“Enhanced data management techniques for real time logistics planning and scheduling”

Deliverable D2.2: Specifications of the data collections and enrichment pipelines

Dissemination level:
☒ Public ☒ Confidential, only for members of the consortium (including the Commission Services)

Version number: 1.0

Submission deadline: 31/05/2019

www.LOGISTAR-project.eu

DOCUMENT INFORMATION

Authors

Name	Organisation
Martin Kaltenboeck	SWC
Nenad Gligoric	DNET
Jürgen Jakobitsch	SWC

Reviewers

Name	Organisation
Bastien Pietropaoli	UCC
Enrique Onieva	Deusto
Naia Merino	Deusto

Document control

Version	Date	Comment
0.1	April 2019	Initial setup of document
0.2	21/05/2019	Refinements and expansions
0.3	22/05/2019	Section 2.2 added
0.3	22/05/2019	Refinements by wp2 team of whole document
0.4	24/05/2019	Data Handling Section expanded
0.5	27/05/2019	Review and expansion of several sections to create review version. Update of all figures, list of abbreviations and references
0.6	28/05/2019	Reviewed version including comments et al
0.7	30/05/2019	Final version
1.0	31/05/2019	All partners

Document approval

Version	Date	Partners
1.0	31/05/2019	All partners

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

BASIC PROJECT INFORMATION

Horizon 2020 programme

H2020 - Mobility for Growth- 5-2-2017. Innovative ICT solutions for future logistics operations

Grant Agreement No. 769142

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

Executive Summary

This document describes the mechanisms (algorithms, components, workflows) for **handling the large-scale, multi-faceted, real-time and noisy data** found in real logistics networks. We describe the specified output format of the data, in order to enable the structuring and speeding up of computations in the machine learning and optimization tasks. **This will be done through effective data annotations, pre-processing and APIs**.

Furthermore, we describe the **handling of metadata** as well as input and output specifications that will be based on requirements and specified services and will take into account work from different work packages, thus providing the envisioned functionalities and support or all WPs with focus on the capabilities required for data collection (WP2), AI and Data Analytics (WP3, 4, 5) and implementation of the services in WP6.

The document also provides a **comprehensive overview of the LOGISTAR deployment environment** that is part of work package 2 and work package 6 – here referrals to other deliverables are provided that should be taken into account to read the full picture of the LOGISTAR infrastructure, deployment environment and storage infrastructure.

For the **data model definition** existing standards in the field of transport and logistics have been identified, evaluated and will be taken into account, like Datex 2 (road) or TAF/TSI (rail) as well as the Logistics Interoperability Model (LIM, https://www.gs1.org/lim).

This report also explains the planned **enrichment and entity linking**, lists identified relevant controlled vocabularies (e.g. taxonomies) and knowledge models (e.g. ontologies) that are established as standards in the domain and can be reused and adapted where needed and will be integrated to the **LOGISTAR Knowledge Graph** of logistics.

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

1. Introduction

D2.2 Specifications of the data collections and enrichment pipelines is a deliverable of type REPORT. It is an important basis for other work packages in the project, mainly for Artificial Intelligence (AI) and Data Analytics (WP3, 4, 5) and implementation of the services in WP6 - but also for the WP2 Data Acquisition and Data Storage itself.

Thereby this deliverable is highly connected with other deliverables - meaning such deliverables need to be read in combination with this deliverable on hand - to provide the full picture for data identification, harvesting, storage and processing.

List of interdependent deliverables

- D1.1 Market research: interviews, user needs & functional requirements analysis
- D2.1 Report on identified and specified data sets in the form of a data management plan
- D2.3 Data storage infrastructure,

and in a later stage of the project

- D2.4 and D2.5 Data Acquisition and extraction layer of the LOGISTAR Platform V1 & V2
- D2.6 and D2.7 Metadata Layer of the LOGISTAR Platform V1 and V2
- D6.1 and D6.2 LOGISTAR architecture and detailed design V1 and V2
- AND the deliverables of work packages 3, 4 and 5 that describe how the LOGISTAR data is being used to develop several services that will be integrated into the LOGISTAR platform in WP6

The deliverable on hand describes and summarises LOGISTAR how metadata and data in LOGISTAR is being pre-processed, enriched and provided to other work packages to make use of it for LOGISTAR services, that will be integrated into the LOGISTAR platform.

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

2. Data and Metadata Handling in LOGISTAR

Receiving, harvesting, enriching and storing data for downstream data processing by LOGISTAR services in the platform is crucial for the success of the whole LOGISTAR project.

There are several types of data to be handled in the project as follows

- **Data from the use case partners**: mostly sensitive and also sometimes personal related data, coming from transport management systems.
  - For service development purposes, such data is provided in the form of data samples as CSV or XLS files, e.g. dumped from the respective databases with a description of database models, etc. (see Annex I - Guidelines for LOGISTAR UC Data Samples of this document for details on related guidelines). For this a secure data sharing space is used (based on an OwnCloud (https://owncloud.org/) installation in the infrastructure of LOGISTAR partner SWC). Later in the project the LOGISTAR infrastructure will be used for this - in detail this means a secure HDFS (Hadoop) environment
  - For production LOGISTAR system purposes, such data will be used directly coming from the TMS (Transport Management Systems) of the partners in 'real time' to be processed and analysed by the LOGISTAR services / platform to provide the results of the services. For this the LOGISTAR infrastructure will be used - see D2.3 for the different types of stores used in LOGISTAR (data bus, event store, HDFS, metadata store) and the architecture specification of WP6.
- **Data from Open Data sources**: open data per definition, thereby NOT personal related information, data with an open license for free use / re-use. Such data will be harvested from public open data portals like the European Data Portal (https://www.europeandataportal.eu/) as well as the national / local open data portals of the target regions: UK (and potentially Ireland), Italy, and Spain.
  - Such data will be harvested (harvesting mechanisms to be described in detail by D2.4 and D2.5 Data Acquisition and extraction layer of the LOGISTAR Platform V1 & V2.) and stored in the LOGISTAR (storage) infrastructure mainly as data files in HDFS, if available via API through the data bus in the event store.
- **Metadata for above-mentioned data**: describing the data sets / APIs in detail in regards of their attributes, timely and geographical coverage, license, etc. Here DCAT-AP profile (https://joinup.ec.europa.eu/release/dcat-ap/11) will be used as a basis and adapted if required (in clarification at the time of creation of this deliverable). DCAT-AP is the de facto standard to describe data in the European Open Data ecosystem and can be applied to LOGISTAR very well.
  - The metadata will be stored in a data catalogue that is based on a graph database / triple store and provides features to search and browse available datasets and make use of data samples in place. A data catalogue evaluation is taking place in the course of the project but is ongoing at the time of creation of this deliverable. Annex II - Criteria for Data Catalogue Evaluation provides information about the criteria of this evaluation.
- **Data in the form of controlled vocabularies, taxonomies, ontologies and / or data models**: mostly open licensed data that can be used and reused as well as adapted. Sometimes commercial and thereby proprietary.

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

- Such data will be imported into the PoolParty Semantic Suite (component of LOGISTAR partner SWC, https://www.poolparty.biz) to be stored, maintained, linked and enriched.

2.1. Identified requirements for data provision by LOGISTAR service providers

The requirements of the technical partners that develop the services on top of LOGISTAR data - mainly WP 3,4,5 - has been collected in order to be able to provide the clear vision of the technical work related to datasets. The main functional logistics platform requirements are detailed into sub-requirements and further analysed covering all operations of the envisioned system:

- **Optimisation constraints & decision criteria:** Routing and scheduling data for single companies and co-loading for multiple days with the decisions and predicted orders that should lead to minimized cost.
- **Integration, automation & access to information:** Automatic push of the data back to companies internal ERP systems
- **Data feeding:** Real-time data communication must be supported for GPS data and open data. Orders and route timings as well as collaboration history must be fed historically and in real-time into the LOGISTAR system.
- **Event detection:** In real-time to calculate different predictions based on current events.
- **Predictions:** predictions of travel and turnaround times, detection of delays, predictions of orders / loads, risk / failure predictions
- **Preferences learning:** learning how companies are dealing with risk and the kind of proactive actions and risk prevention strategies they are using.
- **Modal, unimodal and multimodal route planning and optimization**, with collect and delivery,
- **Visualisation and validation of initial planning** and re-scheduling of initial planning and execution monitoring.

All data-related requirements are provided in more details in Table 1.

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

Main Requirement	Sub-requirement	Comment
Optimisation constraints & decision criteria	Routing & scheduling should ensure loads are single company where possible & backhaul preferred over co-loading/consolidation.	
	Overarching principle is that the total of all loads should be cost minimised.	
	Multiple day planning for all current and predicted orders	Certain orders may be excluded for consideration in collaborative routes.
	LTL orders should be combined into FTL for offer to hauliers (depends on contract pricing).	
Integration, automation & access to information	Intelligent adjustments to master data rules (e.g. turnaround time for delivering at customer premises).	Automatic push of LOGISTAR data to companies' systems must be implemented by companies themselves.

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

	Mechanism for cross-charging of collaborative partners.	
	Knowledge of stand trailer locations to help deal with imbalances between locations.	
	Live visibility of hauliers performance at the time a journey is assigned to haulier.	
	Allocation of loads to the “best, most suitable, haulier”.	
Data feeding **(priorities will depend on the priorities of the services for which the data is required)**	**Current orders** are pushed to the LOGISTAR system	Required for orders / loads predictions.
	Historic orders are pushed to the LOGISTAR system.	
	Historic route timings are pushed to the LOGISTAR system.	Required for travel time predictions.

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

	Historic turnaround timings are pushed to the LOGISTAR system.	Required for turnaround time predictions.
	Vehicle position / GPS tracks are pushed to the LOGISTAR system.	Required for fine-grained travel time predictions at the road-segment level.
	Rail time tables are pushed to the LOGISTAR system.	Required for travel time predictions for railways.
	Rail historical timings are pushed to the LOGISTAR system.	
	Historic collaboration parameters are pushed to the LOGISTAR system.	Required to learn companies' preferences.
	Open data (weather, traffic, events, etc.) are pushed to the LOGISTAR system.	**Required for timings predictions.**
Event detection	Live events detected are pushed to affected companies.	Might be pushed via the dashboard.

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

Predictions of travel times	The origin-destination travel time matrix is pushed to the LOGISTAR system.	Based on time of day, day of the week, week of the year, and potentially weather forecast, and detected events.
	Single-route travel time prediction via an API.	Useful for on-the-fly replanning.
	Shortest travel time of the day / week for a given route via an API.	Useful for planning.
Predictions of turnaround times	Single turnaround time for a client / DC at a single time via an API.	Useful for planning when the delivery is fixed.
	Shortest turnaround time of the day / week for a given client via an API.	Useful for planning.
	Turnaround time matrix is pushed to the LOGISTAR system.	Useful for optimisation of the plan.
Prediction & detection of delays	Delays are detected and notified to the company based on predictions of propagation.	Might be pushed via the dashboard.

