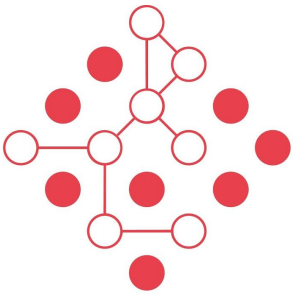


DATA MARKET

AUSTRIA

www.datamarket.at

D6.1 Service Technology Specification and Development Roadmap

Deliverable number	***D6.1***
Dissemination level	***Public***
Delivery date	***31 May 2017***
Status	***Final***
Author(s)	***Johann Höchtl (DUK), Artem Revenko (SWC, Hermann Fürntratt, Herwig Zeiner (JRS)***

FFG

The Data Market Austria Project has received funding from the programme "ICT of the Future" of the Austrian Research Promotion Agency (FFG) and the Austrian Ministry for Transport, Innovation and Technology (Project 855404)

bmvit

Executive Summary

This document presents the DMA Data Technology Specification and Development Roadmap regarding the ongoing work in WP6 of the DMA project. The document relates to deliverable D2.2 “D2.2 Community-driven Data-Services Ecosystem Requirements (1st version)” which formulates the requirements of the DMA community. It will be explained in what way the requirements formulated in D2.2 are going to be addressed. The document gives an overview about the components which are going to be developed in WP6 and how these components are planned to be employed to provide DMA core services. The expected outputs of these services are described and a procedure of describing a 3rd party service for DMA platform is addressed.

Table of Contents

List of Abbreviations

AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
CI/CD	Continuous Integration / Continuous Delivery
DCAT	Data Catalog
DCT	Dublin Core Metadata Initiative Terms
FOAF	Friend of a Friend
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
IDL	Interface Definition Language
RPC	Remote Procedure Call
SLA	Service Level Agreement
SLO	Service Level Objective
SOA	Service oriented Architecture
TLS	Transport Layer Security
URI	Universal Resource Identifier

1 Introduction

A software service is a tool that is capable of taking the input in a specified format and providing a specified output. A user of the service is not assumed to have any other knowledge about the service except from input and output specification and service level agreements. Moreover, the deployment and maintenance of the service is in the responsibility of the infrastructure provider, not the user.

Data Market Austria will deliver a platform for commercialization of data and services. In order to make it possible, some core services, like access control service, are necessary. A comprehensive list of requirements is collected in WP2 and referenced here. In Chapter 2 we describe how different required capabilities are addressed in WP6.

Besides the core services developed by the members of the consortium, the DMA platform will be able to work with 3rd party services. In order to do this efficiently there is a need of a unified framework for processing these services. In particular, we define a metadata scheme for service description and make an overview of the tools to describe the interfaces of the 3rd party services.

We provide a development plan for each individual task of WP6 in this deliverable. Since the tasks are to a large extent independent of each other, there is no common development timeframe. We focus on describing the expected functionalities rather than time perspective. The natural time plan is imposed by the description of work and deliverables 6.2 and 6.3.

Interdependencies to other WPs

The infrastructure of the DMA platform is developed in WP4. As any service is dependent on the infrastructure for its deployment all the tasks are dependent on the outputs of WP4.

The access policies and the metadata description is applicable to both datasets and services in the DMA platform. Tasks 6.3 processes the metadata and Tasks 6.4 provides access control, therefore the outputs of these tasks are included into the data ingestion pipeline of WP5.

WP7 provides search and brokerage functionalities of the platform. Both functionalities process the metadata and are therefore dependent on the Task 6.3.

Task 6.2 deals closely with the functionalities required in Pilots WP8 and WP9.

2 Required Capabilities

Data Market Austria will deliver a platform for commercialization of data and services. In this deliverable we focus on services and in this chapter we describe the expected capabilities with respect to services.

Many requirements were clear at the stage of writing the plan of the project and are therefore reflected in the description of work. Additional requirements from the communities (first of all from the FFG core domain communities) are collected in the frame of WP2. All the requirements are reflected in D2.2 “Community-driven Data-Services Ecosystem Requirements” (link). By the time of writing this deliverable the D2.2 is not finalized yet, therefore the current status is reflected in this document. Among others, the key findings of D2.2 feature the following:

- ***Comprehensive search and browse mechanisms on data & service should be provided.*** The search will be provided by WP7, however, the basis for the search, namely, the leveraged and enriched metadata is to be prepared in Task 6.3.

- ***DMA shall take into account Volume, Velocity and Variety needs for efficient data analytics.*** Relevant Big Data challenges are to be addressed in Task 6.2
- ***DMA shall provide mechanisms for data quality assessment and improvement and take care about interoperability and standards across industries and domains***. This requirement will be addressed in Task 6.1 via providing a homogeneous approach to describing the interfaces of different services, therefore enabling the interoperability of the services.

Though the security and privacy challenge is not directly reflected in the key findings of D2.2, no doubt that these challenges are prominent. These challenges are addressed in Task 6.4.

2.1 Interoperability of Services

A key feature to promote a greater interoperability of the DMA system with 3rd party sub-systems proposed by the project community 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 DMA platform, exporting service profiles for matchmaking in WP7, semantic enrichment of service profiles, and guided input process and GUI for service description and publication.

This requirement stems from the description of work and is also reflected in the key findings of D2.2.

2.2 Data Analytics and Big Data Technologies

In frames of requirements elicitation it is found out that in four out of five FFG core domains (except AAL) Data mining and Machine learning are the key functionalities for the domains. For most of the industries, the challenges of high volumes, velocities and heterogeneity of data are in the focal point.

Task 6.2 will focus on scalable data analytics in the corresponding services for the mobility and earth observation domain. Therefore, this requirement will be addressed in this task.

Moreover, for many even big organizations a service of data consulting would be of key importance. Such a service could help the organizations to create their own data agenda and help to

- manage security of data, i.e. help the organizations to manage the data in a secure way and prevent any possibility of sensitive data becoming available to the outside world;
- leverage commercial value of data;
- conduct data analysis.

2.3 Semantic Enrichment

In order to efficiently interlink, search, and recommend datasets and services, 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 enrich the information with the help of background knowledge graphs.

In D2.2 it is reported that for many FFG core domains there is a need for (domain-specific) thesauri and ontologies. This may indicate additional opportunities for commercialization of controlled vocabularies as datasets in the DMA platform.

2.4 Data Access

Access to data and services which are not hosted or provided through the centralized DMA storage mechanism, need to be protected from unauthorized access and, once access is granted, tracked and traced to support billing and to enforce SLAs. This requires the inception of services which provide a seamless integration into the remaining DMA service infrastructure and enable more cautious DMA users to participate in the network without compromising data protection regulations or disclosing data unencrypted. This also touches the domains of licensing and mashups, as many of these access algorithms will be user/usage specific and will have to be implemented in a pluggable manner. Licensing and mashup principles have been identified to be very important aspects for FFG core domains and are further detailed in section 5.4.

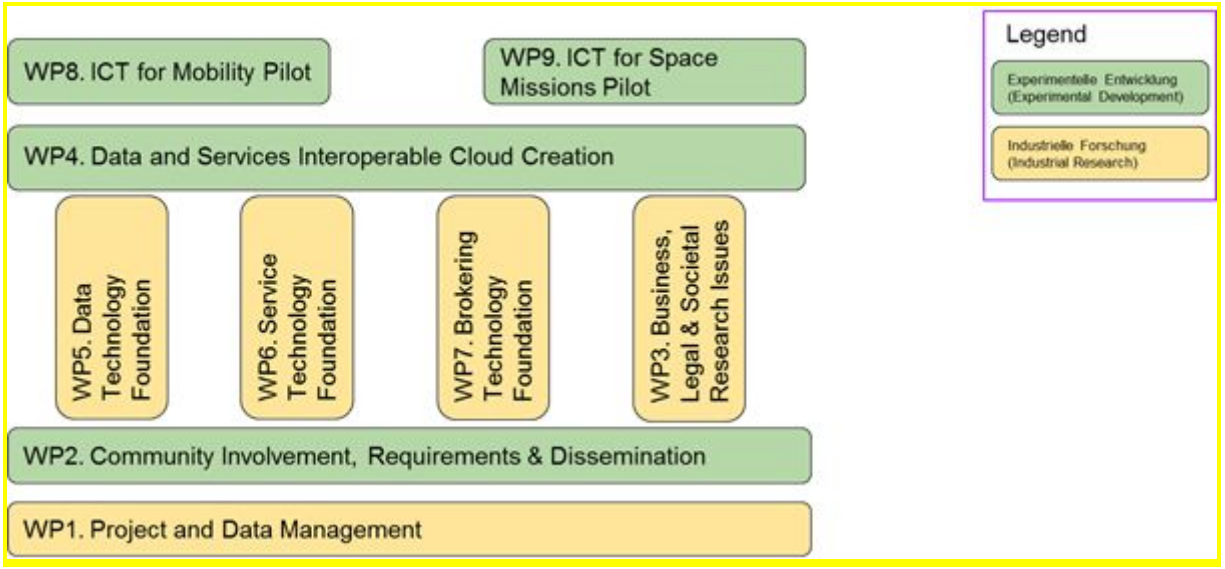
3 Technology Foundation

3.1 Approach

The Work Packages are divided into four tracks:

1. Technology Foundation (WP5, WP6, WP7);
2. Business, Community and Regulatory Aspects (WP2, WP3);
3. Interoperable Cloud (WP4);
4. Pilots (WP8, WP9).

