Eddie Olsson, RISE, 2023-01-16

Deplide for FEDeRATED

The purpose of this document is to describe how Deplide supports the needs of several Living Labs in FEDeRATED. Deplide is a data-agnostic demonstration platform for ecosystem innovation in multi-modal transport chains. Deplide is based on solid experience from similar platforms in several large-scale projects within the maritime sector adapted to more generic multi-modal transport needs. It has been developed primarily to support data sharing around a transport node, where main events along the supply chain occur. In addition, Deplide can collect and share data from several nodes along the supply chain to increase end-to-end supply chain visibility and similar use cases.

Deplide builds on earlier work with data sharing platforms for Collaborative Decision Making. As a result of its legacy, it specifically supports distribution and aggregation of data from a variety of sources with local object identifier coordination to allow for flexibility and adaptability. This supports building visualisations using data from Deplide by simplifying correlation between data sources.

In FEDeRATED, Deplide is extended to support the needs of the Living Labs. It implements the FEDeRATED basic architecture by acting as one or more data sharing nodes. More concretely, Deplide implements the FEDeRATED technical specifications:

- semantics (described in Data layer and Semantics)
- IAA (described in Communication layer and IAAA)
- Service Registry (described in Service Registry)
- Index (described in Index)

Communication layer and IAAA

There are two main technical solutions for interacting with Deplide for submitting and fetching data:

1. REST API endpoints. This works well with existing systems and allows for distribution of data with minimum effort for integration with other systems
2. SPARQL endpoint which supports a richer data exchange from a single URL endpoint

All endpoints will be secured using OAuth 2 with a Keycloak-based identity provider. Identification, authentication, and authorisation is based on the mechanisms made available through Keycloak. Technical implementation for Access management will be selected in Q1-Q2 of 2023.

Identity broker functionality can be made available for increased flexibility.

All communication between systems uses HTTPS with authentication designed according to the proposed model in FENIX.

Data layer and Semantics

Incoming data will be stored in a triple store and made available through a SparQL endpoint via a semantic adapter. Technical platform for both is to be decided during Q1 2023 with implementation following shortly.

Processing Layer

Deplide's processing layer consists of Kafka Stream Processors that break down incoming data into atomic events and Stream Processors that aggregate atomic events into operational structures representing trains, trucks, containers and other carriers. Stream processors are also used for data aggregation, filtering and transformation of incoming data before distributing to both internal and external systems, including the triple store.

Two types of aggregation are implemented internally in Deplide to help build end user visualisations consistently:

- Builders – Builds data structures representing common objects, physical or virtual, that are central to transport and logistics operations such as Trains, Port Calls, Containers or Shipments
- Indicator system – Generates indicators and warnings based on knowledge of and expectations for relationships between parts of the transports and logistics operations

Deplide architecture from a FEDeRATED perspective

All deplide components run inside a Kubernetes cluster and are only accessible through a forward proxy which implements URL-to-microservice mapping, authentication and access control on a microservice level. Identity provision will be managed by a keycloak instance, giving us a full Oauth2/OpenID Connect stack for authentication and identification. Incoming data are either pushed to Deplide REST endpoints or pulled using Kafka Connect Source Connectors. Outgoing data is pushed to Ahola's Flow Manager using Kafka Connect Sink Connectors.

REST Push to Deplide

SIMPLE

SIMPLE delivers data to Deplide in RCMF, a railway-specific JSON format, to Deplide for use in LL#5 RFID in rail

STA RFID

STA the Swedish Transport Administration, delivers data in XML format according to an extended EPCIS schema to Deplide for use in LL#5, LL#6 RRTCDM and LL#23 Ahola Driver

Alleima

Alleima delivers data in JSON-LD format and schema.org schemas for use in LL#8 MMIS3

REST Push from Deplide

Ahola

Deplide delivers three streams of data to Ahola for use in the LL#23 Ahola Driver. Two of these are in RCMF and one is an aggregated format representing trains. All three use JSON. Implementation uses Kafka Connect sink connectors for data filtering, transformation and delivery.

REST/SOAP Pull to Deplide (Using Kafka Connect Source Connectors)

RFID/MQTT

Deplide fetches data from an MQTT broker supplied by the KTH Royal Institute of Technology for use in LL#5, LL#13, LL#14, LL#15, and LL#23

STA API

Deplide fetches data from two distinct APIs supplied by the Swedish Transport Administration using a proprietary HTTP-based API specification. Data from the open API is used in all Swedish LL that use Deplide. No data from the closed API is currently used within the context of FEDeRATED

GeoDis

Shipment tracking data from GeoDis will be delivered using a SOAP endpoint. Data format is proprietary.

SMA

Integration with SMA for LL#8 is still in development.

Persistent Storage

All relevant data will be kept in a triple store that can be queried through a SparQL endpoint to allow for semantic queries of any data supplied by Deplide.