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

Predictions of orders / loads	Predictions of orders / loads for the 5 next days are pushed to companies.	Might be pushed via the dashboard.
	These predicted flows will be sent through the TPM to the transport planning system for acceptance/modification & returned to TPM.	
Risk / failure predictions	Failure predictions & risk assessment are pushed to companies.	Might be pushed via the dashboard.
	Live failure predictions based on delay and history are pushed to companies.	
Preferences learning	Learned preferences are pushed to the LOGISTAR system.	**Useful for automated negotiations.**
Data feeding	**Current orders**, **vehicle availability** and **driver information** are pushed to the LOGISTAR system	
	Facilities, **vehicle information**, **train schedules**, **Products & vehicles**	

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

	compatibility and **driver regulation** are pushed to the LOGISTAR system	
Modal and multimodal route planning and optimization Routes & schedules will be produced for next 5 days using the matrices supplied by the prediction module, taking into account any intermodal options available, and regulations associated with drivers hours.	Unimodal Routes (Trucks)	*Pending on validation on 3 or 5 days for planning
	Unimodal Routes with multiple orders/collections should be checked for product and vehicle compatibility	
	Unimodal Routes will be allocated to schedules which minimise the number of vehicles being used	Where possible carrier loads will be scheduled in the same way
	Multimodal Routes (Trucks + Train)	Planning is considered for an horizon of 3 or 5 days *Pending on validation on 3 or 5 days for planning
	Ability to synchronise with rail if goods are compatible, timetables fit and cost is OK.	
Unimodal route planning with collect and delivery	Collect (C) -> Deliver (D) (DTDVRPTW1) traditional collect FTL from DC and deliver, return empty	

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

Optimise vehicle fill & backloading taking into account timing & offset distances to collect/deliver backhaul. Co-loading should be considered taking into account cost and timing. Ability to have multiple collect and deliver in the same route. Vehicle routes should not have to start and end at the same location. Sequencing of multi drop routes should consider backhaul opportunities. FTL loads should also take into account return load opportunities and have the option of steps/legs as in “pony express”, i.e. a system of relays via DC locations (drop and swap), ensuring timescales are met. Flexible routing to either return vehicle at end of route to origin or another location.	C->D->D->D traditional collect from DC and multi drop route, return empty	
	C->D->C->D traditional collect FTL from DC and deliver, followed by collect & deliver backhaul	
	C->D->C->D->C->D->C->D multiple collect & deliver on a single route, ideally with backhaul	
	C->C->D co-load from 2 or more DC's for a single FTL delivery	
	C->C->D->D->D co-load from 2 or more DC's for a multi drop route	
	C->C->D->C->D co-load from 2 or more DC's for a single FTL delivery, with collect & deliver backhaul	

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

	C->C->D->D->D->C->D co-load from 2 or more DC's for a multi drop route, with collect & deliver backhaul	
Visualization and validation of initial planning These routes and schedules will be visually presented to company load planners	The system will show route planning and schedules to planners	
	Where there are collaborations these will have been subject to automated negotiation algorithms and the outcomes will be highlighted for each company to accept or reject	
Re-scheduling of initial planning Real time route monitoring and changes made if efficiencies develop, or transport (truck/rail) is going to be late to prepare for knock on impact	This may be due to delayed departure from warehouse, congestion on route, accident, etc. which will be flagged by IoT GPS	

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

Execution monitoring	**Information to be displayed** on company dashboard: 1. Planned versus actual travel/delivery time, expected arrival time, etc.; 2. Status of transport (location plus loading, travelling, break, unloading, and unexpected situations such as stationary for longer than, say, five minutes or off planned route); 3. List of target and current KPI's; 4. Event responses & incident prediction probability; 5. Relevant open source information on likely weather/congestion/incidents 6. Alerts & recommendations This will appear on each companies systems and allow the transport planners to react and modify journeys as appropriate	

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

Company informed of any suggested route changes with option to intervene.	Any delays displayed via dashboard to allow them to contact customer if required.	
Real time route timings will be passed to WP3 to update the historic route time database		
Visibility of haulier trucks in real time (only for single user or co-loaded users - will be issues with regulations if shared user)		

Table 1 LOGISTAR data functional requirements

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

2.2. Environment of Data Storage and Processing

The details of the data storage are described in D2.3: Data storage infrastructure, but the overall concept and the list of components are provided as follows to provide a quick overview and insight into the data storage and pre-processing landscape of LOGISTAR.

2.3. Data Storage Layer in LOGISTAR

The architecture of the LOGISTAR system will be developed in iterations, meaning that a first version of the platform will probably be built using decentralized layers of components, i.e. some components will be based on the same cloud and some will be deployed separately, this applies to all ongoing work packages from (WP2, 3, 4, 5).

The full and detailed description of the LOGISTAR (Data) Storage Layer can be found in Deliverable 2.3 Data Storage Infrastructure. As follows, a brief overview of all components to provide context in reading the deliverable document on hand.

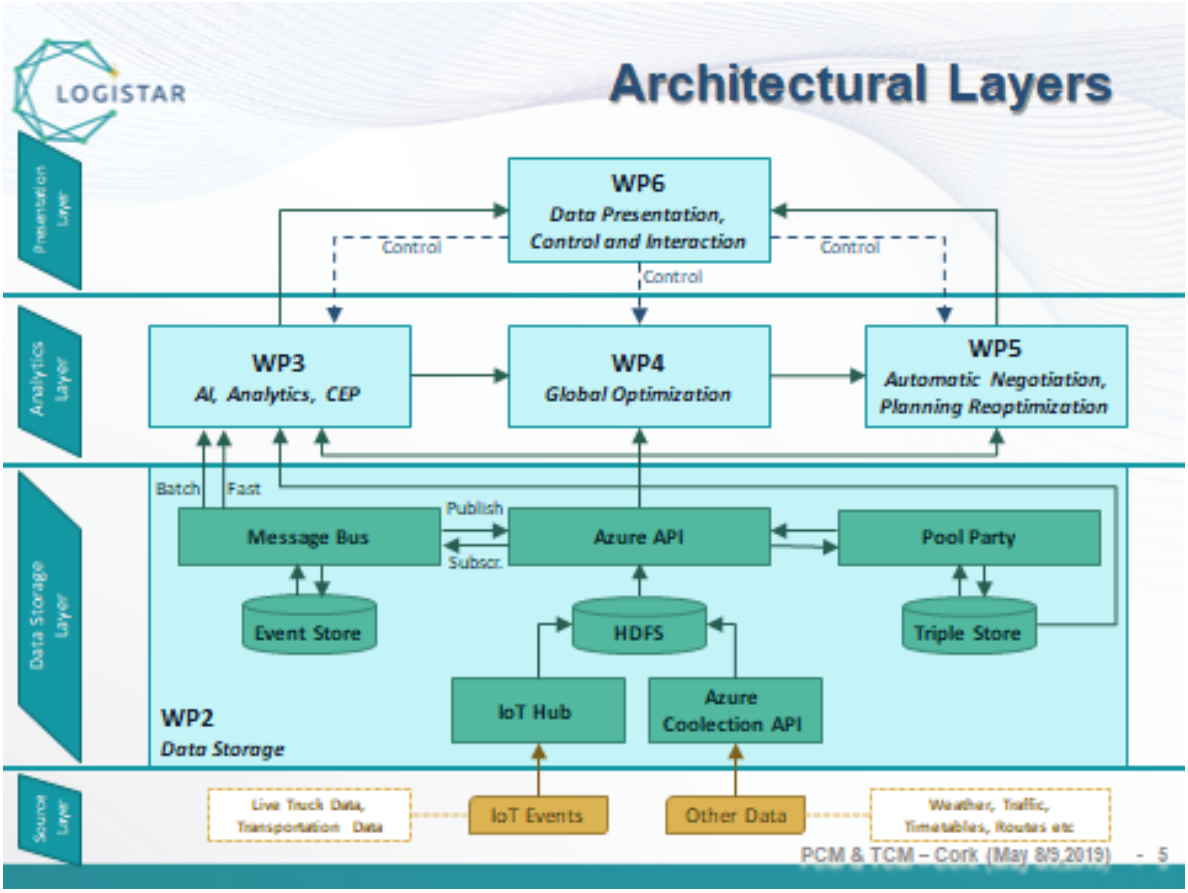

Figure 1: Architectural layers of the LOGISTAR system.

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

Components by Software AG

Message Bus (Universal Messaging)

Universal Messaging (UM) will serve as the message bus. UM is a Message Oriented Middleware product that guarantees message delivery across public, private and wireless infrastructures. Universal Messaging has been built from the ground up to overcome the challenges of delivering data across different networks. It provides its guaranteed messaging functionality without the use of a web server or modifications to firewall policy.

Event Store

Universal Messaging will use a data storage facility, such as Software AG's Terracotta, MySQL (or even Cassandra) to make messages persistent, as needed.

Components by DNET

Protocol adapters

Protocol adapters are a software layer that will be deployed directly on the devices to enable Communication with the platform for devices which are not capable to communicate using native API that are going to be developed. The development technology will depend directly of device, the programming language supported by the device will be used (e.g. C, C#, etc). This layer will provide automatic device provisioning for communication with the IoT Hub.

IoT Hub

IoT Hub is a service with central message hub for bi-directional communication between the cloud and devices. It will serve to collect more or less all data in the LOGISTAR platform (data collected to be used sloley of processed by different functional blocks). This service supports multiple messaging patterns such as device-to-cloud telemetry, file upload (can support different types of data available in the project: pdf, weather, csv, etc), and request-reply methods to control devices from the cloud using HTTP/1.1. IoT Hub service also monitors events such as device creation, device failures, and device connections.

Azure Cloud Service

Azure Cloud Service is platform as a service (PaaS) with enhanced control over virtual machines. Cloud services will be developed as custom application to allow different processes required to enable fluent work of components first in WP2 then in the complete project. Custom software could be installed on VMs that use Azure Cloud Services, and accessed remotely. All VMs in a single application run in the same cloud service. Users access the application through a single public IP address, with requests automatically load balanced across the application's VMs. The platform scales and deploys the VMs in an Azure Cloud Services application in a way that avoids a single point of hardware failure. Cloud service applications could be deployed in .NET, Node.js, php and Python.

Integration Layer

Integration Layer is used to connect Cloud Service with the rest of the platform.

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

Components by SWC

PoolParty Semantic Suite

PoolParty Semantic Suite is a semantic technology suit that offers sharply focused solutions to transform knowledge organization and content business. It offers a wide variety of options to benefit from semantic technologies. It provides components for data modeling (taxonomies, thesauri, ontologies, kgraphs), text analysis and –extraction and for RDF (Resource Description Framework) data integration by making use of W3C semantic web standards.

PoolParty Graph Search

PoolParty GraphSearch is a faceted search for RDF data. It's an entity centric search. The usual workflow of a PoolParty GraphSearch project starts with the gathering of structured and unstructured data. You can use data from various sources by using UnifiedViews and/or by transforming documents into RDF by means of the PoolParty Extractor. The processed RDF data is stored in a search index like Apache Solr or Elastic Search or an Enterprise graph database. The PoolParty GraphSearch Server offers a web service API that follows the RESTful principle and produces results in JSON for 'traditional' document search applications with additional beneficial features like synonym search and hierarchical drill down based on the knowledge graph that is managed with PoolParty Thesaurus Server.

UnifiedViews

UnifiedViews is an ETL tool for RDF data. This tool provides interface to define, execute, monitor and schedule processing of RDF data tasks. The tool already provides certain predefined plugins so no heavy programming is required to use this. And pipeline creation is also very simple. The user can extend custom plugins to define their own plugins. UnifiedViews DPUs (Data Processing Unit) or plugins are divided in three categories: Extractors, Transformers and Loaders. Extractors obtain data from external sources like csv, excel or RDF files. Transformers transform data to other formats, for example CSV files to RDF data or relational tables to RDF data. Loaders finally load data into external system or repositories.

Metadata Store: Triple Store / Graph DB

The storage system, database for metadata stored in Resource Description Framework (RDF) in the Logistar system – in the form of a graph data base / Triple store that allows to store RDF data properly and enables performant querying.

2.4. Data Preprocessing along the (linked) Data Life Cycle

Data and metadata management approaches follow these days the data life cycle to provide useful data that is harvested from several sources, cleaned, enriched and preprocessed to finally be used and being useful for several applications / analysis. See as follows the data life cycle of linked data. LOGISTAR follows this approach to ensure efficient data acquisition, preprocessing, linking and cleaning and finally data provision for the services of the LOGISTAR platform, means for the service development in WP 3, 4, and 5.