As illustrated in the following diagram, Tracks 1-3 (i.e. everything but the Pilots) provide the technological, business, regulatory and infrastructural foundations of the ecosystem on which the pilots are hosted. The pilots provide application area-specific “biotopes” in the ecosystem which facilitate work in these areas, and illustrate the use of these biotopes with demonstrators that implement the solution to a concrete challenge.



The goal of WP6 is to provide three classes of fundamental functionalities: First, we develop novel approaches to scalable data pre-processing and processing algorithms for the DMA platform. Second, we develop methods for semantic enrichment and linking of data. The third type of functionality is for analysing and fusing distributed data with differing access levels. Finally, we provide APIs for application developers to use these algorithms as a service.

3.2 Service Description

Each service in the DMA platform will have a corresponding metadata describing it. The purpose of this description is to enable comparison, recommendations, discovering, and composition of services. The description consists of tuples of attribute predicate and attribute value, where the attribute predicate defines which piece of information is provided with the help of the corresponding value. For example, for describing the license of the service the predicate could be "dct:license" and the attribute value would be the text of a license.

The tables below reflect which properties of the service should be described. The metadata specification document and the predicates in particular are not finalized by the time of writing this deliverable, therefore, the current state is reflected in the current document. The up-to-date metadata document[1]. The description is divided into 5 topical tables:

1. General Service Properties (20 items);
2. Technical Properties (5 items);
3. Performance (5 items);
4. Security (2 items);
5. Rating (1 item).

Colorcodes:

- Mandatory
- Recommended
- Optional
- DMA Specific

Identifier	Definition/Description
UID	Unique identifier of the DMA service
Name (Title)	The name identifier of the DMA service
Description	A description of the DMA service
Owner	ID of the service owner within the DMA
Contact Point	This property contains contact information that can be used for sending comments about the Service.
Licence	Terms of use
Domain	Application domain in which the service is located
Category	Data type on which the DMA service is built upon
Price Model	Provides some information about the pricing system used by this service
Price Range	The price range of the service

[1] Can be found here: https://drive.google.com/open?id=1V7rbosKCBhs51VsqXPyXDQrnv6Pjn0eRtiThMNxSmDI

Documentation	References to further documentation of the DMA service
Tags	Free annotations describing the DMA service
Created	Date and time of DMA service creation (automatically generated by DMA)
Last Modified	Date and time of last DMA service modification
Version	This property contains a version number or other version designation of the Service
Version Info	The current version number of the service
Version Notes	Documentation of DMA service versions
Service Dependencies	Dependencies on other DMA services (e.g. preprocessing, clustering, format conversion)
Dataset Dependencies	Dependencies to certain DMA datasets (e.g. lookup-, training-data)
Service Type	The type of service a customer can expect
Related / similar Service	Automatically generated field

Table 1: General Service Properties

Identifier	Definition/Description
Input Interface	Interface with which the DMA service receives input
Output Interface	Interface with which the DMA service is delivering output
Operating Location	Describes where the DMA service operates
Operating Platform	Describes the operating (eco) system the service requires to run
Processing Type	Describes the temporal behaviour / response of the DMA service

Table 2: Technical Properties

Identifier	Definition/Description
Availability	Describes service accessibility and availability. It describes whether the data service can actually be used. It is typically necessary to specify numeric values for availability to make meaningful statements that are useful for data service customers. It describes the number of service provisioning requests completed within a defined time period over the total number of service provisioning requests, expressed as a percentage.
Response Time	Response time is the time interval between the time when a data service customer initiated an event and

	when a data service provider initiated an event in response to that initial event.
Capacity	Capacity is the maximum amount of some property of a data service.
Computational Resources	Describes the computational resources necessary to run the service
Service Levels	SLAs: which service levels (response time, resolve time etc) are provided

Table 3: Performance

Identifier	Definition/Description
Service Reliability	Describes the ability of the service to perform its function correctly and without failure over some specified period
Security	Describes the security features of the DMA service

Table 4: Security

Identifier	Definition/Description
Quality of Experience Rating	The overall acceptability of the DMA service as perceived subjectively by the end-user

Table 5: Rating

3.3 Service Levels Agreements

The primary goal of service level agreements (SLA) between service providers and service customers is to clarify expectations and assumptions at both sides and hence make services more comparable and comprehensible.

Service level agreements may e.g. express payment conditions like pay per use, long term contracts, advertising, or others. In case that service level objectives (SLO) could not be fulfilled, SLAs may also contain compensation terms and conditions, such as refunds of charges, free services or the like.

Description of relevant non-functional elements[2]:

- **Availability** - a key service level objective. Describes the time in a defined period the service is available, over the total possible available time, expressed as a percentage.
  - ***Level of uptime***;
  - ***Percentage of successful requests***;
  - ***Percentage of timely service provisioning requests***.
- **Response Time** - can be a highly significant aspect of the user experience.
  - ***Average response time*** refers to the statistical mean over a set of cloud service

[2] Cloud Service Level Agreement Standardisation Guidelines: http://ec.europa.eu/newsroom/dae/document.cfm?action=display&doc_id=6138

response time observations for a particular form of request.
    - ***Maximum response time*** refers to the maximum response time target for a given particular form of request.
- **Capacity** - the maximum amount of some property.
    - ***Number of simultaneous connections*** refers to the maximum number of separate connections to the cloud service at one time.
    - ***Number of simultaneous cloud service users*** refers to a target for the maximum number of separate cloud service customer users that can be using the cloud service at one time.
    - ***Service throughput*** refers to the minimum number of specified requests that can be processed by the cloud service in a stated time period. (e.g. Requests per minute).
- **Computational resources** - refers to the amount of a resources necessary to run a service
  Example resources include data storage, memory, number of CPU cores.
- **Service reliability** - allowable downtime, which accounts for scheduled maintenance
  Also covers the capability of the service to deal with failures and to avoid loss of service or loss of data in the face of such failures.
- **Security capabilities**
    - ***User authentication and identity assurance level;***
    - ***Authentication***;
    - ***Mean time required to revoke user access***;
    - ***Third party authentication support***.

4 Proposed Approach for Service Development in WP6

4.1 Service API Description

Part of the DMA API governance is to take care that all service APIs follow commonly stated rules. Therefore several open source standardized description tools and frameworks have been evaluated in order to facilitate and unify the workflow of API development.

Main requirements are:

- automatic API creation on various software platforms,
- testing with generated stubs and mock-up services,
- generated description in human- and machine-readable format.

Further important criteria are:

- sufficient critical mass of developers that use the tools and drive development,
- and support by large corporations.

Different velocities in data communication via API apply constraints for service protocols, see Table 6.

Service data velocity	batch-mode, periodic	near-realtime	realtime
Protocol	RESTful HTTP/1.1	AMQP[3]	HTTP/2

Table 6: Protocol types associated with service data velocity

[3] AMQP is fully realtime-capable, but using a RESTful HTTP “integration broker” slows communication down.

