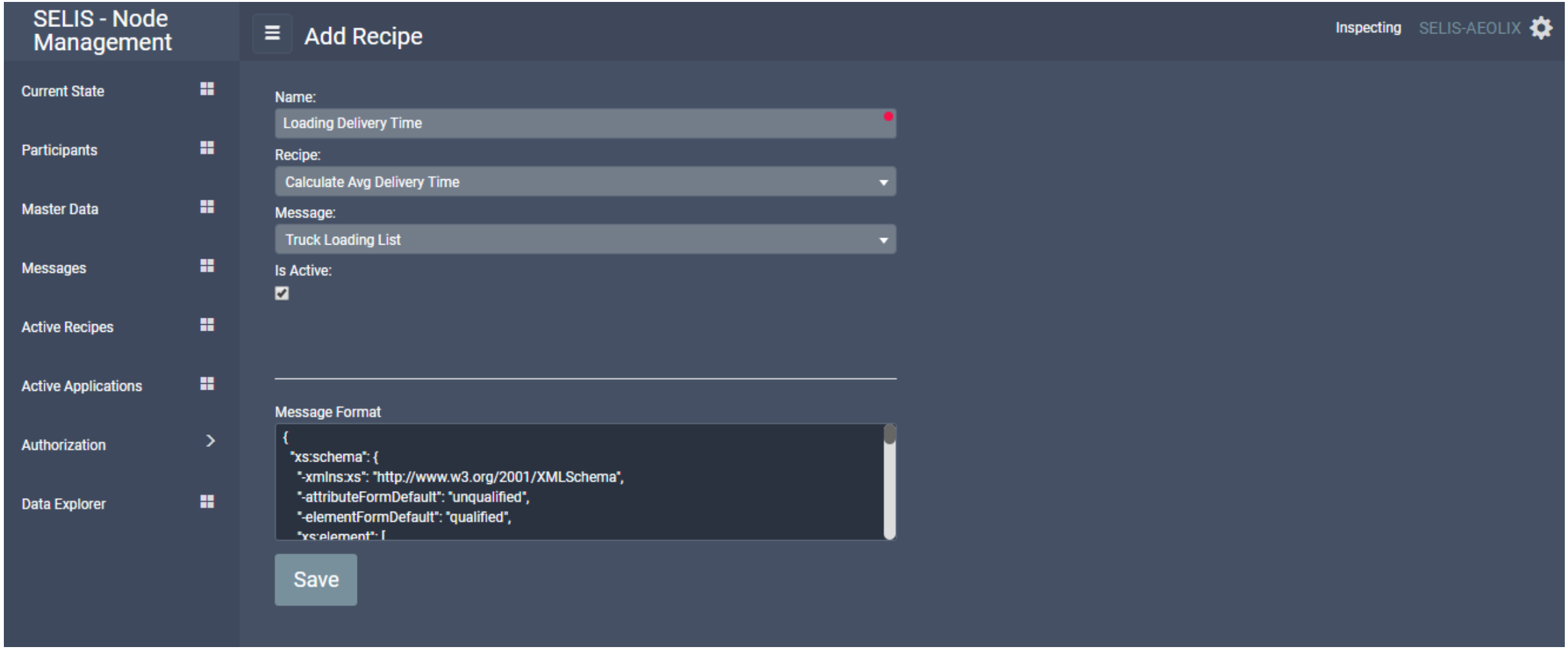
The project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under the Grant Agreement No 690588