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

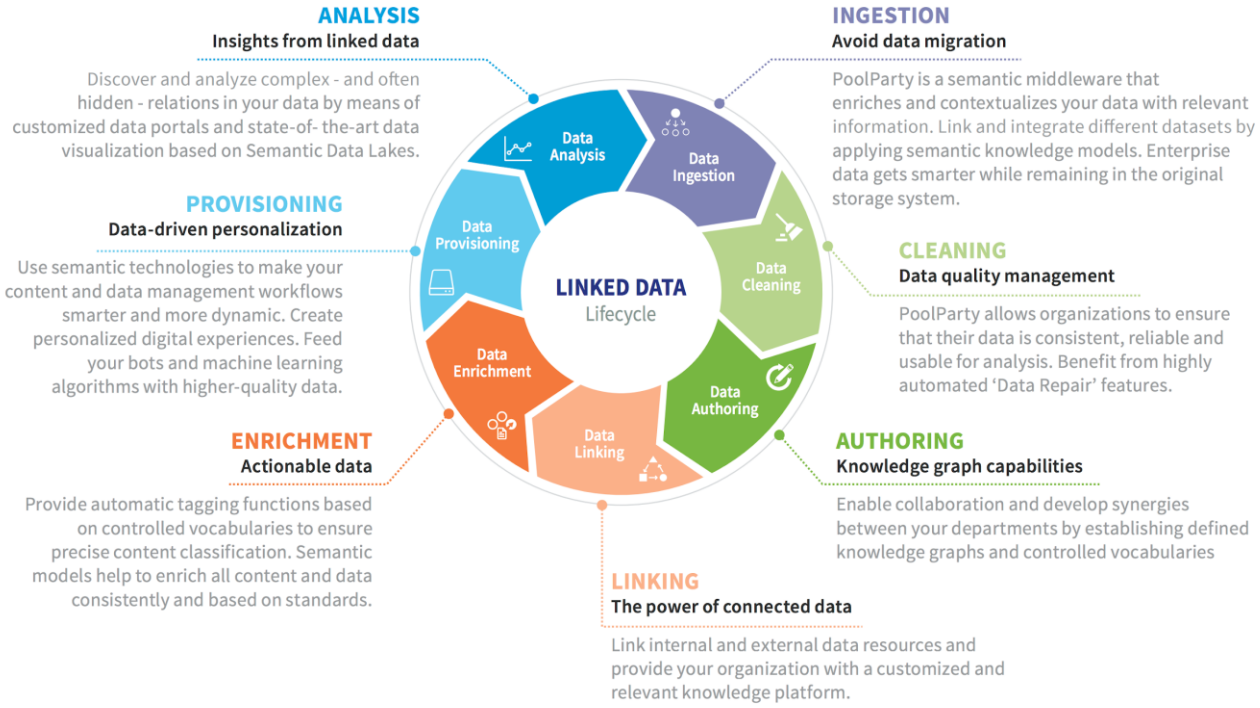

Figure 2: The (Linked) Data Life Cycle.

Data preprocessing

Data preprocessing is a process of cleaning and aggregating the data after receiving it from data sources. This concept is applied before machine learning and it is executed once during the collection. So far in the projects, data samples are collected and preliminary analysis done. The goal is to support machine learning algorithms by the requirement of the learning/classification model that will be applied over the collected dataset. In the phase when machine learning is in place, data attributes will be mapped to different models in order to compute more complex attributes or to extract attributes from complex context. The data will be first collected using **IoT Hub** and **Azure Cloud Service** where the further processing using different libraries will take place.

At the moment, the analysis of the current version of libraries (the current state-of-the-art) for preprocessing are mostly Python-based and freely available online with libraries such as Pandas [https://pandas.pydata.org/] and Numpy [https://www.numpy.org/].

The data preprocessing tasks require the following steps:

- *Data cleaning,* which is the first step with a focus to handle missing, noisy data or detection and removal of duplicates and outliers;
- *Data mapping, linking and enrichment,* consisting of mapping of data attributes (e.g. controlled vocabularies and terminologies used), interlinking of data by making use of common attributes and/or the LOGISTAR Knowledge graph and finally enrichment of data by the Knowledge Graph to enable more efficient data analysis;
- *Data integration,* process of gathering various data sources combined into a single data pool (repository);
- *Data transformation,* conversion of raw data info specified format based on the defined model;

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

- *Normalization,* transformation or scaling of the data into specified ranges or values;
- *Aggregation,* grouping or combining several data categories into one;
- *Reduction*, removing redundant and unnecessary data;
- *Data analysis,* to make use of acquired, cleaned and preprocessed data to develop the LOGISTAR services that will be integrated into the LOGISTAR platform.

The libraries are going to be deployed as a part of Azure Cloud Service and will perform certain predefined above tasks. Currently, maturity of the algorithms is not in the phase that allows composition of these preprocessing services and it will be installed later in the next phase of the project. Having in mind that there will be different algorithms from different partners, as Azure Cloud services consists of number of worker and web roles, each of which performs a conceptually separate task, we will separately replicate across virtual machines different preprocessing algorithms where necessary. These "preprocessing workers" are going to be deployed as separate (web) applications and can be built and further developed from this point having in mind that any web framework that supports WSGI can be used to develop an application (as supported by the Web project template).

The overall orchestration of data processing in LOGISTAR will be managed by making use of the message bus, a software component brought by SAG into the project. For more information, please see Deliverable D2.3 Storage Infrastructure that explains the capability of the message bus in detail.

2.5. LOGISTAR Infrastructure: security, deployment, storage of data and algorithms

To get a full picture of the overall LOGISTAR infrastructure, please read this document (and especially this section) together and interlinked with the two deliverables D2.3 Data Storage Infrastructure and D7.4 System Deployment and Testing Specification.

2.5.1. Security

Securing data sharing in the course of data harvesting and processing is a crucial issue in the LOGISTAR project, as (i) sensitive data is shared by use case partners and 3rd parties that should ONLY be used for service development and for data processing, but should not be shared between project partners (in raw data form), neither should data be accessible as raw data by any other parties than the technical partners of LOGISTAR that have a need to access such data to develop special services.

To ensure this secure data sharing, a **data governance** has been developed and will be further expanded in the course of the project that includes at the moment the following parts / parameters:

- **Non-Disclosure Agreement** (NDA), between (i) the use case partners and (ii) the overall LOGISTAR consortium represented by the project lead Deusto.
- **A secure space for data sharing** for data samples, in the form of an OwnCloud (https://owncloud.org/) service installed in the infrastructure of LOGISTAR partner SWC. A secure HTTPS encrypted data space for data sharing where access rights have been specified and set according to the above described guidelines of data access and respective restrictions.
- **Access rights management policies**, ensuring that, in the secure data space, use case partners can only see and access their own data but not any data from other use case partners and that

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

all technical partners who require it (listed in additional document) have access to such data. This includes continuous maintenance in regards to the the access rights to data in LOGISTAR.

- **Secure data spaces and storage,** following industry standards and making use of HTTPS encryption where required
- **Additional secure data sharing mechanisms,** like for instance iSHARE, a protocol for secure data sharing developed by the iSHARE project (https://www.ishareworks.org/), that is analysed by LOGISTAR partners at the moment, but also other similar available mechanisms.

Collection of data will be done with full consideration of data protection principles and will satisfy data protection requirements in accordance with EU and non-EU directives and national implementations as explained in the deliverable: D2.1 Report on identified and specified data sets in the form of a data management plan. The following components and technological measures will be applied to minimize associated privacy risks (in addition to the data governance rules listed above):

- **Data storage:** use of secure data storage using industry standards to protect data from any access from the outside, encrypted transfer (HTTPS) of data over the capturing channels, controlled and auditable access for different classes of data;
- **Anonymisation:** obscuring/removing user identities at the source of data collection/ generation to prevent direct user tracking. Data controllers will be implemented to ensure that anonymisation is done before the actual storing, or handling the collected data before any processing takes place. The controller will execute suppression (the removal of values from the original dataset by replacing the values) and generalisation banding techniques to produce coarser-grained descriptions of values than in the original dataset (difficult to identify any individual from this abstraction, but still providing rich information that can be analysed, e.g. location postcode may not be needed in the majority of data analysis, but the area or region is required). This process of obscuring personal data through indirect or delayed routing will prevent individual localisation as much as possible and limit user tracking through correlation of depersonalised data based on its location.

2.5.2. Infrastructure

When setting up a complete infrastructure to run perceived state-of-the-art and complex software systems and algorithms there is a broad range of requirements that need to be met. First and foremost, the system needs to be flexible in terms of the fact that we cannot know which tools and components will be used to code the actual algorithms. We can do educated guesses but must be ready for changes down the road. Different versions of code that create prediction models and the likes must be deployable more or less instantly. The sheer number of components that will make up the whole LOGISTAR system makes it clear that we would “nuke” any hosting operating system with respect to dependencies and required libraries in different versions.

Additionally such a modern environment, forming the basis of a research project, must be capable of exposing “High Availability” in the sense of being fail-safe, be ready for full parallel processing in case the actual algorithm is coded that way and finally be (relatively) easy to maintain. That means that algorithms, components and other pieces of code must be easy to deploy and adhere to the principle of “code once - run anywhere”.

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

To fulfill the above requirements, LOGISTAR's software infrastructure will be built around "Docker" [x-1]. This will give developers of algorithms the opportunity to create software with the most appropriate tools available to them and still have those algorithms available in a way that is easy to deploy without touching the underlying operating system and in a manner that guarantees runnability, not only on the nodes they were developed on but also on our demo cluster and in production when set up within companies.

It is understood that Docker alone won't solve all problems that can be foreseen as of writing. Usually, Docker is only part of an overall system that guarantees a fail-safe environment that is known as "Container Orchestration" [x-2].

Today, there are already a lot of container orchestration frameworks available, Docker Swarm [x-3], Kubernetes [x-4] or Mesosphere Marathon [x-5], that are deployable on premise or OpenShift [x-6], a cloud-based solution, just to name a few.

Since Semantic Web Company already gained a lot of experience with Mesosphere Marathon in a previous H2020 project, Big Data Europe [x-7], we will build upon it and expand our knowledge within the LOGISTAR project.

Our proposed and implemented solution will therefore be based on DC/OS (Datacenter Operating System) [x-8], which adds an additional layer of abstraction above the orchestration framework. With DC/OS, it is possible to handle multiple orchestration frameworks, like Mesosphere Marathon or Kubernetes. We will exploit this possibility to not only gain insights with respect to the readiness for production of the system, but also to be as flexible as possible.

In the following chapters we will give an overview of the components that will make up the infrastructure described above as well as an outline about the storage layer, set up in a secure manner that will serve as the backbone of many other components.

2.5.3. Hardware

For a modern environment as described in the introduction, a simple notebook is clearly not suitable. Since the LOGISTAR project did not include a dedicated hardware in the first place, consortium members will rent a couple of root servers from the German provider "Hetzner" [x-9]. At least within the own infrastructure of the partners of the project, who are responsible for software deliverables, it is not possible to setup nodes, also if this would be possible from a perspective of available resources. Within the security restrictions and guidelines put forth by Semantic Web Company's ISO 27001:2013 certification [x-10], it is easier to setup external nodes for this project to not interfere with internal network segments, certificates, firewall zones, open ports and the likes.

DC/OS has a minimum of four nodes to run. We have chosen to go with the EX42-NVMe model [x-11] from Hetzner. This root server comes equipped with two Intel Core i7-6700 Quad-Core CPUs. This will enable algorithms to run on 24 cores in parallel in our initial setup. It is expected that algorithms, working on route optimization will also need this kind of environment to be able to run. It should be noted that we do not support GPU acceleration in our current setup due to hardware constraints. Usually, this can also be considered an optional feature in the implementations of algorithms which can make use of it only when it is available.

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

64MB RAM and 2x512GB of storage on each working node, will guarantee a suitable base system for the anticipated workload.

CentOS 7.6. [x-12] was chosen to be all nodes' operating system.

2.5.4. Container Orchestration

Container orchestration can be considered as the management and continuous delivering of (micro) services packaged as containers. It is well-known that containers, as opposed to virtual machines, can take different forms. This document refers to Docker containers unless stated otherwise.

As described in the introduction, our preferred solution for container orchestration for the LOGISTAR project will be based on Apache Mesosphere as the container orchestration framework and on DC/OS as a layer above for utmost flexibility and ease of use. The components making up this base system as well as a secure version of HDFS as an example of things to consider when deploying services will be described below.

2.5.5. Deployment

Apache Mesos

Apache Mesos [x-13] forms the basis of the system forming the basis for the LOGISTAR project's software infrastructure. Apache Mesos is a cluster manager acting very close to the underlying hardware. Mesos was initiated as a research project at the University of California at Berkeley. A first version was available in 2009, still under the name Nexus [x-14]. Unlike Google's cloud computing infrastructure Borg [x-15], it was an open-source project from its initiation. In 2016, Mesos became a top-level Apache project under the name of Apache Mesos. It is not often visible to end users but powers applications for a long list of companies [x-16].

On its homepage, Apache Mesos is described as a "Distributed Systems Kernel". As such, it provides applications with the capability to manage hardware resources like CPU, RAM or HDD on up to tens of thousands of nodes.

Two basic concepts should be mentioned in this introduction for better understanding how Mesos enables computing on a whole host of nodes. First, "Resource Offers" and "Frameworks". Apache Mesos relies on resource offers for different tasks. Resource offers are mostly hardware resources, but also other resources like network ports that an arbitrary node within the managed cluster can offer. A given application's requirements are matched against such resource offers by arbitrary nodes. Apache Mesos then decides on the placement of the application on a specific node within the cluster. It is understood that fine-grained options are available but out of the scope of this document, some will be mentioned during the following chapters.