For RESTful APIs, which are intended for plain HTTP/1.1 based services, four different API composition tools/toolsets have been evaluated:

- Swagger tools around the OpenAPI Specification[4]
- RAML Specification[5]
- API Blueprint[6]
- ServiceStack[7]

Amongst them, the Open API Initiative is governed by the Linux Foundation and offers great potential for community driven development. Currently, version 3 is in the process of being released (RC2), with improvements in organization, better naming, being more descriptive, and adjusting to cloud based CI/CD workflows.

For task driven APIs, which are based on the standardized advanced message queueing protocol (AMQP) Apache's open source AMQP broker Qpid offers a RESTful API, that can benefit from the API tools described in the previous section.

For HTTP/2 based services, which are intended for high performance bi-directional communication and streaming, the open sourced gRPC toolchain provides an interface description language (IDL) and tools to generate implementations for all important platforms and languages.

The frameworks for creating a service API description require various licenses, like

- Apache License V 2.0
- Eclipse Public License V 1.0
- 3-clause BSD License
- MIT License

All of them are business-friendly and hence allow to be used within the DMA portal.

4.2 Upload Service

In case a service is hosted in the DMA cloud, it requires the DMA upload service to get deployed. A standard container format is stipulated by the DMA service orchestration platform, which may also perform security checks (e.g. a malware-scan), and clearing processes according to the metadata profile and DMA security policies (e.g. vulnerability checks) in order to ensure the integrity of the deployed service.

Applicable cluster policies, (e.g. restart and update policy along with resource policies dealing with quotas, auto-scaling, or similar) along with terms and conditions for payment will be defined in the DMA service level agreement, which gets its input from WP3 and WP4.

5 Development Plan

5.1 Task 6.1 Service API and Profile Creation Framework

A typical service API lifecycle traverses five states:

1. Define
2. Develop

[4] https://github.com/OAI/OpenAPI-Specification

[5] http://raml.org

[6] https://github.com/apiaryio/api-blueprint/blob/master/API%20Blueprint%20Specification.md

[7] https://servicestack.net/

3. Publish
4. Support, and
5. Retire the API.

During this task, a DMA internal service is built, that allows potential DMA service providers to register their own services within the DMA portal and manage API lifecycle states. Therefore, a metadata profile is generated for each service. If available, the service API itself can be defined using an interface definition language (IDL). Based on this IDL data, a human-readable service API description can be generated. The metadata profile along with the IDL data is delivered to the DMA Broker service for semantic enrichment purposes. Furthermore the IDL data in the Broker environment enables service discovery and client stub code generation for different programming languages.

In the initial state, the API metadata profile within the DMA portal is created depending on general properties of the service vocabulary, mainly the domain property (with values e.g. Financial, Transport, Travel, Weather, ...) and the processing type property (e.g. query (for search), calculation (for aggregation, clustering, classification,...), content delivery (e.g streaming), decision support (optimization including scheduling of taxi trips, recommender systems)) along with SLA related non-functional properties, the type of the service protocol should be assessed.

It might be necessary to help potential service providers finding the best suitable API choice for their business model. This will be accomplished by showing successful examples. Example space is part of the DMA portal, although mainly available for data, for RESTful services an example space offers mock-up service calls. On the one hand, it should help service providers to overcome old complex monolithic API approaches, that don't work well in contemporary CI/CD pipelines, and on the other hand, it should allow to explore more modern lightweight concepts like micro-services, to find out how a composition of such services can accomplish e.g. billing or assure data-sovereignty.

Potential service providers can also request further information or assistance via a DMA Broker service.

API lifecycle state	DMA management activity
Service definition	Create metadata profile according to DMA metadata vocabulary. Create unique Provider ID if necessary. Fill in auto-generated metadata. Show possible service level agreements. Explain billing strategies. Send metadata profile to DMA Broker service on submission or discard the profile item on cancellation.
Service development	Generate an IDL item for RESTful HTTP/1.1 service protocol, or HTTP/2 bi-directional protocol. If available generate IDL data for AMQP. If available, create API documentation. Send IDL data and documentation to DMA Broker service.
Service publishing	Update profile metadata. Fill in auto-generated metadata. Send metadata to DMA Broker service.
Service support	Update profile metadata. Send metadata to DMA Broker service.
Service retirement	Update profile metadata. Send metadata to DMA Broker service.

Table 7: API lifecycle state bound to managed activities

In the API lifecycle management, organizing the API repositories according to a consistent structure is useful, as it helps to keep track of APIs drifting slowly apart over time from the initial service requirements, for instance: if service developer fix bugs and at the same time add new features with a slightly different semantic. This may especially become problematic if services are connected with other service providers and the new semantic changes don't fit well into the service API chain any more.

5.2 Task 6.2 Novel Approaches to Large Scale Data Analysis

This task develops novel scalable data analytics algorithms to support users in their decisions and actions. These scalable analytics algorithms are part of a toolbox. A service is an algorithm (or a set of algorithms), combined with the corresponding API. In this section we describe and give a first outline of first set of algorithms.

Mobility Services

With logistic companies and organizations producing more and more data, larger sets of interesting datasets have become available. Furthermore, some of these logistics/mobility driven organizations are embracing the concept of open data (e.g. open street map), enabling public dissemination and the use of the data by any interested partner. In a modern fleet management solution, such as taxi companies or other logistics service companies, they are based on the latest mobile technologies and backend software services. A full set of vehicle information is collected and communicated in near real-time to a central service where it is recorded for statistical purposes. This includes the actual position of cars, the starting and ending time of trips, users, starting and ending km, fuel-filling etc. Based on the position information, it is already possible to visually identify streets or places with high traffic or traffic jam issues, allowing the driver to dynamically select alternative routes. The current services, however, only have limited prediction algorithms implemented.

Here, DMA comes into play to provide data and services to support capacity and demand planning that is location dependent. Both public data offers, e.g. actual arrival and departure times of planes and trains, information on large cultural or sport events or congresses, and proprietary data, e.g. knowledge on amount of persons at places based on mobile phone usage, or weather prediction can be used to implement prediction models and thus allow the fleet operator an optimized planning of the available capacities. New services can also be developed based on such information, e.g. if I book a trip to the airport for a particular flight, I could be informed in case of a delayed departure and my trip can be automatically rescheduled. This is a clear additional benefit for customers. On the other hand, while having a complete fleet (> 50 vehicle) distributed within a region, the vehicles and the data of the mobile phone network can be used to collect data within small and large regions which can then be offered to other parties in the DMA eco-system.

In the following section we describe and give a first outline of a first set of challenging scaleable algorithms, namely shared rides for taxis with time windows and a taxi heat map, which are relevant for the mobility use scenarios in DMA.

Shared rides for taxis with time windows

The time and distance optimization of transports - particularly transports persons and materials - in the mobility/logistics sector is a significant contribution to the efficiency and quality improvement of the overall data oriented services model. The overall goals are: efficient allocation of trip orders, high utilization of transport personnel and their vehicles, as well as minimization of transport routes. Overall this will lead to considerable cost savings. Finally, short waiting times for

transported customers increase the satisfaction of customers and service users. Since the short-term changes have to be taken into account at all times, adherence to time constraints is essential for the underlying algorithms. In order to meet these requirements, the planned transport optimization system must, on one hand, be based on intelligent, goal-oriented and practicable as well as very fast optimization algorithms (e.g. scheduling should be done in less than 30sec), and, on the other hand, on an efficient state-of-the-art software architecture. The central scheduling unit with all the available fleet information can automatically calculate best schedules for individual and/or shared trips and the corresponding routes according to minimize e.g. fuel or energy consumption and to minimize carbon footprints in this way, too. Relevant parts of these optimization models are:

In our scenario, we study a dynamic problem of taxi sharing (or ride sharing) with time windows. In DMA, we consider a scenario where people needing a taxi or interested in getting a ride use a phone app to designate their pick-up and destination points. Therefore it is also possible to define a maximal allowable time to reach the destination. The first goal is to maximize the number of shared trips. Another advantage is that the people in a taxi can share the costs of the trip. The overall problem is a dynamic one, since new calls for taxis arrive on demand. To increase the shared rides we can also try to minimize the distance traveled or the duration of rides. If several criteria are used, we can add them into a corresponding cost function. However, the overall optimization problem is NP-Hard.