SCN Management Platform – Messages



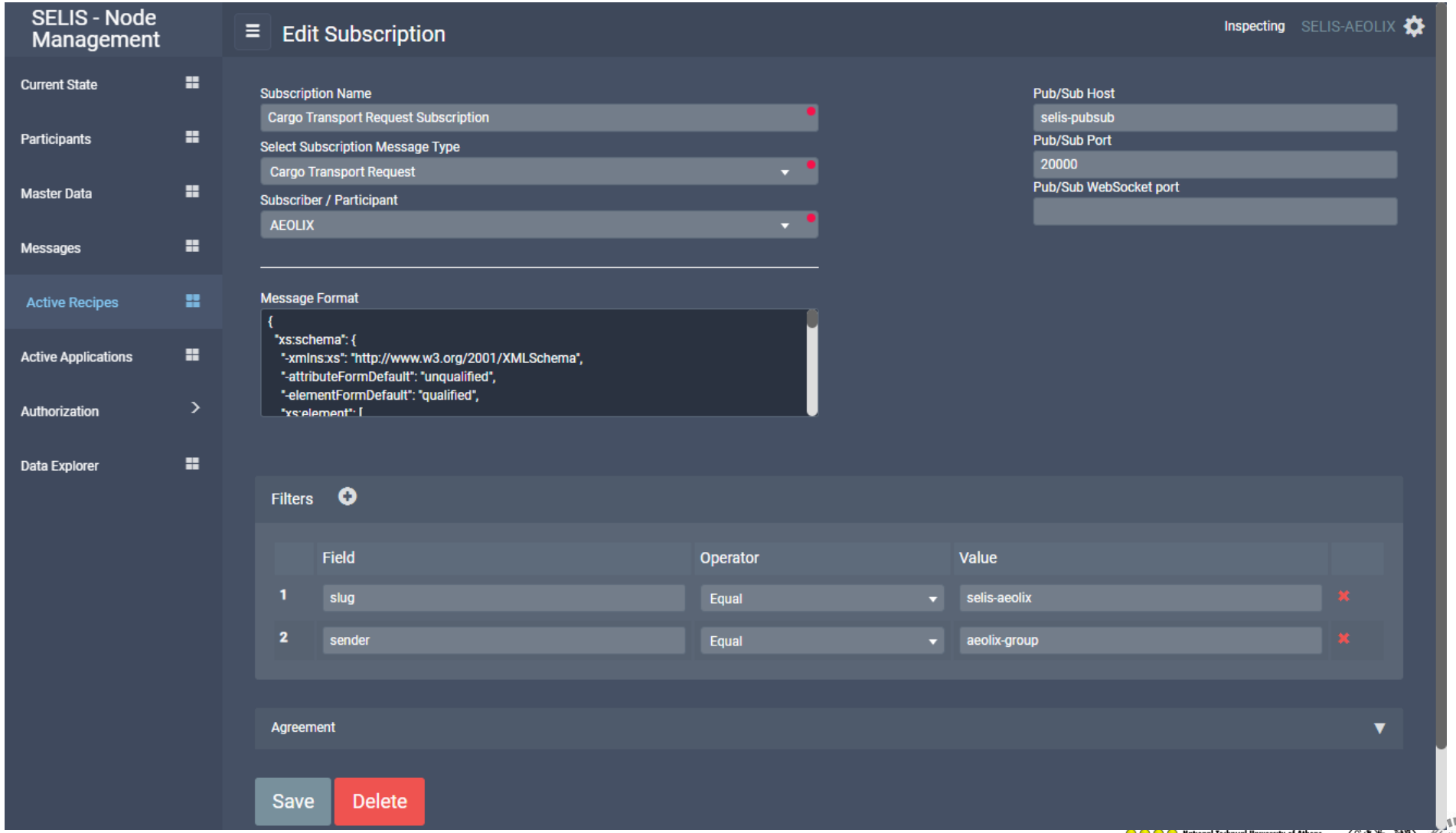
The project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under the Grant Agreement No 690588

SCN Management Platform – Recipes and Jobs



The project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under the Grant Agreement No 690588

SCN Management Platform – Publications and Subscriptions



The project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under the Grant Agreement No 690588

Data Governance in SELIS

Architecture decisions and Information Governance

- SELIS Community Node architecture specifics:
  - Access Control through Node Management: Transient resources are tagged based on the owner and content and are accessible only as permitted by rules set during setup of the Node.
  - Authorized consumers: Data is never accessed directly by consuming applications. All requests are handled by the SCN services that handle client authorization accordingly.