Usually, one does not interact directly with Apache Mesos but through the gateway of "Frameworks", which are dedicated applications running on Mesos. A Framework would then run the application handed over to it on a to-be-decided node in the cluster. Mesosphere Marathon is such a framework with which it is possible to orchestrate docker containers.

See below (Installation) for a limited description of the installation process.

Mesosphere Marathon

Mesosphere Marathon is the actual orchestration framework. It's specifically designed to run as an Apache Mesos framework for long-running tasks. Via its RESTful API, it is possible to launch Docker containers defined by a runtime description in JSON. This configuration file will describe the entire application's runtime environment, specifying its docker image, network configuration, mounted volumes, placement within the cluster, CPU requirements and so forth. An excerpt of such a configuration can be seen below.

```
{
 "id": "/namenode",
 "acceptedResourceRoles":[
  "slave_public"
 ],
 "cmd": "cd $MESOS_SANDBOX && sh init-namenode.sh",
  "instances": 1,
  "cpus": 0.5,
  "mem": 2048,
  "disk": 0,
  "gpus": 0,
  "fetch": [
     {
       "uri":"http://example.org/init-namenode.sh"
     }
  ],
  "container": {
     "type" : "DOCKER",
     "docker": {
     "image": "openjdk:8-jdk-alpine",
     "forcePullImage": false
     },
...
```

All marathon configurations for LOGISTAR's components and applications will be hosted next to their respective required resources on the project's GitLab repository.

Docker

Docker is currently the most widespread container tooling suite on the market. Unlike virtual machines, containers run as isolated processes on the host machine. Docker will enable us to easily version, build, maintain and run the anticipated complex system that form the LOGISTAR code base. As described below in the Conventions section, we will use simple base images in which the application to be run will be configured by init scripts.

DC/OS

DC/OS, the Datacenter Operating System, exposes features of multiple systems, like a container runtime, a cluster manager and an operating system. It does so by the virtue of the underlying tools, that are mentioned above, when looked at from different perspectives. With DC/OS, we will gain the most flexibility, since multiple orchestration frameworks can be used, for example we could also deploy Kubernetes as an own DC/OS service in the future. In addition to that, DC/OS also serves

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

graphical user interfaces to manage network configurations and a package manager for already available services. As a layer above Mesosphere Marathon, it exposes a host of services for the maintenance of the IT ecosystem, like log aggregation and health checks.

Installation

For the LOGISTAR project we will install the latest stable version 1.13 of DC/OS according to its documentation.

Bootstrap Server

A bootstrap server in the sense of DC/OS is a simple node from which the initial installation and arbitrary upgrades take place. There are hardly any special requirements for this node other than 16GB RAM and free disk space of 60GB. In addition to this limited role, we will assign two other tasks or roles to this node within the LOGISTAR project. The first role is to serve resources required to be available for download from within running Docker containers and second to serve as a persistent volume other than the container's host.

An example for the first role would be that a Docker image to run HDFS wants to download a specific Apache Hadoop distribution (like 2.9.0, see below). In this case, we do not want to rely on external servers and host the binary distribution on our bootstrap server and make it available via an arbitrary web server like nginx [x-17].

As an example for the second role, it should be mentioned that some applications will require persistent volumes that are not tied to a specific host in the cluster. Such directories must be available to a running Docker container regardless of its placement within the cluster.

Master Server

In the anticipated setup and due to limited hardware resources for the LOGISTAR project, we will resort to installing one Master node only. Masters have a minimum requirement of 4 CPU cores, 32GB RAM and a hard disk of at least 120GB. Our rented servers all meet those requirements. This node will serve as the Mesos master node. In addition to that, it also serves as the Mesosphere Marathon master node and with that as the node where Apache Zookeeper is installed. An Apache Zookeeper quorum is required as the coordination tool for Mesosphere Marathon. Last but not least, several internal DC/OS services will be available from this node, among them the internal DNS just to name one.

Public Nodes

Public nodes fall under the category of agent nodes. Those are Apache Mesos worker nodes where actual computing will be done based on the offered resources, see above. Public nodes are special in that running services are exposed with fully qualified domain names based on the conventions offered by DC/OS, see below. This means that the placement of Docker containers will be well defined and that applications or services exposing endpoints that should be available via HTTP will be running on public nodes.

Private Nodes

The second type of agent nodes do the actual computing lift. Other than public nodes, these do not expose any service for external access. It should be noted that clustered applications can certainly

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

be spread across a mixture of public and private nodes. In that case, intercontainer communication needs to be defined by other means, like a virtual network.

Administration

Administration of the cluster will be done in two ways. First of all via DC/OS graphical user interface on our master node on https://node-one.LOGISTAR-project.eu and secondly via DC/OS CLI, the "Command Line Interface". This tool will be installed on the master node and is available for administrator of the cluster. Note that both options are only available to a selected list of technicians after an introduction on the how to work with the cloud computing infrastructure. Both of those tools are rather user interfaces, the administration of services deployed to DC/OS is done via their respective marathon application descriptions, containing all details about the resources needed. Cluster administration that requires intimate knowledge about the linux operating system and the datacenter operating system will have to done by a specialized administrator and are out of scope of this document.

2.5.6. Storage Infrastructure

Secure HDFS

As a common multipurpose storage infrastructure we decided to go with HDFS [x-18], the Hadoop Distributed File System, a module of the Apache Hadoop [x-19] family of tools related to big data processing.

HDFS is a distributed virtual file system that not only will serve as a common storage layer to share data in a secure manner, but also as the backend of some of the components making up the bigger architecture.

The fact that we are dealing with highly sensitive company data from the very beginning, we skip the description of HDFS setup in standard way. Instead, we will be describing the Secure HDFS from the start.

Secure HDFS not only includes an authorization and authentication system but also guarantees secure communication between the single nodes of the cluster on the one hand side and secure communication with the outside world by means of TLS + SSL and HTTPS.

Kerberos

Kerberos [x-20] is a network authentication protocol based on issuing so-called “tickets” [x-21] that are re-validated and used to prove an identity to a client, like arbitrary services or the likes. Kerberos originated from the Massachusetts Institute of Technology where it was primarily used for the University / Institute itself. It started spreading beyond the boundaries of the university from version 4. The current version, which will also be used within the LOGISTAR environment is 5, the exact distribution version for Alpine Linux, forming the base of most of our Docker containers, will be 1.15.4.r0 corresponding to the Kerberos version 5.

In essence, Kerberos authentication works on the basis of authenticating against the Kerberos service, which will return a “ticket” from the Key Distribution Center (KDC). The “ticket” is encrypted using the Ticket Granting Service along with a session key. A service making use of Kerberos sends upon invocation above Ticket Granting Ticket (TGT), the ticket, along with the Service Principal

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

Name (SPN) back to the Ticket Granting Service for authorization. If authorization is verified, a session key is sent back to the calling service and the process can start.

In general, it is possible to compare Kerberos to the well-defined authentication workflow used in web applications, where an authenticated user is mapped to a service user to connect to the database, however Kerberos is not a custom implementation of this behavior but well-defined by RFC 4120 [x-22]

See below for details on what service principals have been created for the secure version of HDFS, how those are deployed and used.

Kerberos Installation

For the LOGISTAR project, we have created a dedicated service running inside a Docker container to handle all aspects related to Kerberos. As with all our services deployed to Mesosphere Marathon via DC/OS, we are using Alpine Linux [x-23] whenever possible. Necessary libraries and binaries are all available for this Linux distribution. We are using Alpine's default package manager "apk" [x-24] to install krb5-server [x-25] upon start of the Docker container. Said package contains the necessary binaries, krb5kdc [x-26], kadmind [x-27] and kdb5_util [x-28] amongst others. With those libraries, we can set up our database for a defined "Realm", corresponding to an administrative section. We will gather our principals to run HDFS within the realm of "LOGISTAR-PROJECT.EU".

Kerberos Administration

To set up Kerberos to use HDFS for our realm, there are two configuration files necessary, krb5.conf [x-29] and kdc.conf [x-30], both containing specific settings for this project's environment. Within krb5.conf, the realm as well as some basic settings, like the lifetime of tickets, will be configured. kdc.conf contains pointers to the directories to the files that Kerberos will create for this realm.

Note that those configs and the resulting kerberos files are a good example of the necessity of one node serving resources to all nodes within the cluster as described within the "Bootstrap Server" section above.

The first step, when having krb5.conf and kdc.conf with the realm defined and pointing to the desired paths, in the valid places, is to create the database by invoking kdb5_util with the desired realm, e.g.

```
$ kdb5_util -r LOGISTAR_PROJECT.EU create -s
```

After creating the initial database, we can use Kerberos' binary "kadmin.local" to create the principals we need. For our initial setup, we will create the following principals required to run HDFS's namenode and datanodes: HDFS, host and HTTP. Note that those principals will be domain-specific. We will make those principals available in the form of "key tabs" to all nodes within the cluster. The specific HDFS Docker container will upon startup retrieve the correct key tab, store it inside an HDFS configuration directory and make it available to the HDFS runtime for further use. An example to create the HDFS keytab to use on node-one.LOGISTAR-project.eu can be seen below.

```
$ kadmin.local
kadmin.local: addprinc -randkey hdfs/node-
one.LOGISTAR.eu@LOGISTAR
kadmin.local: xst -norandkey -k hdfs.node-
one.LOGISTAR.eu.keytab hdfs/node-one.LOGISTAR.eu
```

This results in the keytab file hdfs.node-one.LOGISTAR.eu.keytab which can be downloaded in an unambiguous way from the resources repository upon Docker container startup.

It is recommended to create administration principals to be able to access the Kerberos server running within a Docker container also from within other Docker containers, especially in an initial phase if there are still things to debug. A first test, if the Kerberos system is set up correctly can be done by issuing “kinit” with an existing principal as an argument to receive a ticket, klist will show the validity of the received ticket, the “kdestroy” command kills the running session and ticket.

HDFS + Kerberos

As mentioned in the introduction, we will set up HDFS in secure mode due to confidentiality of the datasets provided by our use case partners. Secure mode with respect to HDFS means that every service, e.g. HDFS, yarn or mapred, as well as every user must be authenticated using Kerberos. Besides having Kerberos running and suitable Kerberos principals for all services as described above, we will also need to create x509 certificates for the communication between datanodes and namenodes, see below, as well as for accessing HDFS’s user interfaces over HTTPS. This is commonly referred to Service Level Authorization, which will guarantee that arbitrary application trying to interact with the filesystem will have appropriate permissions to do so. Encrypted transmission of bits and bytes between services as mentioned above is generally known within the Hadoop ecosystem as Data Confidentiality.

There are some consequences of running HDFS in secure mode, especially with a cloud environment like DC/OS. This mostly refers to network-related issues like the required privileged ports and network configurations. Since secure communication between services and nodes is achieved by using TLS + SSL, we need to run the Docker containers with “host” network configuration. That means that the Docker containers are resolvable by their respective hosts’ hostname. We need to do this to be able to create valid x509 certificates. In a first step, we will work with self-signed certificates, which is proven to work from the HDFS perspective. As soon as we will expose web services, hosted on our infrastructure as well, we will create x509 certificates using the “Let’s encrypt” [x-31] service at no cost. Although self-signed certificates will do the trick from a technical point of view, we do not want users of LOGISTAR services to have to add security exceptions to their browsers.

Secure Namenodes, Secondary Namenodes and Datanodes

In an arbitrary HDFS cluster, there is a clear distinction between (secondary) namenodes and datanodes. Namenodes are acting as masters, responsible for storing metadata about stored data as well as instructions on how and where datanodes should store files. Datanodes slaves, so to

speak, are the worker nodes within such a cluster. These nodes store the actual files, or parts of them (blocks) in a replicated setup as instructed by the namenode.

For the LOGISTAR project, we have created distinct DC/OS services with specific settings and configurations for each type of node. This is mainly necessary due to special requirements with respect to the placement within the cloud infrastructure. Datanodes will primarily be hosted on DC/OS private nodes as there is hardly any need to interact with them directly. Due to hardware restrictions, we will go with one single namenode, placed on the DC/OS's public node, so that interaction with the overall filesystem is possible via that namenode.