Orders: The time and distance optimization of a transport request for a customer consists of the starting point, the destination point and a desired pickup time. Furthermore, a maximum deviation time should be considered as well. As a result, time windows can be defined for the pick-up at the start node or at the destination point. Due to the definition of the time windows, a premature arrival at the destination is not possible, while a delayed arrival at the destination is possible. The time windows are strict. However, only deviations from the critical destinations (e.g. airport) are very critical, deviations for non-critical destination points are not punished in the same way.

Vehicles: The vehicles are placed on one or more depots. The vehicles of a fleet are equipped differently (heterogeneous fleet). Each vehicle has a certain capacity for each type of transport (e.g. person or material)

Constraints: An important constraint of scheduling dynamic transport orders is that not all information is known at the beginning of the planning process. The number of transport orders and the traffic condition, for example, are constantly changing in such a service.

Computing: The shared ride algorithm works in such a way that it reschedules the available transport resources (trips, vehicles ...). The calculations that lead to this allocation are carried out at regular intervals, so that all new and also short-term intervals can be processed. All available transport resources within a defined planning horizon are considered. The transport resources must then be informed of their next scheduled and/or re-scheduled orders via their mobile devices.

Heatmap

The taxi heatmap pilot is a dynamic service for taxi fleet optimisation to demonstrates the usage of DMA services. The goal is to deliver a heatmap continuously to predict the demand for taxi vehicles. This will provide benefits for both, taxi platform and end-users. Dynamic predictive scheduling method will raise efficiency and utilization of taxi vehicles, since they will be in the right place at the right time. On the other side, end-users will receive improved service through minimised travel time what will increase their overall experience.

Taxi heat map implementation: This pilot includes a combination of open and closed data and

scheduling algorithms that will produce predictive strategies for a taxi company. Taxi heatmap requires following data packages: set of vehicle information, historical traffic flow, cellular data, weather reports, events, airport and train station schedules etc. Employing artificial intelligence (AI) will accelerate the development of this pilot by a factor of thousands, therefore it was decided to use AI in developing and improving algorithms and prediction. Base algorithms include regression analysis, deep neural networks etc. that imply automatic test-bench.

Observed territory (e.g. the city of Linz) is divided in a grid, where each part of the grid present the range of 200x200m2 (as it is shown on the picture):

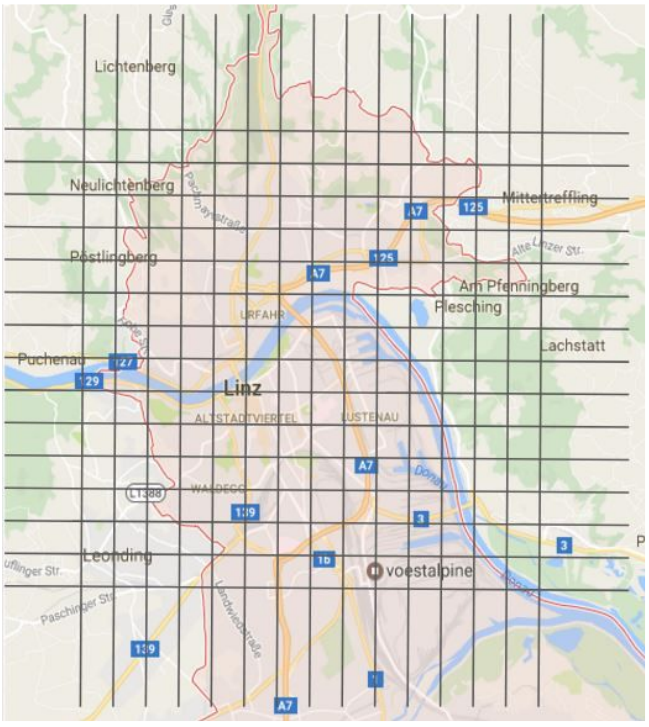


Figure: grid for base algorithms analysis

As AI learn by repetition, each part of the grid will be trained separately with its own algorithm. Later on, algorithms will compete against each other and that training data will be processed. Based on received data it will be possible to develop a ground for prediction. For each part of the grid, cluster backend receives following data:

- temperature conditions (rain, snow, wind...) from Austrian Weather Agency
- Mobility of people (with following description: age, gender, type of a trip..) from T-Mobile
- Train schedules information from ÖBB
- Traffic information from Asfinag etc.

As the amount of received data is growing, the learning curve will grow and prediction will be more precise.

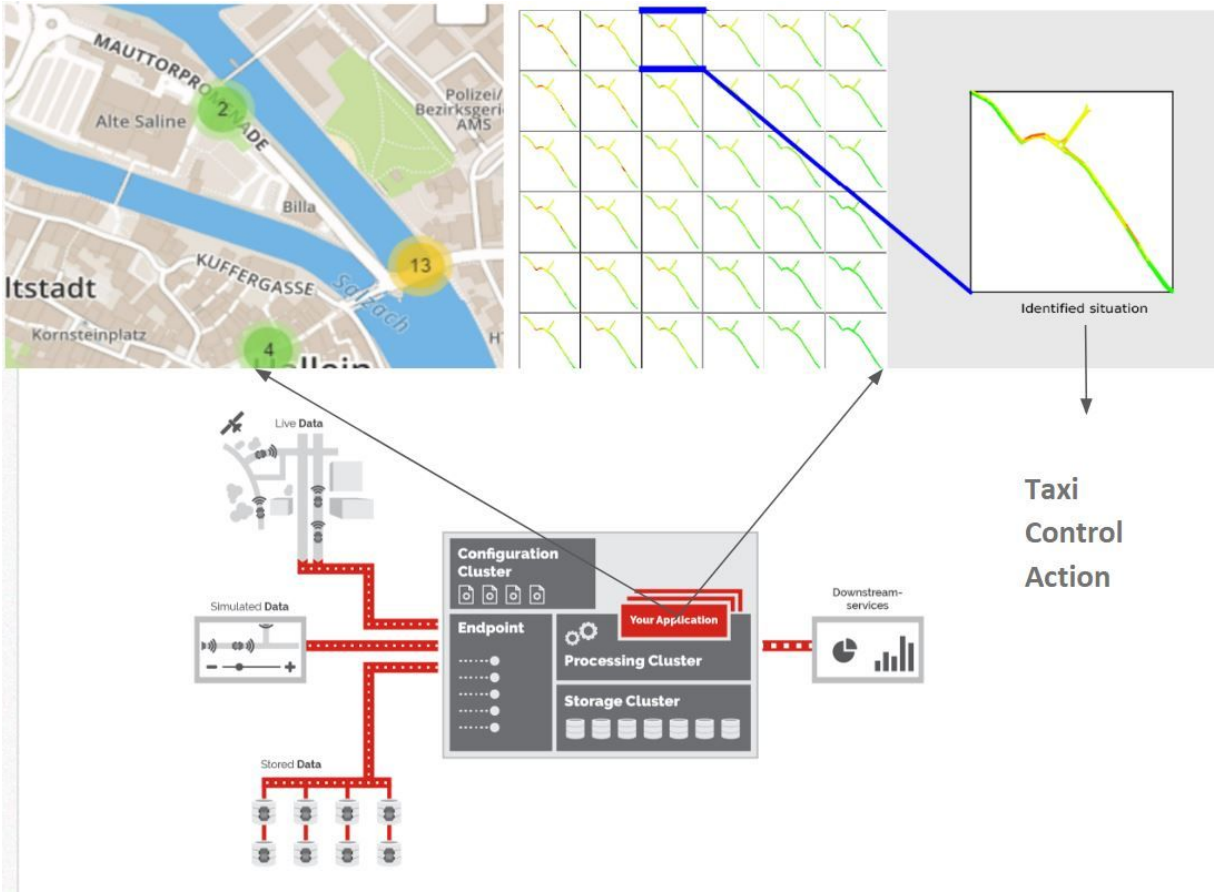


Figure: IoT Cluster

Constraints: One of the biggest constraints for this pilot was data privacy. In order to address it, it was decided that the service will run in the T-Mobile data center, since they have full access to the mobility data. This will diminish all concerns with data privacy, while the output of the algorithms is checked by a data privacy algorithm of the T-Mobile.

All in all, AIs provide a situation assessment for each observed part. All these situations help AI in defining rules (which can be adapted). Employed AI algorithms that compete and create rules will be able in short time to predict the need of taxi vehicles in near future. Increase in available data is directly proportional with the increase of prediction accuracy.