The project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under the Grant Agreement No 690588

Data Governance in SELIS

Aligning individual corporate policies in a multitenant platform

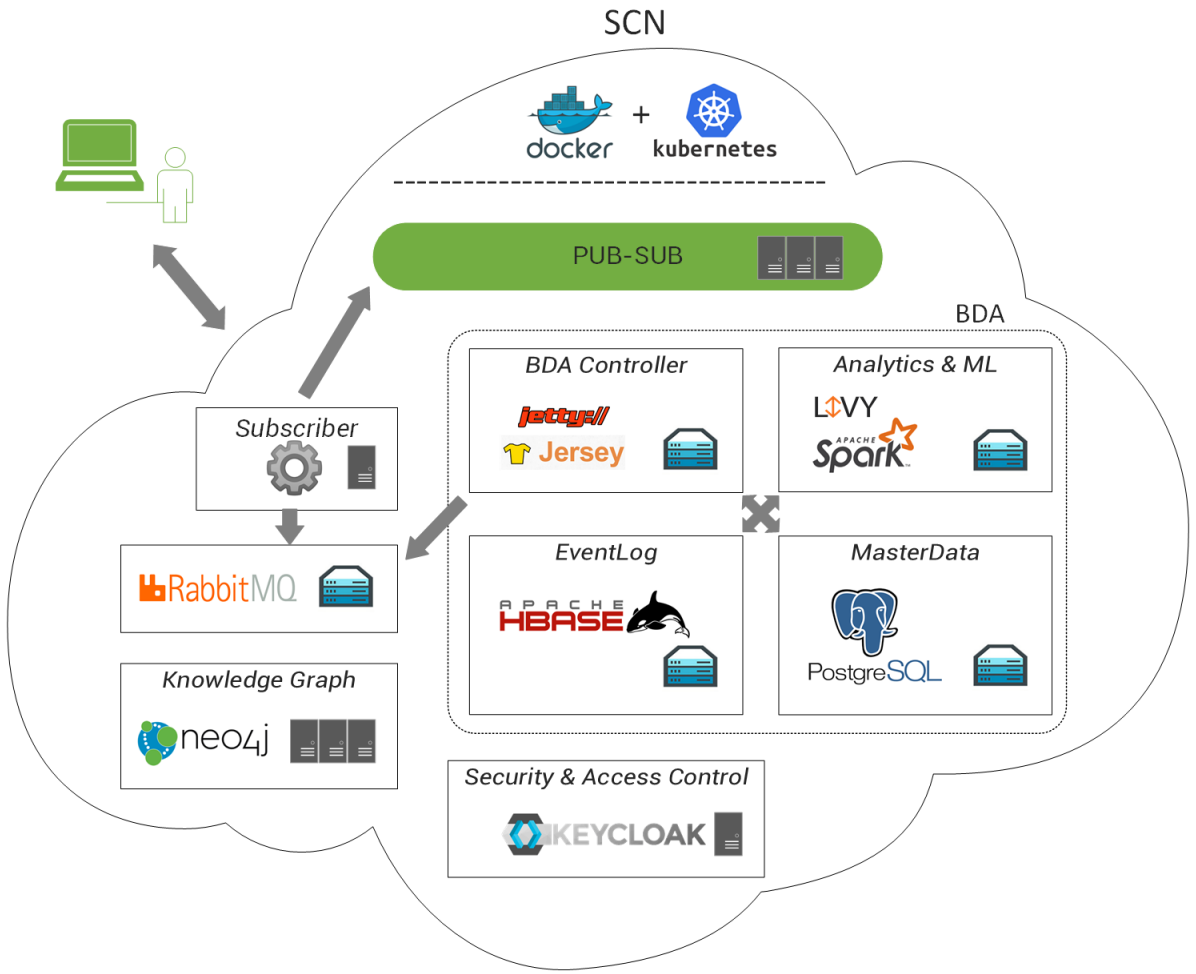
- Dynamic **configuration** and **application** of every stakeholder's **Data and Information Governance policies**

  - Protecting digital assets is a top priority.
  - SELIS supports the definition of generic role and permission-based DG processes and policies.
  - SELIS uses content-based message delivery, which can support more complex DG configurations.
  - A flexible ontology-based model is used to tackle Real-life application scenarios which often result in complex, potentially conflicting workflows

The project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under the Grant Agreement No 690588

Build on Open Technologies

- Based on open-source technologies
- Cloud-ready over Docker containers and kb8etes
  - Easy install on both on-prem and public cloud
- Scalability and elasticity
- To be released as open-source



Let’s Put everything to the cloud!!!!


Q&A

Contact Details

ICCS

Ioannis Konstantinou

ikons@cslab.ece.ntua.gr