The most important settings to run a namenode in secure mode are the following: hadoop.security.authorization, hadoop.security.authentication, hadoop.rpc.protection, hadoop.http.authentication.kerberos.principal, hadoop.http.authentication.kerberos.keytab and hadoop.http.authentication.type. All those settings are taken from core-site.xml configuration file and specify the general service protection of the namenode. An excerpt of our actual core-site.xml can be seen below.

```
<property>
 <name>hadoop.security.authorization</name>
 <value>true</value>
</property>

<property>
 <name>hadoop.security.authentication</name>
 <value>kerberos</value>
</property>

<property>
 <name>
  hadoop.http.authentication.kerberos.principal
 </name>
 <value>LOGISTAR_KRB5_HTTP_PRINCIPAL</value>
</property>

<property>
 <name>
  hadoop.http.authentication.kerberos.keytab
 </name>
 <value>LOGISTAR_KRB5_HOST_KEYTAB_FILE</value>
</property>
```

The above XML snippets also highlights another feature of how we are dealing with general configuration issues. Our best practice is to download an init script along with all other required resources upon startup of the container. This init script prepares the whole installation according to the current environment. Again, in an orchestrated container environment, we only want to hard-code the placement of a Docker container for very specific reasons. In this example the init script for instance determines the current hostname by issuing the following command:

```
$ hostname
```

This will give the running container knowledge about its own placement within the infrastructure due to the fact that we are using the host's network. With that "knowledge", it is possible to download the correct keytab files from our resource repository as indicated in previous sections. The Linux "sed" binary is used to update configs for the filesytem, e.g.

```
$ LOGISTAR_KRB5_HOST_KEYTAB_FILE= \
    "hdfs.$(hostname).keytab"
$ sed -i 's,LOGISTAR_KRB5_HOST_KEYTAB_FILE,'
"$LOGISTAR_HDFS_CONFIG_DIR/$LOGISTAR_KRB5_HOST_KEYTAB_FILE"',g'
\ "$LOGISTAR_HDFS_CONFIG_DIR/core-site.xml"
```

Secure Communication (SSL)

It is understood that all forms of communication, between nodes within the cluster and between external clients in secure mode, require deployment of x509 certificates as well as the creation of keystores and trustores for HDFS to work properly. In our test environment, we have setup self-signed certificates and verified that those are suitable for inter-node communication. These self-signed certificates also work when logging into the running Docker container or its hosting node within the DC/OS ecosystem, however they are not suitable to communicate from the outside, e.g. to check the contents of HDFS directories via the graphical user interface using a browser.

A bash script to create all necessary self-signed certificates for all nodes of our infrastructure will be provided.

After the first smoke test, however, we will move to "real" x509 certificates for the reasons mentioned above. We will use the well-known "acme.sh" [x-32] script to create "Let's encrypt" certificates for all our nodes.

Privileged Resources and Ports

According to the Hadoop documentation [x-33], deploying HDFS in secure mode requires the usage of privileged ports that are ports below 1024. However, operating system users other than the root user do not have the permission to launch services on such ports. This creates an issue with ports, especially also in connection with the ports offered by Apache Mesos and hence DC/OS by default. To overcome the issue of ports, we have created a port system that work with HDFS in secure mode and still is not in conflict with the ports offered by DC/OS. Additionally we have adapted settings for HDFS to enable secure mode with non-privileged ports. The semantics of our port system, which our instance of HDFS in secure mode will run on are as follows. All ports, except for the default file system port follow this scheme. LOGISTAR's HDFS ports are five digits long and will all start with

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

10. The third digit represents the node type, 1 = namenode, 2 = secondary namenode, 3 = datanode. The fourth digit represents the protocol running over this port, 0 = not specified, 1 = HTTP, 2 = HTTPS, 3 = IPC. The fifth digit can still be enriched with semantics in case we need it in the future, currently it is 0 on all ports. This will make it also easier to communicate interaction with technical partners within the consortium who have to deal with the file system in a programmatic manner. An algorithm requiring interaction with the namenode over HTTPS can only use port 10110 by definition of the scheme without having to look up arbitrary ports somewhere.

DFS + Kerberos + Authorization

While we have spoken only about authentication up to now, it is also required to deal with the topic of authorization that is granting access to specific contents that is files and directories stored on HDFS. Hadoop's permission model is closely related to the POSIX [x-34] permission in terms of the concepts exposed. Similar to POSIX, it is possible to work with read and write flags for users, groups and everyone else. It is clear that we cannot "execute" any binaries on the virtual file system, hence setuid and setgid are obsolete. In addition to POSIX style permissions it is also possible to employ Access Control Lists (ACL). For the expected interaction with the filesystem this, however, will not be necessary.

In general, we will setup the permission system to reflect the different kinds of users within the LOGISTAR project. First, data providers like Nestlé, Pladis, Zailog, Codognotto and Ahlers and secondly, algorithm/service providers like the technical partners of the consortium. In essence, we need to guarantee that the read and write permissions for the data providers are mutually exclusive so that sensitive data cannot be exploited by design.

Algorithm and service providers will gain access to all data directories. This is necessary to be able to fulfill the requirements of specific work packages.

In the final version of the storage layer, we will make sure that algorithms themselves are equipped with their own specific users and fine-grained file permissions.

Specifics of the LOGISTAR Deployment

All LOGISTAR environment specific helper scripts and configurations will be collected on our GitLab code repository, provided by Dunavnet [x-35]. This will include helper bash scripts to easily replay the creation of Kerberos principals, x509 certificates and the likes. It should be mentioned that we follow the best practice to use tiny Alpine Docker Base images wherever possible to reduce the amount of disk space docker containers take on our cluster and minimise the number of different behaviours throughout the cluster's core components.

This has also the consequence that we start from a fresh Alpine Linux upon startup of the container and everything required to start the specific implementation will be collected in a script, starting with init-, e.g. init-namenode.sh, init-datanode.sh, init-kerberos.sh and so forth. Those scripts will be hosted on our GitLab and made available to the starting Docker container via our own HTTP server on the bootstrap server as indicated above.

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

Scripts

Create Kerberos Principals

Although this only has to be done once for a domain, we have to make sure the outcome is consistent throughout installations. A simple bash script allows us to create the initial principals and their respective key tab files required to run the basic HDFS services.

Our script takes some arguments: first the Kerberos domain, second the principals that should be created, e.g. HDFS, yarn, and others, thirdly the actual fully qualified domain names for which each principal will be created. Additionally we can specify for which principals key tabs files should be exported. Note that in order to run Hadoop services using those key tabs, it is required that these contain more than one principal, in addition to the service principal this is in most cases the HTTP principal.

Two excerpts of the bash script can be seen below, first the creation of the principals:

```
if [ $CREATE_PRINCIPALS == "true" ]; then
  for PRINCIPAL in $PRINCIPALS; do
      for DOMAIN in $DOMAINS; do
      kadmin.local -q "addprinc -randkey \
         $PRINCIPAL/$DOMAIN@$REALM"
      done
  done
fi
```

And second the export of the corresponding keytab files:

```
for PRINCIPAL in $KEYTAB_PRINCIPALS; do
 for DOMAIN in $DOMAINS; do
  kadmin.local -q "xst -norandkey -k \
   $KEYTAB_DIRECTORY/$PRINCIPAL.$DOMAIN.keytab \
   $PRINCIPAL/$DOMAIN HTTP/$DOMAIN"
 done
done
```

Starting Namenodes and Datanodes

As described above namenodes and datanodes must be started using the dedicated service user, “hdfs” in this case. The actual start script is included in the init-*, preparing the application running inside the Docker container. A sample for the namenode is given below.

```
$ cd /mnt/mesos/sandbox/hadoop-2.9.0
$ sudo -u hdfs ./sbin/hadoop-daemon.sh --config \
    /mnt/mesos/sandbox/hadoop-2.9.0/etc/hadoop \
    -- script hdfs start namenode
```

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

Conventions

To make development of further services as easy as possible, we propose a set of conventions and best practices. These measures are also important to guarantee a consistent system and to maintain a clean environment.

Hostnames

All nodes' hostnames within our infrastructure make use of the project's top level domain LOGISTAR-project.eu, they start with "node-" and an incremented numbers, e.g. node-one.LOGISTAR-project.eu, node-two.LOGISTAR-project.eu. This will make it easy to create further subdomains and to adapt the creation of certificates and Kerberos principals by the corresponding scripts.

Docker Images

Alpine JDK8

We will make sure to use as few different Docker base images as possible. All Docker images should be plain, unaltered Alpine Linux images. For applications based on the java programming language, this will either be the Alpine JDK or the Alpine JRE image. Those images will be used without modification from the perspective of Docker. All necessary resources, like binaries or init scripts will be downloaded into the running Docker container and the application is supposed to be configured by an init shell or bash script.

It should be noted that by virtue of DC/OS, Mesosphere Marathon and Apache Mesos, all downloads that are specified in the application's service description will be downloaded to a specific directory within the running Docker container: /mnt/mesos/sandbox. All applications must use this directory as their application's parent directory. For example our hadoop installation will be found inside the container in the following directory: /mnt/mesos/sandbox/hadoop-2.9.0.

Persistent Volumes

We will mount a specific directory from all nodes of the cluster on the bootstrap server. This can be used for two specific cases. First to have the chance to persist data of an arbitrary application without having to care about the placement of the application within the cluster and secondly for resources that should not be exposed via our HTTP server hosting downloadable resources, for example for certificates and Kerberos keytab files. If a service deployed to DC/OS wants to make use of those common directories, it must contain the volume mapping in its marathon application configuration. An example of how to map a directory between the Docker container and its host can be seen below.

```
...
"container": {
  "type" : "DOCKER",
    "docker": {
      "image": "openjdk:8-jdk-alpine",
      "forcePullImage": false
    },
    "volumes":[
      {
        "containerPath": "/var/dcos/data",
```

```
        "hostPath": "/var/dcos/data",
        "mode": "RW"
      }
    ],
...
```

In the above example, the directory specified under “hostPath” is, as described above, also mounted on our bootstrap server. The running application will then use this directory as a base directory for whatever it needs. As an example, we have configured HDFS’s data directory to be /var/dcos/data/hadoop/2_9_0/datanode/ and /var/dcos/data/hadoop/2_9_0/namenode. This also shows our directory naming convention. Besides using above base directory, it is agreed to use a service identifier, like “hadoop” or “kerberos”, as the first subdirectory of the base folder. Those service directories will have a child directory specifying the version of the service using underscores instead of dots. The directory structure below that version directory is free for the implementer to choose and dependent on the application.

2.5.7. Conclusions on Deployment Infrastructure of LOGISTAR

For the LOGISTAR project’s anticipated workload, we will setup a state-of-the-art cloud infrastructure, ready to host micro-services using various runtime environments, Docker among them, as the preferred option. This infrastructure will put algorithm developers in a position to be able to make use of hardware resources (CPU, RAM,..) that go beyond those of a single computer and hence put the whole project’s IT onto a solid foundation.

As the first application to run on above infrastructure we will expose HDFS, the Hadoop Distributed File System as the main, fail-safe, clustered and state-of-the-art storage layer of the LOGISTAR project. By that, the storage layer can be used as the file system for our project’s resources as well as the storage backend for other components, like the triple store Apache Rya. In addition to that, the setup of the whole storage layer will serve as a template and a showcase of best practices and conventions with respect to the project’s cloud infrastructure.

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

3. Enrichment and Preprocessing of Metadata and Data

3.1. Required Capabilities

LOGISTAR will deliver a platform for commercialisation of logistic services. In this section, we describe the expected capabilities with respect to such services and the relation to enrichment and preprocessing of data and metadata.

Many requirements were clear at the stage of writing the plan of the projects and are therefore reflected in the description of work. Additional requirements from the stakeholders are collected in the frames of WP1. All the requirements are reflected in D1.1 “Market research: interviews, user needs & functional requirements analysis”. Among others, some key findings of D1.1 feature the following:

- Comprehensive search and browse mechanisms on data & service should be provided for internal use to develop services for the LOGISTAR platform. The metadata search will be provided by WP2, together with the leveraged and enriched metadata.
- LOGISTAR shall take into account Volume, Velocity and Variety needs for efficient data analytics.
- LOGISTAR WP2 shall provide mechanisms for data quality assessment and improvement and take care about interoperability and standards for the logistics domain together with data from open data sources. This requirement will be addressed by WP2 together with WP6 by providing a homogeneous approach to describing the interfaces of different services, therefore enabling the interoperability of the services.

Furthermore, security and privacy challenges are – without any doubt - challenges that are very prominent in the LOGISTAR project

3.2. Interoperability of Data and Services

A key feature to promote a greater interoperability of the LOGISTAR system (i) internally and (ii) with 3rd party sub-systems (e.g. Transport Management Systems, TMS of the use case partners and above) is to specify the use of, where possible, standard API connectors based upon REST or message protocols, rather than on technology and program language-specific APIs. These APIs support documentation and interface description validation, the approval and release of services on the LOGISTAR platform, and supports semantic enrichment of data and metadata, and the input data for services as well as GUI for service and data description and publication.