Forestry Services

Forest areas are often affected by natural impacts which can cause economic damage up to damage on infrastructure. Storms or rockfalls can occur immediately and everywhere and it is a-priori not known where changes or forest damages can occur. Changes of specific forest parameters affect rockfall propagatio; a continuous updating of modelling results gives important information about the protective function of forests. Therefore it is necessary to observe very large areas on a regular base. New satellite systems offer a very efficient tool to obtain information of possible change caused by biotic or abiotic mechanisms. The twin satellites SENTINEL 2A and 2B (launch 7th March 2017) are very well adapted for this task, because they enable a monitoring every 5 days. This section describes the list of services for the earth observation pilots.

Forest (Change) Monitoring

The monitoring of forest areas offers a very efficient tool to obtain information of possible change caused by biotic or abiotic mechanisms. However, it is a-priori not known where changes or forest damage can occur. In order to overcome this drawback it is necessary to observe very large areas

on a regular base. In this context the new SENTINEL satellite system offers a very powerful tool with high spatial, spectral and temporal resolutions. As this system has a swath width of up to 290 km and a repetition rate of every 5 days the frequency for updates can fully satisfy forest monitoring needs. Storage capacities have to be adapted to the huge amount of data recorded for all systems, including Landsat TM. This vector of threshold values must be specified as percentage of the pre-event pixel value which shall define the 'corridor' for non-change pixels. Pixel values outside the range +/- Threshold are set to the value 1 in the change-map file, all other pixels are set tono-data. NOTE: the thresholds may be specified individually for all channels as this parameter allows the user to define a vector of percentage values. In case there are more input bands than there are threshold values, the last entry in the threshold vector is reused for these bands

Storm Damage Resilience

Nowadays complex data sets are available for many forest related applications. With respect to storm events this situation is a prerequisite to analyze storm damage resilience. Forest experts emphasize the need of certain key parameters for this evaluation. For example, data on wind speed, topography or forest features are a prerequisite, together with satellite derived information, to model this resilience. From the availability of the data these parameters can be grouped into:

- environmental factors (e.g. such as wind speed, geomorphological features, geological / pedological information);
- tree attributes (e.g. such as tree type involving the root system, tree height);
- stand situation (e.g. development stage, species mixture, crown pattern, stand structure, intensity of thinning).

From the above listed input data sets the resilience of forest areas against potential storm events can be modelled and thus the foresters can obtain important information for their management plans.

Updated Rockfall Propagation Modelling

Area-wide high-resolution rockfall modelling results based on LiDAR data are available for the whole Province of Styria. Two main aspects were taken into account: (a) the identification of potential source zones, and (b) the estimation of rockfall propagation zones. The runout distances were modelled by velocity calculation based on a one parameter friction model. Whereas potential source zones do not change within short time, the modelled run-out zones strongly depend on friction values based on forest parameters. The service will consist of updated rockfall propagation models whenever relevant forest changes are recorded (Service 1).

Recommender Services

Lively interactions between consumers and providers of datasets and service are an important success factor of DMA. WP7 develops a recommender system to foster these interaction by providing recommendations of possible combinations of services and datasets. Hence, the recommender system, or recommender for short, will automatically provide suggestions for possible combinations of the datasets and services available to users browsing through the offers available at the DMA platform. Additionally, a search service is also provided, where users can manually trigger a search for any desired offer on DMA.

The recommender and the search service need information describing the datasets and services in DMA. Hence, connections to different data sources providing these descriptions are needed. Metadata from service and data providers characterising their offers are used as information sources. An automated assessment of service performances provides additional information about the offered services. Information about the interaction of users with the datasets and services is another information source. Interactions can be any action taken at the DMA platform by a user like reading the description, creating a new offer, or purchasing a dataset or service on the DMA

platform. The recommender and search service offer interfaces to integrate them into the DMA portal. The interfaces and components developed by WP7 are depicted in below:

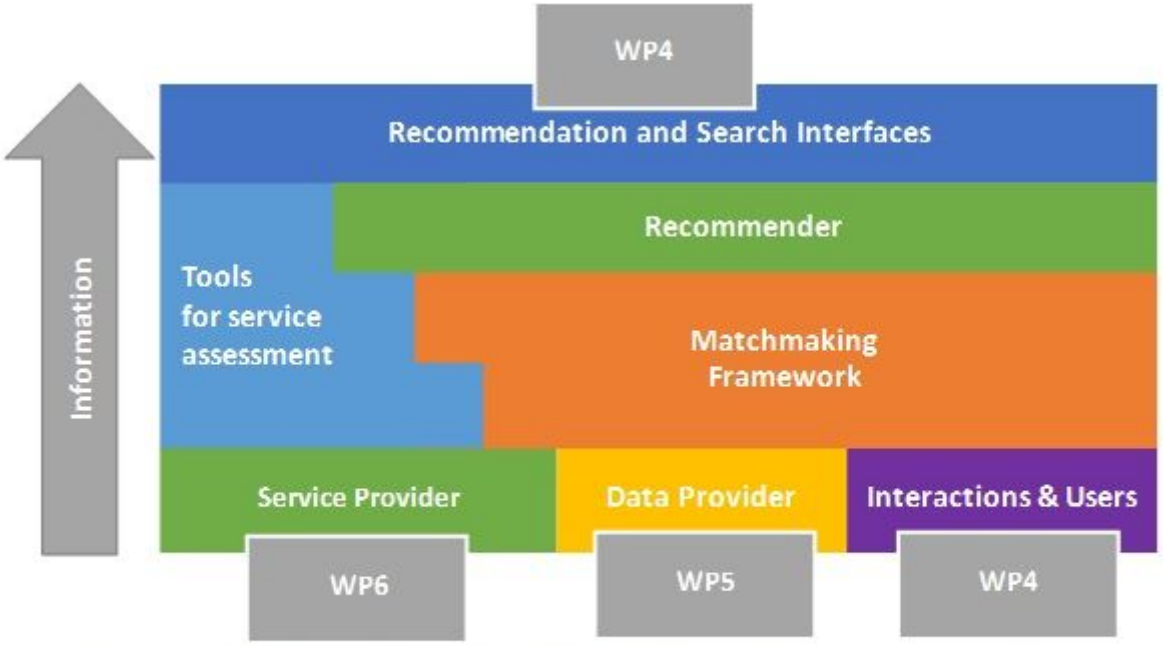


Figure 1: WP7 uses input from WP4, WP5, and WP6 to generate the recommendations. The Recommendations are then provided to the user in the portal maintained by WP4

5.3 Task 6.3 Semantic Enrichment and Linking of Data

Metadata-related Workflow

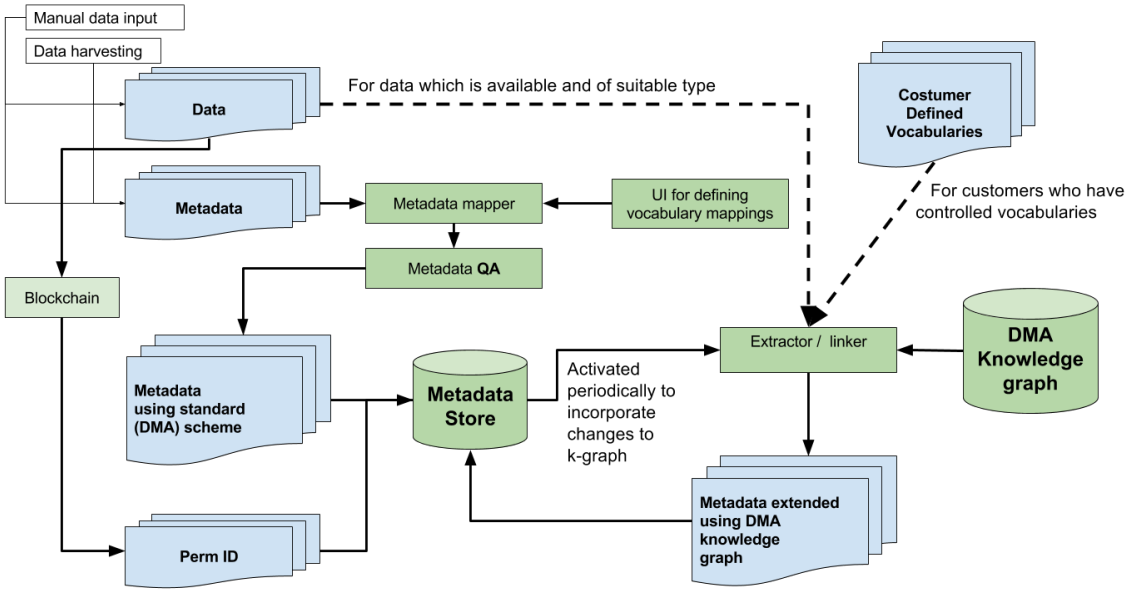


Figure 2: Metadata-related Workflow

The metadata will be provided for every dataset and every service (thereby every asset) stored in the DMA 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 asset arrives (either provided manually by a registered user or harvested by the DMA harvester) its metadata is first analysed and mapped to the DMA metadata scheme. If the checks pass, the metadata for the asset is stored in the DMA metadata store.