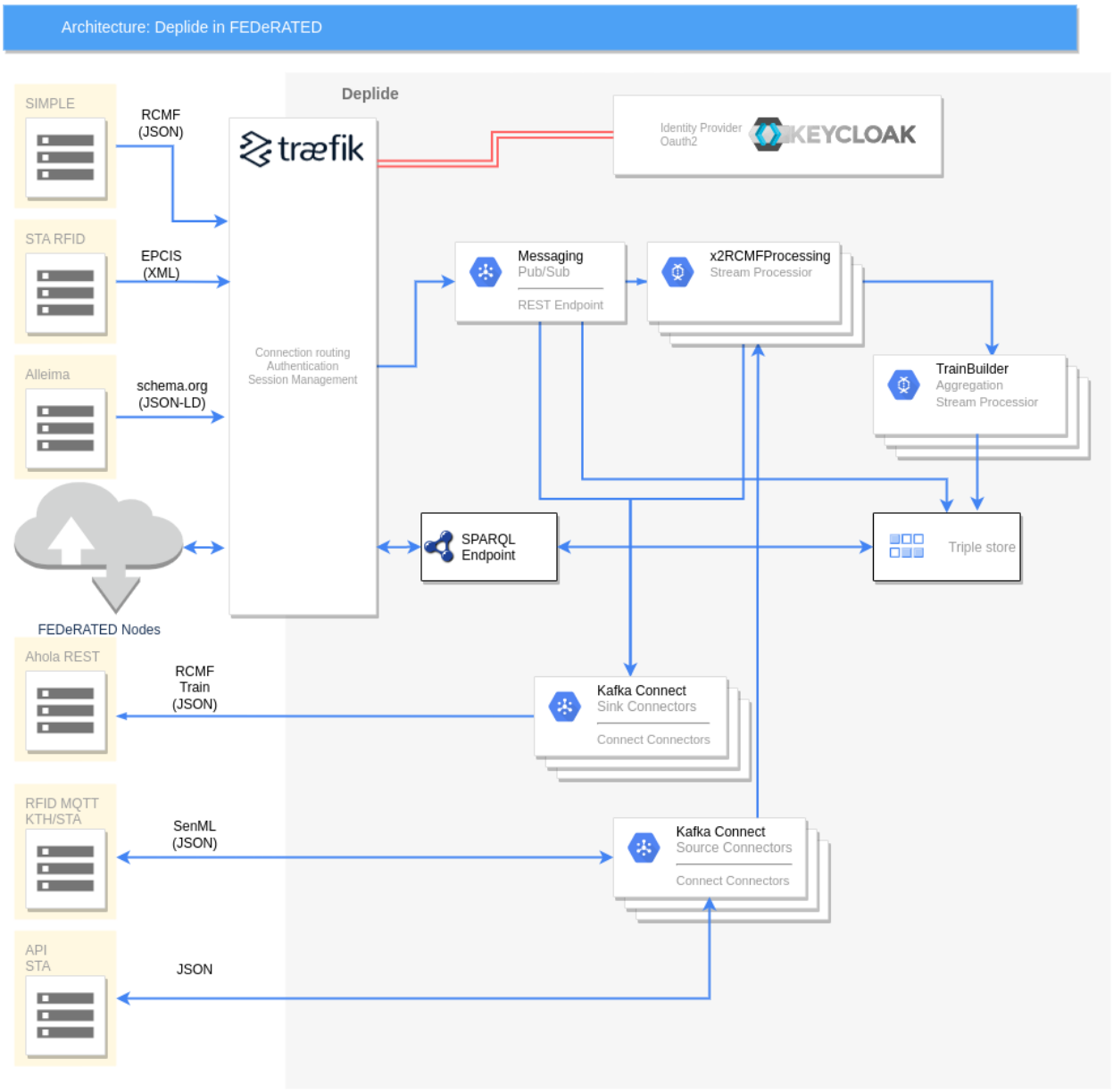

Figure: Graph showing the outline of Deplide's communication architecture as used in FEDeRATED

Service Registry

Initial implementation of Service Registry will be a simple, mostly static, web page with textual descriptions of available services with descriptions, both human and machine-readable, for connection details, including API details, data models and authentication mechanisms used. There will also be a machine-readable list of other service registry nodes so that the service registry as a whole will be distributed by nature. Further development of the Service Registry implementation will be done in cooperation with the other participants in the Service Registry workgroup.

Index

All messages in Deplide are keyed by messageID. Implementation details on the technical solution to retrieve events by ID will be determined during Q1 of 2023.

Conclusion

In conclusion, Deplide is both a data sharing solution and an enabler bringing a network of transport and logistics actors closer together. As a data sharing solution, it involves properties for event streaming and event processing. By providing solutions for Service Registry and Identification, Authentication, Authorisation and Access Management (IAAA), Deplide facilitates and improves security for connectivity between actors in the transport and logistics ecosystem.

Sharing of data is primarily based on publication and subscription according to the pub/sub model. Data consumed from the Deplide environment is often used in front-end applications responding to different use cases within FEDeRATED Living labs but also for deriving foundations for analytic services. Deplide's event processing engine validates incoming data and extracts transport and logistics event timestamps for distribution to facilitate correlation and aggregation of data from various sources, thus improving the quality, granularity, and usability of shared data.

The table below indicates which aspects of Deplide are validated in the respective FEDeRATED Living labs

	Data sharing solutions				Network mechanisms	
LL #	Event Streaming	Event Processing	Object Identification	Audit Logging	Service Registry	IAAA
5 RFID in Rail	X	X		X	X	X
6 RRTCDM	X	X	X	X	X	X
7 RTPVS-SMA	Publishes data to Deplide					
8 MMIS3	X	X	X	X	X	X
13 BetTerFlow	X	X	X	X	X	X
14 SIMC	X	X	X	X	X	X
15 OptiPort	X	X	X	X	X	X
21 SIMPLE	Publishes data, used in LL #5, to Deplide					
23 Ahola Driver	Exchanges data between Deplide and Flow Manager					