3.3. Data Analytics and Big Data Technologies

In terms of requirements elicitation, it is found out that for most of the industries / use cases, the challenges of high volumes, velocities and heterogeneity of data are the focal point.

WP2 and WP6 will thereby support scalable data analytics (together with the service work packages WP 3, 4, and 5) in the corresponding services for the logistics domain.

Moreover, for many organisations (even large ones), a service of data consulting would be of a key importance. Such a service could help the organisations to create their own data agenda and help to:

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

- manage security of data;
- leverage commercial value of data;
- conduct data analysis.

3.4. Semantic Enrichment

In order to efficiently interlink, search, and recommend datasets, there is a need for efficient metadata mapping and entity extraction mechanisms. Such mechanisms allow to provide a mapping of metadata predicates (either semi-automatically or pre-defined), therefore allowing to automatically fill in the necessary descriptions of the datasets and services. The linking mechanisms help to enrich the information with the help of the background knowledge graphs.

In logistics and for the LOGISTAR platform and use cases, there is a need for (domain-specific) thesauri and ontologies. This may indicate additional opportunities for commercialisation of controlled vocabularies as datasets in the LOGISTAR platform.

3.5. Data Access

Analysing issues and developing fusion algorithms for data access on the closed to semi-closed to open data spectrum, are taking the specific access conditions into account.

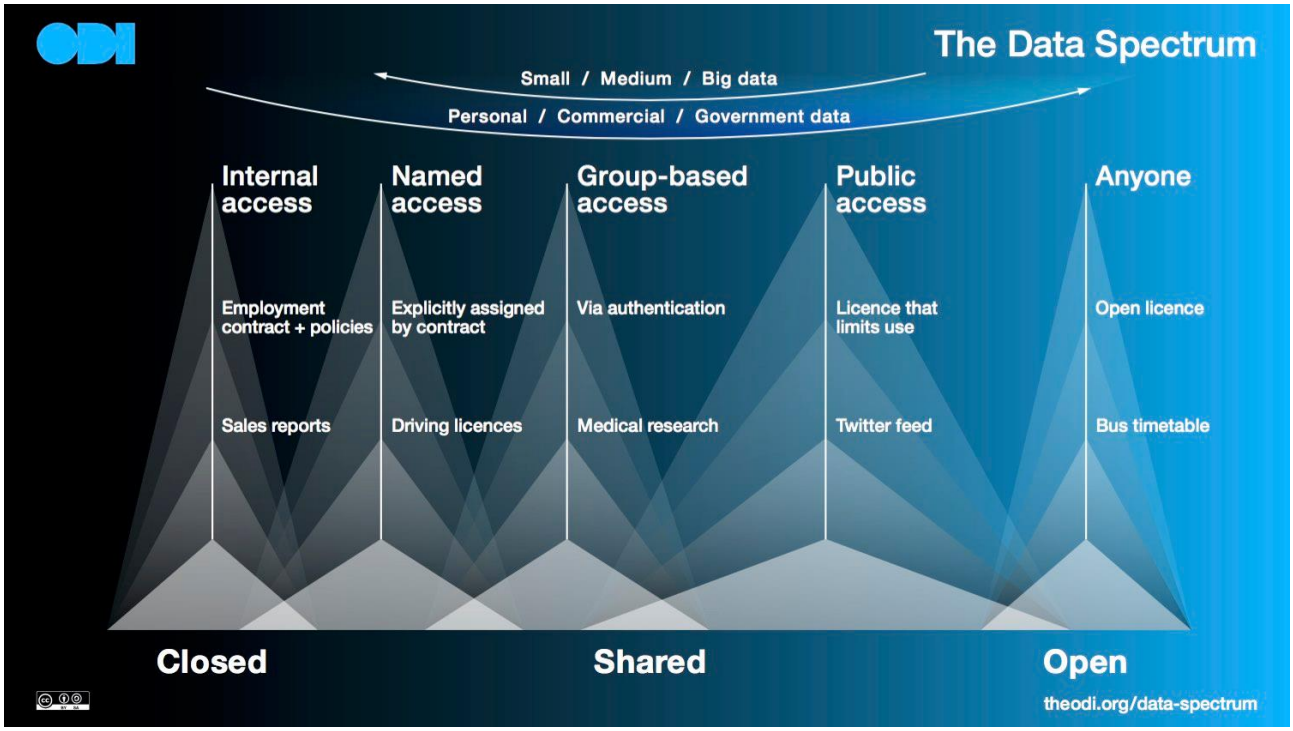

Figure 3: The ODI (Open Data Institute) Data Spectrum.

Such classification of data (along the ODI Data Spectrum) will be taken into account for the creation of controlled vocabularies and thereby metadata enrichment as part of the LOGISTAR Knowledge Graph (see further below in this document, please).

Additionally, secure data sharing / access, privacy, licensing, and mashups were identified as very important aspects for the LOGISTAR project.

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

3.6. Metadata Workflows

Metadata-related Workflow

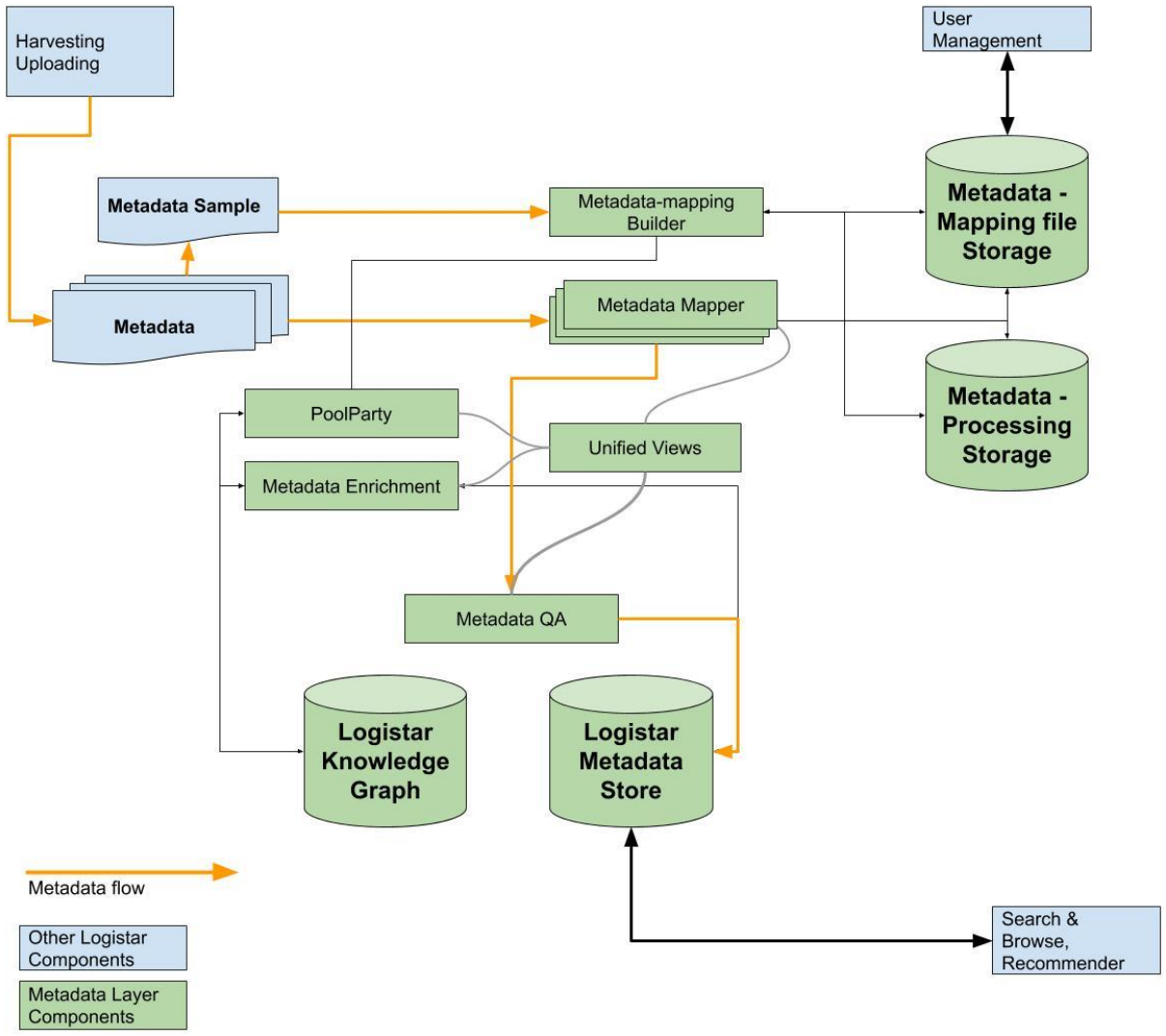

Figure 4: Metadata-related Workflow.

The metadata will be provided for every dataset stored in the LOGISTAR platform. The figure above provides a general overview of the metadata-related workflow, i.e. how the platform deals with metadata. The metadata is stored centrally. When a new (data) asset arrives (either provided manually by a registered user or harvested by the LOGISTAR harvester), its metadata is first analysed and mapped to the LOGISTAR metadata scheme. If the checks passed, the metadata for the asset is stored into the LOGISTAR metadata store.

Moreover, if the users of the LOGISTAR platform are able to provide their own vocabularies for improved analysis of the metadata, then these vocabularies may also be taken into account.

This is of high importance to ensure to provide harmonised data to WP 3, 4, and 5 where the LOGISTAR services are developed and then integrated in WP6 into the LOGISTAR platform. As an

example, imagine that two different Transport Management Systems are using different attributes (controlled vocabularies, term lists, etc.) for COUNTRIES. TMS1 uses ISO standard codes 2 characters like AT (Austria), IT (Italy), UK (United Kingdom etc) BUT the TMS2 uses the full names like Austria, Italy and United Kingdom. Then the LOGISTAR system is not able to understand that e.g. AT = Austria is the the same concept (what a human being would easily understand usually) - thereby mappings between these vocabularies are needed (models that ensure mapping between - in our example - AT = Austria) when harvesting data and metadata to ensure a harmonised data landscape in the LOGISTAR platform.

3.7. Metadata Mapping

In the table below, an example of a predefined metadata mapping is given. In the column “Predicates”, there are 4 sub-columns representing different vocabularies. Different entries from these vocabularies could be used to specify the same property. For example, the owner of the dataset can be specified either with “dct:publisher” (i.e. “publisher” entry of the DCT vocabulary) or with the “provide” entry of schema.org.

Identifier	Definition/Description	Predicates			
		DCAT	DCT	FOAF	schema.org
UID	Unique identifier of the LOGISTAR data		dct:identifier		
Name (Title)	The name identifier of the LOGISTAR data		dct:title		
Description	A description of the LOGISTAR dataset		dct:description		
Owner	ID of the service owner within the LOGISTAR		dct:publisher		provider
Contact Point	This property contains contact information that can be used for sending comments about the dataset.				
Licence	Terms of use		dct:license		
Domain	Application domain in which the dataset is located	dcat:themeTaxonomy			

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

Category	Data type on which a LOGISTAR service is built upon	dcat:theme			category
Documentation	References to further documentation of the LOGISTAR dataset	dcat:landingPage		foaf:homepage	
Tags	Free annotations describing the LOGISTAR dataset	dcat:keyword			
Created	Date and time of LOGISTAR data creation (automatically generated by LOGISTAR)		dct:issued		
Last Modified	Date and time of last LOGISTAR data modification		dct:modified		
Version	This property contains a version number or other version designation of the dataset	owl:versionInfo			
Version Info	The current version number of the dataset				softwareVersion
Version Notes	Documentation of LOGISTAR dataset versions				releaseNotes
Service / Data Dependencies	Dependencies to other LOGISTAR datasets or services (e.g. preprocessing, clustering, format conversion)				

Table 2 Examples of Metadata Mapping

Different platforms naturally define distinct metadata schemes to describe their assets (services, datasets and other digital assets). While importing a dataset itself from a different platform may be straight-forward, importing the dataset's description might not be not possible without manual effort. Therefore it would be necessary to store the mappings between external metadata schemes and the LOGISTAR's metadata scheme.