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

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 services 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 DMA service		dct:identifier		
Name (Title)	The name of the DMA service		dct:title		
Description	A description of the DMA service		dct:description		
Owner	ID of the service owner within the DMA		dct:publisher		provider
Contact Point	This property contains contact information that can be used for sending comments about the Service.				
Licence	Terms of use		dct:license		
Domain	Application domain in which the service is located	dcat:themeTaxonomy			
Category	Data type on which the DMA service is built upon	dcat:theme			category
Price Model	Provides some information about the pricing system used by this service				
Price Range	The price range of the service				

Documentation	References to further documentation of the DMA service	dcat:landingPage		foaf:homepage	
Tags	Free annotations describing the DMA service	dcat:keyword			
Created	Date and time of DMA service creation (automatically generated by DMA)		dct:issued		
Last Modified	Date and time of last DMA service modification		dct:modified		
Version	This property contains a version number or other version designation of the Service	owl:versionInfo			
Version Info	The current version number of the service				softwareVersion
Version Notes	Documentation of DMA service versions				releaseNotes
Service Dependencies	Dependencies to other DMA services (e.g. preprocessing, clustering, format conversion)				
Dataset Dependencies	Dependencies to certain DMA datasets (e.g. lookup-, training-data)				
Service Type	The type of service a customer can expect				serviceType

Table 8: Metadata Mapping

Different platforms naturally define distinct metadata schemes to describe their assets (not only services, but possibly also 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 possible without manual effort. Therefore it would be necessary to store the mappings between external metadata schemes and the DMA's metadata scheme.

The mapping could be specific for each external platform because we cannot assume that the same URIs are used to denote the same entities across different platforms. Therefore, even if the predicate "dcat:landingPage" of DMA platform is mapped to the predicate "foaf:homepage" of platform A then it does not mean that the mapping "dcat:landingPage" of DMA 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 mappings to the user, however, the user is responsible for reviewing and confirming this information.

Moreover, it could be that an external platform defines its 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 DMA metadata scheme.

Within the frame of this task we plan to develop a tool that provides a GUI for creating mappings between metadata schemes. This 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 data platforms to ensure seamless synchronization out of the box.

Semantic Enrichment

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 thesaurus. This thesaurus provides additional information and rich context to understand the concepts and interlink the data. This will enable the ***semantic enrichment*** of metadata to improve interpretability of the metadata.

In the course of the semantic enrichment process, named entities are going to be extracted from the metadata. 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 fits 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 the rating, usage, etc. of multiple assets stored in the DMA platform one could recognize conceptual patterns that make higher ratings / more usages more likely for the asset. 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. The outcome of this task is a method for solving the specified challenge and its implementation in Python. As follows from the above description, method takes the asset's metadata including ratings and usages, the target attribute (either rating or usages), and the extracted concepts and outputs patterns of concepts that help to maximize the target attribute.

Thesaurus-less Interlinking

The modern state-of-the-art techniques for extracting keywords (Anette Hulth 2003, Hasan, Kazi Saidul, and Vincent Ng 2014) 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 no enrichment is performed. 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 frames of this task we will estimate if thesaurus-less linking is potentially beneficial for the project and, if so, we will evaluate the efficiency of different approaches to accomplish thesaurus-less interlinking.

5.4 Task 6.4 Analysing and Fusing Distributed Data with Differing Access Levels

The analysis and fusion of access to different data sources within the context of a data market is a delicate topic. Several requirements have to be tackled in parallel:

1. Restricting access to data: Depending on the nature of the data, its content, prospected value, intention for publication and legal requirements, most prominently personal data protection regulations, the owner or originator of data has an incentive to restrict the access to data. He/She will restrict the access to data because it is him directly who is the owner of the data, he is the publisher of the data and thus legally responsible or he is acting in the name of a third entity.
2. Billing: Data and services upon data will be provided according to different service level agreements which in turn impose duties and rights for example to the data provider and the data market customer. One of these obligations will be to compensate the provider for providing access to data. In order to charge data usage, it will be necessary to trace who is using what data and to what extend.
3. Provenance tracking: Both the restriction to data as well as the use case of billing can be generalized to concept of provenance tracking. Provenance can be understood as the history of whatever item, be it tangible or digital.

In the case of digital goods, provenance tracking ends most of the time the moment the digital good leaves the sphere of its owner. Techniques like Digital Rights Management (DRM) try to extend that reach by keeping “strings attached” on digital goods and to extend the possibilities of the owner to track provenance beyond its premises.

In DMA, Task 6.4 analyses possible solutions to provide provenance tracking of distributed and heterogeneous data sets and associated use cases. In detail, T6.4 promises to deliver:

i. ***blockchain technology to tackle access to restricted data on a per access basis***. This means to perform a feasibility analysis on using mechanisms provided by blockchain mechanisms to model the manifold and heterogeneous data access use cases expressed in D2.2, D2.3 (Community-driven Data-Services Ecosystem Requirements 1st and 2nd version) and D3.2 (First Report on Business Model Development).
ii. ***access mechanisms to privately stored data on cloud services or personal vaults like Dropbox or Google Docs***. During the inception phase of DMA it was indeed not widely clear whether the DMA would be an exclusive, mostly closed market, or available to the general public. The general openness became more and more clear and as such accessing data hold by private entities on popular cloud platforms is an important use case
iii. ***algorithms for automated anonymisation and pseudonymisation***. The whole discussion on data as the basis of an entire new business system gets constantly encumbered by personal data issues. Detecting personal data within data sets and automatic anonymisation (by means of aggregation or pseudonymisation) would be a holy grail. Unfortunately whole projects are devoted to “solveing” this open issue. On the other hand, reports have shown that even after carefully obscuring credit card data, correlating with easily available data sets was enough to correctly re-identify 90% of credit card purchases to the individual (Montjoye et al. 2015).

Research exchange held with Christian Jung from Fraunhofer/IESE, participating in the german research project Industrial Data Space[8] and discussion with the WP leader led to the decision NOT to pursue automated anonymization / pseudonymization of data in any way.

For algorithmically enforced automated anonymization / pseudonymization to be effective, a rigorous metadata regime would have to be in place, clearly and unambiguously identifying personal data. This would impose a lot of work to the data provider he/she is unlikely to devote. The decision not to provide automatic anonymization of personal data is also supported by the stakeholder workshops performed part of D2.2 (eg. Section 3.2.3.2: ***Data aggregation / anonymizing services which allow to process personalized data. Such a service may be run by a special trustee, who guarantees conformity with privacy regulations***)

iv. private/closed networks: This item was included in T6.4 to decide on the openness level of the DMA which in turn generally affects DMA's architecture and service framework.

During the research done in D5.1: Data Technology Specification and Development Roadmap and the discussions during the first consortial meeting in Salzburg it was decided that DMA will have to support the integration of data sets on private/closed networks without the need to upload such data sets to central infrastructure. As such it is an architectural decision to be addressed by WP5 instead of WP6.