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

The mapping could be specific for each external platform because we cannot assume that the same URIs are used to denote the same entities in the same sense across different platforms. Therefore, even if the predicate “dcat:landingPage” of LOGISTAR platform is mapped to the predicate “foaf:homepage” of platform A (e.g. a Transport Management System of a use case partner in LOGISTAR) then it does not mean that the mapping “dcat:landingPage” of LOGISTAR to “foaf:homepage” of platform B is a valid mapping. However, it is very likely to be the case. Therefore, this information is reused to suggest this mapping to the user, however, the user is responsible for reviewing and confirming this information.

Moreover, it could be that a 3rd party platform have defined their own vocabulary (or its own extension of an existing vocabulary) to describe some properties. In this case, a comparison of the predicate name may yield useful insights and suggest a mapping to a predicate in the LOGISTAR metadata scheme.

Within the frame of this task, we plan to develop a tool that provides a GUI for creating mappings between metadata schemes. Moreover, the tool will be capable of providing initial suggestions to the user for the mappings. These suggestions are based on:

1. Previous knowledge of the predicate mappings for different schemes;
2. Comparison of the predicate names.

Finally, in this task, we will pre-define several mappings to some of the most important partner systems (use case partners TMS) to ensure seamless synchronisation out of the box.

3.8. Semantic Enrichment

Taxonomies, Thesauri and Knowledge Graphs

The understanding of data depends on the context. One expects metadata to be concise, therefore the context is usually poor. However, the description is more extensive and includes concepts that provide additional insights for experts. In order to leverage this information, we employ background knowledge in the form of a taxonomy, and later in the form of a thesaurus that later will be expanded as a knowledge graph. This thesaurus provides additional information and rich context to understand the concepts and interlink the data. This will enable the *semantic enrichment* of metadata (and where possible also data) to improve interpretability of the metadata.

In the course of the semantic enrichment process, named entities are going to be extracted from the metadata (and if possible, data). We only aim at extracting the entities contained in a thesaurus, therefore, one essential requirement for a good entity extraction is a good thesaurus, because it defines which entities are going to be discovered, linked, and enriched. Within the frame of this task we will identify which thesauri could be reused and how well does it fit our purposes. It is possible to use domain-specific thesauri to better interlink domain-specific descriptions, however, for this purpose it is necessary to know *a priori* that the descriptions belong to a certain domain and to have the corresponding domain-specific thesauri.

Pattern Recognition

When the entities are extracted, we obtain a set of concepts describing the asset. Moreover, the thesaurus defines relations between different concepts and even allows introducing different similarity measures between concepts. Taking into account data from the DMA platform about the

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

asset, such as rating, usage, etc., one could recognize the pattern that makes the asset useful, highly ranked, etc. Having a thesaurus featuring the extracted concepts is an advantage because it provides formalized background knowledge, thereby enabling more flexible methods for extracting patterns.

In the context of this task we will develop approaches for pattern recognition with the help of thesauri.

Thesaurus-less Interlinking

The modern state-of-the-art techniques for extracting keywords may enable a possibility to interlink descriptions without having background thesauri. The task is to find words that are common in both descriptions, and, therefore, interlink the descriptions. However, most common words will not be keywords, but rather stop-words or general-purpose words. Therefore, it is important to interlink only keywords that are specific for describing and understanding the contents of the described assets.

The disadvantage of such an approach is that it is not possible to enrich the information based on any thesaurus. The advantage is that it is not necessary to have a thesaurus to interlink the data. Therefore, the approach could be useful for finding similar assets and recommending assets. In terms of this task, we may evaluate the efficiency of different approaches to accomplish thesaurus-less interlinking.

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

4. Data Models and the LOGISTAR Knowledge Graph on Logistics

As already mentioned in the previous sections, data models, controlled vocabularies as well as taxonomies, thesauri, ontologies and finally knowledge graphs are important for interoperability of data, for smooth data integration and harmonisation but also for data quality and the provision of explainability (in regards to explainable Artificial Intelligence, XAI).

This whole area falls into the section of symbolic AI, and together and in the form of an interplay with statistical AI builds the overall spectrum of Artificial Intelligence (AI).

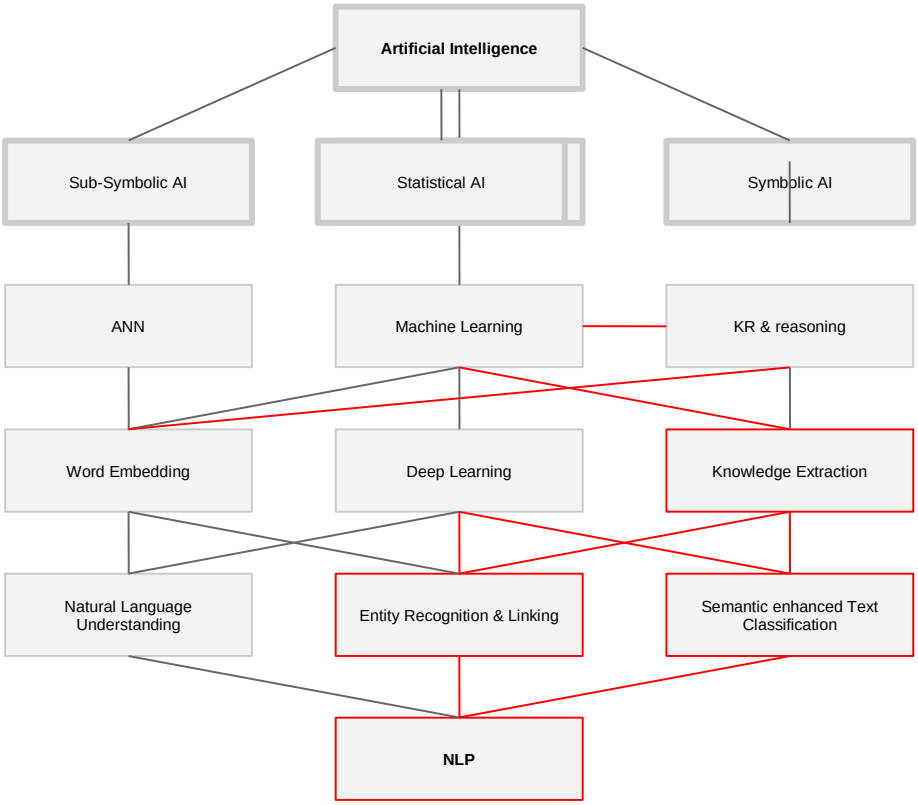

Figure 5 Overview of Artificial Intelligence

Where work packages 3, 4 and 5 focus more on the field of statistical AI, WP2 has its focus – beside on storage – more on symbolic AI to enable smooth and efficient data integration and pre-processing. Nevertheless statistical AI methodologies (e.g. machine learning on entity linking) are used also in WP2, and for the final services of LOGISTAR, the full range of AI methods will be used.

As shown in the following figure, WP2 provides a '4-layered data architecture', that provides the whole storage layer of all data and content of LOGISTAR, that is connected with the LOGISTAR metadata layer that provides information about the data, that is connected with the semantic layer providing links between resources in the data (e.g. countries or goods etc.) and thereby context and meaning, and finally the navigation logic – that here can be seen as the interface of data provision to the service development in work packages 3, 4 and 5.

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

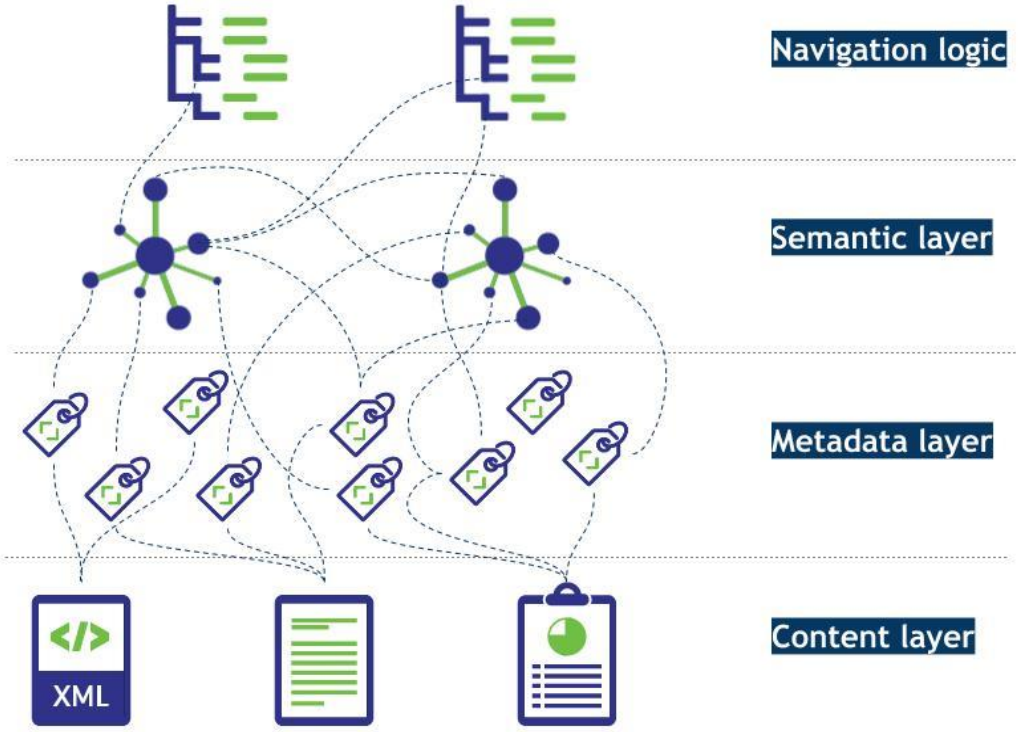

Figure 6 Four-layered data architecture in LOGISTAR

To make use of these mentioned benefits of knowledge models and a knowledge graph (interoperability, data quality, interlinked data etc.) existing standards and models have been identified and analysed and will be used as a basis where possible and/or will be adapted according to the requirements of LOGISTAR to unleash the full potential of such models.

Data models in this regards are

- Datex 2 (https://datex2.eu/)
- RailML (https://www.railml.org/en/)
- TAF/TSI (http://www.raildata.coop/taf)
- Logistics Interoperability Model (LIM, https://www.gs1.org/lim)

For modelling of metadata: DCAT-AP (https://joinup.ec.europa.eu/solution/dcat-application-profile-data-portals-europe) will be used, that is the *de facto* standard for metadata in the area of European open data (open data to be used in the LOGISTAR project and integrated with closed and semi-closed data from project partners and above).

Taxonomies, thesauri and ontologies are identified at the moment of the creation of this deliverable and will be identified and analysed in a next step, to evaluate if such can be used and reused (such sources are identified through the whole project duration).

Furthermore, the LOGISTAR data samples have been analysed as well as possible (as only little data samples have been available at the time of creation of this deliverable) – and will be analysed

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

in detail in the coming weeks – and relevant controlled vocabularies and attributes have been identified that will be used for mappings to ensure smooth and efficient data integration and harmonisation:

- Countries
- Cities
- Points of Interests
- …

All of these data models, controlled vocabularies, taxonomies, thesauri and ontologies that are relevant for the final data layer of LOGISTAR will be harvested, adapted where needed and imported into the PoolParty Semantic Suite software of SWC (https://www.poolparty.biz), where these models are maintained and interlinked and also can be further developed along LOGISTAR requirements over time.

To develop the final LOGISTAR Knowledge Graph, these interlinked models and vocabularies are connected with the metadata and the data itself of LOGISTAR where possible and are helpful to realise mappings, data harmonisation, data enrichment etc. to ensure interoperability, high data quality and a harmonised data basis for LOGISTAR services and platform.

A good example for Knowledge Graphs is for instance the Google Knowledge Graph that connects the whole Google search index and combines it with Google news, pictures or YouTube videos. If someone searches for example for 'Who was the inventor of the World Wide Web (WWW)' the WWW is shown as a resource on the right hand side of the search results. There, a person is mentioned – as the WWW has been invented by this person, that is Sir Tim Berners Lee – and by clicking on the given link of this name, the Knowledge Graph provides detailed information about this person on the right hand side of the search result. Doing this shows that this person is working for the MIT – the Massachusetts Technology Institute - that is provided as a link again, and the user is able to click this link and thereby learn more about the MIT, and so on and so forth.

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

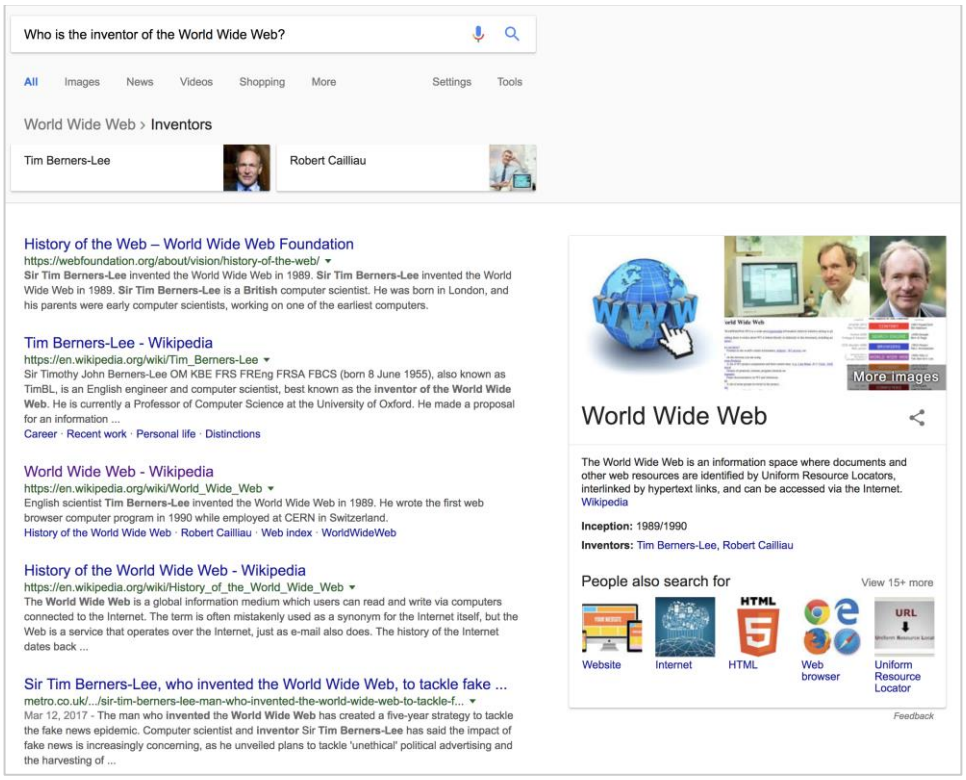

Figure 7 The Google Knowledge Graph – example: World Wide Web

This example of the Google Knowledge Graph explains the basic idea of knowledge graphs and/or the semantic web and/or Linked (Open) Data (LOD), namely that resources can be discovered by following links between objects (or here things that have persistent URIs (identifiers).

Another good example for a Knowledge Graph is the Siemens Industry Knowledge Graph:

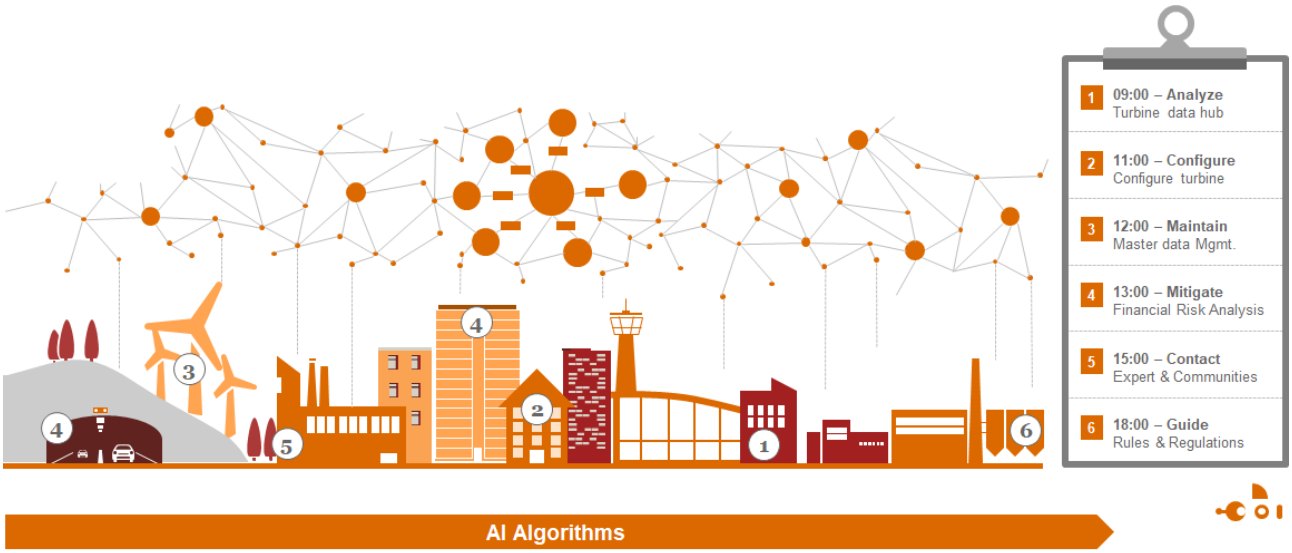

Figure 8 The Siemens Industry Knowledge Graph.

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

5. Conclusion

The specific data and data analysis and data processing requirements (big data, real time data, variety of data and data sources) in LOGISTAR as described in this document set the ground for the needs in regards to infrastructure, deployment, data storage and data pre-processing for the LOGISTAR data collections and enrichment pipelines.

To fulfill these needs comprehensive data and metadata analysis has been carried out and will be carried out in the project, interfaces to Transport Management Systems need to be evaluated in a next step and a powerful 4 node cluster to process big data performant and scalable has been specified and set up for LOGISTAR as described in this document.

Metadata and data pipelines have been specified along such needs – and along the (linked) data life cycle, means from harvesting, over ingestion, cleaning etc to analysis – and will be implemented as a next step in the project to realise (i) data harvesting, (ii) data ingestion and cleaning, (iii) data and metadata enrichment and storage and (iv) pre-processing of data and harmonized data provision for the WPs 3, 4, 5 and 6 to develop services and the overall LOGISTAR platform.

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

List of abbreviations and acronyms

AI	Artificial Intelligence
API	Application Programming Interface
DB	Database
DCAT	Data Catalog Vocabulary
DNS	Domain Name Service
GDPR	General Data Protection Regulation
H2020	Horizon 2020 Programme
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IoT	Internet of Things
ISO	International Organization for Standardization
Json	JavaScript Object Notation
LD	Linked Data
MD	Metadata
MS Word	Microsoft Word
NDA	Non-Disclosure Agreement
ODI	The Open Data Institute
PDF	Portable Document Format
PII	Personal Identifiable Information
RDF	Resource Description Framework
SSL	Secure Socket Layer
TLS	Transport Layer Security
TMS	Transport Management System
XLS	Microsoft Excel
XML	Extensible Markup Language
W3C	World Wide Web Consortium
WP	Work Package

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

List of References

[x-1] https://www.docker.com/
[x-2] https://blog.newrelic.com/engineering/container-orchestration-explained/
[x-3] https://docs.docker.com/engine/swarm/
[x-4] https://kubernetes.io/de/
[x-5] https://mesosphere.github.io/marathon/
[x-6] https://www.openshift.com/
[x-7] https://www.big-data-europe.eu/
[x-8] https://dcos.io/
[x-9] https://www.hetzner.de/
[x-10] https://www.poolparty.biz/information-application-security/
[x-11] https://www.hetzner.de/dedicated-rootserver/ex42-nvme
[x-12] https://www.centos.org/download/
[x-13] https://mesos.apache.org/
[x-14] https://en.wikipedia.org/wiki/Apache_Mesos
[x-15] https://ai.google/research/pubs/pub43438
[x-16] http://mesos.apache.org/documentation/latest/powered-by-mesos/
[x-17] https://www.nginx.com/
[x-18] https://hadoop.apache.org/docs/r2.9.0/
[x-19] https://hadoop.apache.org/
[x-20] http://web.mit.edu/kerberos/www/
[x-21] https://www.varonis.com/blog/kerberos-authentication-explained/
[x-22] https://tools.ietf.org/html/rfc4120
[x-23] https://alpinelinux.org/
[x-24] https://wiki.alpinelinux.org/wiki/Alpine_Linux_package_management
[x-25] https://pkgs.alpinelinux.org/package/v3.7/main/aarch64/krb5-libs
[x-26] https://web.mit.edu/kerberos/krb5-1.12/doc/admin/admin_commands/krb5kdc.html
[x-27] https://web.mit.edu/kerberos/krb5-devel/doc/admin/admin_commands/kadmind.html
[x-28] https://web.mit.edu/kerberos/krb5-1.12/doc/admin/admin_commands/kdb5_util.html
[x-29] https://web.mit.edu/kerberos/krb5-1.12/doc/admin/conf_files/krb5_conf.html
[x-30] https://web.mit.edu/kerberos/krb5-1.12/doc/admin/conf_files/kdc_conf.html
[x-31] https://letsencrypt.org/
[x-32] https://github.com/Neilpang/acme.sh
[x-33] https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/SecureMode.html#Secure_DataNode
[x-34] https://en.wikipedia.org/wiki/POSIX
[x-35] https://gitlab-LOGISTAR.dunavnet.eu/turnguard-LOGISTAR/dcos

Annex I - Guidelines for LOGISTAR UC Data Samples

Agreed Guidelines for LOGISTAR UC Data Samples

Timely coverage: at least 1 month = April 2019 (any additional data welcome!)

Geographical coverage: use case specific (England, UK, UK & Ireland, DE - AT - IT, ES (?))

Format: export from systems (DBs) if possible as XLS or CSV file

PLUS: DB model information (table info, tables joins etc)

PLUS: attributes description (e.g. list of countries, list of Cities and sample entry / information which standard used, e.g. Country ISO 2 characters like AT or full name like Austria)

PLUS: provide a remark which data comes from which system (ID, title etc)

Important: the structure needs to be understood, means which data points are included, e.g. ID, title, country etc.

Contact point for questions: *email address of wp2 team member* (removed for this deliverable)

Upload will take place in two steps:

1. until our cluster and final hardware is setup: https://cloud.semantic-web.at/index.php/login (an owncloud instance, hosted within the networks of Semantic Web Company, see: https://owncloud.org/).
   Remark: UC p NOT see / have access to each others data.
   a. Every UC partner will receive login credentials via email by SWC
   b. All tech partners will receive login credentials via SWC
2. As soon as our infrastructure (LOGISTAR infrastructure) is up and running, we will use HDFS (Hadoop File System, see: https://hadoop.apache.org/), which will make it is also possible for any algorithms to access data sources, based on agreed access policies
   a. Every UC partner will receive login credentials for their own space by SWC
   b. Agreed access policies will be enforced by the file system (https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsPermissionsGuide.html)
   c. Every UC partner will receive a written tutorial on how to interact with the filesystem

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.

Annex II - Criteria for Data Catalogue Evaluation

List of specified Criteria

1. enterprise readiness (in particular scalability)
2. possibility to integrate with LOGISTAR tools (and APIs in general, search api, read & write)
3. tagging
4. license and product type (on-premise vs cloud)
5. price (along license model)
6. technologies used
7. basis info about vendor or main contributors and support (if open source)
8. (main) customers
9. Existing SWC schema mapper should be easy to be implemented (so PP integration / UViews integration) - from perspective of docu is enough for 1st step (edited)
10. Documentation and quality of documentation
11. The way and how easy it is to create a schema and out-of-the box metadata schemas (e.g. DCAT) in place
12. Versioning of metadata sets
13. GDPR compliant?
14. Easy to use for data collection
15. Search over metadata - powerful
16. Rating and reviews/comments
17. Security & governance - policies / roles etc (freigabe)
    a. Data security / protection

Further comments

- create metadata schemas
- create metadata entry, edit, delete etc
- search over metadata sets (edited)

Nice to read for setting up the evaluation framework

- https://www.datasciencecentral.com/profiles/blogs/a-step-by-step-guide-to-build-a-data-catalog
- https://tdwi.org/articles/2018/09/17/diq-all-data-cataloging-comes-of-age.aspx
- GERMAN language: https://www.informatica.com/de/products/big-data/enterprise-data-catalog.html
- maybe: https://docs.microsoft.com/en-us/azure/data-catalog/data-catalog-what-is-data-catalog

List of identified data catalogues to be evaluated

- https://atlas.apache.org/
- data.world
- CKAN
- DKAN
- Azure Data Catalog
- Enterprise Data Catalog by Informatica
- Alation Enterprise Data Catalogue

LOGISTAR project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no. 769142.