However participating in the DMA ecosystem in a private/closed network has substantial consequences on the fusing of distributed datasets / services which is addressed in detail in the sections further below and affects these components:

- Accessing data and services which shall not be uploaded to central storage. This functionality is required to be realised through a DMA Network Access Connector. This connector requires to provide a GUI on which the shared resources can be configured, indexed by a data indexer/ingestor, upload that information to the central data and service discovery to be included into the distributed search index.
- Not all conceivable access technologies to privately stored data / services can be supported by this DMA Network Access Connector, therefore this connector must provide an infrastructure which supports to plug in customized components which give access to this data and services.
- Data access is required to be traced, guarded, encrypted and signed to be saved from attacks while in transit. Tracing of data access requests and guarding access to data will be provided by the Ethereum private network described further below as well as in D5.1 and D5.2. Encryption and signing is described further below where we detail the DMA message bus component.

v. research on mechanisms to off-load data access stored on personal computers via web technologies like WebRTC and/or web workers: In conjunction with the aforementioned item of private/closed networks, the need to decentralize certain privacy-sensitive and/or computationally heavy tasks to the data end users (providers and customers) becomes apparent (edge computing[9]). As browser technology gradually standardizes to support edge computing, it becomes worthwhile to discuss the browser also as the data access and processing technology next to its obvious role for data presentation and visualisation.

The following subsections will detail the required services to implement the drafted

[8] http://www.industrialdataspace.org/
[9] https://en.wikipedia.org/wiki/Edge_computing

functionalities.

Blockchain technology to restrict (track and govern) access to data

A blockchain in essence is a public ledger which records transactions between participating parties in a way, that records can only make it into the ledger once a qualified majority agreed on the validity of the items to be added to the ledger (consensus) and prevents changes to once added items (append only). Various mechanisms of reward and deterrence exists to reach an equilibrium level between a large number of honest participants and few malicious networks participants.

The various implementations of blockchain technology mostly vary by the used cryptographic algorithms which are used for consensus finding. These consensus finding algorithms have the biggest influence on whether the blockchain can conceptually support unknown parties and still be secure against changes to items. In practice these consensus algorithms also have the biggest influence on the transactional performance (once added items are perceived to be "correct") of the blockchain.

Blockchain implementations also vary concerning the provided functionality or supported use cases. The most prominent blockchain implementation is that of Bitcoin[10] which implements a virtual currency. While the Bitcoin ledger can in theory be used to store arbitrary data besides Bitcoin transactions between anonymous participants, it is impractical to do so, restricting its use case to that of a secure money transfer system. The fact that no more Bitcoins can be spent than are available to a certain user is programmatically enforced. As a consensus finding algorithm proof of work[11] is used which requires every network participant to confirm every network transaction by re-calculating (thus verifying) performed hashes which secure transactions. This is time consuming and prevents the Bitcoin network to be used for speed trading.

Another significant blockchain implementation is that of Ethereum[12], which finds itself on the other extreme, as it is a general purpose ledger (agnostic to what type of transactions are verified by participating parties) and which provides a general purpose execution engine (Ethereum virtual machine EVM) and programming language Solidity and compiler, which translates Solidity source code to EVM which in turn gets executed by the EVM. This general purpose virtual machine makes the Ethereum network appropriate to execute smart contracts. A smart contract is an arrangement between a sender and a receiver to perform a certain action once a set of pre-conditions are met. For example, a smart contract could be defined that once the filling level of a printer reaches 10%, it will search for replacement on the internet, be it a generic refill or an OEM product and automatically buy from the cheapest supplier located within the EU who accepts Paypal payment. All items to this contract can be expressed in code and verified by many parties. The Ethereum virtual machine will perform the required lookups on the internet and once the pre-conditions to the contract are met, perform the transaction, which gets recorded in the ledger (the blockchain) and will be verified by all other participating network members. It is worth noting that the term "smart contract" in itself is problematic as it is not necessarily attached to a "contract" in legal terms[13].

⇒ The pros and cons of different blockchain implementations are discussed in detail by DMA deliverable D5.1: Data Technology Specification and Development Roadmap, Section "Blockchain Technology" and D5.2: DMA Blockchain Design and are not repeated here, but references are made instead where appropriate.

[10] https://bitcoin.org/en/
[11] https://en.wikipedia.org/wiki/Proof-of-work_system
[12] https://ethereum.org/
[13] http://www.cotlegal.com/smart-contracts.html

Consensus has been reached at the DMA consortium meeting in Salzburg[14] that blockchain technology will be used to fulfil the requirements as laid out by the DoW and automated execution of legally binding contracts of the DoW can only be sensibly supported by the Ethereum blockchain. Therefore the following discussion will concentrate on using Ethereum to fulfil the DoW and the outcome of the requirements elicitation of D2.2 (3.2.3, 3.2.4, 3.2.5, 3.2.6, 3.2.7).

In the domain **Billing, Contractual and Legal requirements** the participants of the requirements elicitation workshops expressed the following needs:

Transaction based billing models

Guaranteed quality and legal certainty for the trade

Smart billing and contracting possibilities

Guaranteed quality and legal certainty for the trade

Standardized SLAs

A blockchain records all transactions happening between parties in an electronic public ledger. So if a DMA dataset or service is consumed, a blockchain could record the fact that a dataset or service has been invoked, and successfully delivered/executed according to predefined conditions. Using the Ethereum blockchain these conditions can be expressed programmatically using the Solidity programming language, for example. In detail, a blockchain is able to cover the following needs:

Transaction based billing models: Blockchain analysis tools could browse the public ledger, analyse its nature of transactions and on top of that perform transaction-based billing.

Guaranteed quality and legal certainty for the trade: By default, users on the Ethereum blockchain are technically identifiable - there is a clear and unambiguous trace between transaction originator (sender) and transaction recipient (receiver). However by default, Ethereum users are anonymous, which requires a mechanism to math Ethereum users to real-world entities.

Smart [billing and] contracting possibilities: Ethereum supports the definition and secure execution of smart contracts[15].

Guaranteed quality and legal certainty for the trade: The blockchain provides mechanisms which ensure that

- contractual obligations can be expressed;
- that these obligations (constraints and duties) have been put into force by identifiable contracting partners;
- that the contracting partners invested (valuable computing) resources to execute their respective obligations;
- that other network participants can serve as witnesses to confirm that the will of both contracting parties has been fulfilled;

Currently the implications and intricacies of automated smart contracts in respect to their legal legitimacy in the Austrian legal regime is undefined and will be specified in detail in D3.4 ***Final***

[14] http://wiki.datamarket.at/index.php/DMA_Consortium_Meeting_in_Salzburg_in_April_2017#Results_and_TODOs , Login required

[15] https://github.com/ethereum/go-ethereum/wiki/Contract-Tutorial

Report on Legal, Societal and Cultural Aspects. However, it has already been discussed that the term "smart contract" used in blockchain context is unfortunate as the minority of predefined transactions on blockchains, which are called contracts, are contracts in the legal sense. Therefore a distinction will have to be made between the term contract used in the blockchain context, contract in legal terms and when a blockchain smart contract actually constitutes a legal contract. Currently the implications of legally binding automated execution of contracts has yet to be discussed and is an ongoing topic (Buchleitner and Rabl 2017).

Standardized SLAs: D3.4 will develop model contracts to be executed on the blockchain which will also contain standardized SLA conditions. A graphical rule modeler developed by the DALICC project[16] would make a good starting point to be reused by DMA which translates graphically defined constraints into ODRL[17]. An ODRL to EVM transpiler will be required to make these ODRL-definitions executable on the Ethereum Virtual Machine.

In the domain **Provenance and Security** the following requirements have been expressed:

> ***A mechanism that ensures that only "serious vendors" trade on the DMA***
>
> ***Security in trading and using the data. E.g. mechanisms are in place that ensure, that data is not exploited by non-authorised parties***
>
> ***A high level of security and privacy is an absolute must***
>
> ***Quality, Provenance, SLAs and Security of Data***
>
> ***Coupling between provided data, data supplier and data usage, so that the supplier can track usages at any time***

Once again, the core principle of a blockchain is its public ledger of consensus-secured transactions. Considering data as a digital asset bearing a persistent identifier, this identifier can be used to unambiguously identify a resource. For traded files this identifier could be calculated using a checksum or hashing algorithm. In the case of services, this identifier is harder to define as even a set of input parameters is not guaranteed to produce the same results when the service gets called at different times. In detail, the blockchain is able to cover the expressed needs:

A mechanism that ensures that only "serious vendors" trade on the DMA: Users on the Ethereum network are technically identified yet anonymous. The consequences will be described in the next subsection.

Security in trading and using the data. E.g. mechanisms are in place that ensure, that data is not exploited by non-authorised parties: A blockchain can support this requirement insofar as the access to the data resource can be guarded by a smart contract and only granted when the conditions expressed by the data provider have been met by the data consumer. As the blockchain itself makes a very poor general purpose database, the actual content delivery has to be performed using other channels. Securing the data transmission, guaranteeing that no data gets altered or eavesdropped during the transmission, that it cannot be re-requested once it has been successfully delivered or that data cannot be used beyond the intended purpose is outside the scope of Ethereum[18].

A high level of security and privacy is an absolute must: By default Ethereum users are identifiable

[16] https://www.dalicc.net/
[17] https://www.w3.org/TR/odrl-model/
[18] In theory some of this functionality could also be performed by the Ethereum Virtual Machine EVM but other tried and proofed transport mechanisms are in place

but anonymous.

[Quality], Provenance, SLAs [and Security of Data]: Every transaction is secured in an unalterable, append-only way within the blockchain which fulfils the need to track provenance of

- what data
- according to which conditions (by means of the assigned contract)
- has been sent by whom to whom
- and when.

These features are provided out of the box by the blockchain and fulfil the provenance tracking requirement. The data provider has the possibility to specify its terms of service (SLAs) according to which he is willing to provide data the data consumer has to agree with, otherwise no means to obtain the data or service will be provided.

Data quality and data security are outside the scope of services a blockchain (Ethereum) can provide.

Coupling between provided data, data supplier and data usage, so that the supplier can track usages at any time: As long as the data itself is traded unaltered using the DMA provided infrastructure, the resource assigned persistent identifier will have to be re-used by the system, giving the data provider the means by analysing the publicly available ledger when his data has been re-used. If data gets altered, this persistent identifier will no longer be re-used and the data provider loses the ability to track usage. This is also true if once DMA obtained data is traded outside DMAs communication channels. For usage tracking to be effective, some sort of Digital Rights Management DRM would have to be installed which wraps the data in a container and allows data access only via defined container access methods which in turn report data usage back to the data provider. This functionality is outside the scope of this task.

⇒ Technical details on the DMA persistent identifier mechanism are given in D5.1, section "Conduit Dataset Ingest & Preservation" and "Persistent unique identifier (PID)".

Blockchain technologies are intensively discussed in the domain [NB: energy sector]; There is a high interest in blockchain Technology coming from Industry 4.0.: These two requirements express the wish that the DMA blockchain system might be more than a transparent backend service but made generally available to all data market participants.

Recommendation for implementation

A blockchain mechanisms will play an important role within DMA. As knowledge using blockchain technology is not yet widely available, many aspects which can be covered by eg. Ethereum, shall be abstracted away. To fulfil the relevant use case requirements, the following adaptations and extensions to the core functionality of what a blockchain provides out of the box, have to be provided:

Data access restrictions:

a. on a resource basis: The resource under which data becomes available once the contractual preconditions are met must be dynamically provided to the data consumer. The system has to support a secure and once-only download of data sets;
b. on a sub-resource basis: A data owner provides a high-priced CSV file but the data user is interested only in a subset. Ideally the data access layer will be able to dynamically segment data. This might in turn require semantic annotations to data (what subsets can a data user actually request) which can only be partly automatized.
c. on a service basis: Access restrictions to services will also be provided using the blockchain as the URI under which a service is made available to a service user will be protected by a URI whose contents (the result of the service API call) only becomes available once the

contractual obligations by the service user have been met;

Optional user identification: Every transaction to be made on the DMA network involves a registered entity (human or service entity). Ethereum is only operational after creating an account which remains anonymous and unverified though. To raise the security level and, anticipating the requirement that any legally valid contract requires the participating parties to be known to each other, different user registration levels will have to be supported. Contracts, which shall be legally enforceable, requires DMA users to be identified. Even though participating on the DMA network shall be easy and unobtrusive, the user identity should be provided. Supporting the Austrian Citizen Card to elevate an Ethereum registered user to a legally identified user would also ***likely*** fulfil the legal requirement of being an identifiable entity eligible to participate in legally binding contracts and during legal disputes.

Access to private data vaults

Nowadays non-critical data sets are often stored instead of own premises but on cloud storage. Access to cloud-hosted resources requires two items:

1. Providing an access token to the cloud hosted data sources; and
2. Using the cloud services providers access technology to access/query the data.

Acquiring an access token to authorize a third party service has been thankfully standardized in the last years by means of OAUTH[19], and industry standard developed within the IETF. Popular cloud storage data vaults the DMA should support depends on the intended primary user groups. For B2C that would be Google Drive and Dropbox, for B2B at a minimum Amazon Web Services (AWS)[20]. All of these services allow third party access and the restriction of that access to a subset of the provided resources by means of OAUTH[21] scopes. Scopes are defined by the service provider and implement the granularity at which third parties can access cloud hosted services. OAuth Scopes share the same functionality as SAML Assertions and in fact protocols exists which bridge OAUTH 2.0 with SAML.

Viability for Data Market Austria

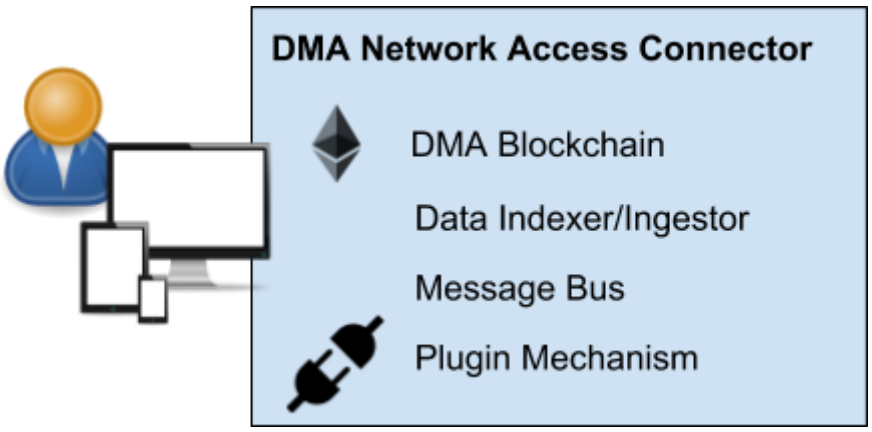
Providing the ability for DMA to access data sets which are under the influence of a data provider yet not hosted on his promises would provide a huge usability gain and would enable the DMA to grow quickly.

Recommendation for Implementation

Granting the DMA access to third party hosted data shall be implemented using the OAUTH scope access mechanism. As different service providers define different scopes, this implementation has to be performed on a per-service basis though will share common code paths like authentication workflow and OAUTH access token handling.

The following image highlights the required infrastructure elements of the DMA Network Access Connector especially perceived from T6.4:

[19] https://oauth.net/2/
[20] https://aws.amazon.com
[21] Google GDrive OAUTH Documentation: https://developers.google.com/drive/v3/web/about-auth, Dropbox OAUTH Documentation: https://www.dropbox.com/developers/reference/oauth-guide, Amazon Web Services OAUTH Documentation: http://docs.aws.amazon.com/cognito/latest/developerguide/authentication-flow.html



The DMA Network Access Connector can be hosted either on the user's premises or centrally by the DMA operator, in which case it will likely deployed in a fault tolerant manner using failover mechanisms. It provides a GUI to define:

- which data sets and services to share
- Smart contracts with SLAs, License, AGBs
- Allowed users / groups / … to access the shared resource
- constraints on the extend the shared resource can be used by a third party

⇒ The details of the GUI part of the DMA Network Access Connector are given in D5.1, "Data Management GUI".

The DMA access component itself consists of these sub-services:

The DMA blockchain. The DMA Blockchain knows about all the registered DMA users, all data sources and services.

- Providing or using services from the DMA requires setting up a user account which will be mapped 1:1 to the account required to participate in the DMA private Ethereum network. Via a centrally provided GUI, the user can promote his registered DMA account to a verified account by e.g. means of authentication with the Austrian Citizen ID card and allows him to consume additional functionality.
- For the blockchain to know all data sources and services, the data source announcement component of DMA will have to create identifiers for each and every data asset (data set or service) the DMA blockchain can reference as the mean to identify the digital resource which shall be protected by smart contracts.

The DMA blockchain will guard access to data as it will grant data access to shared resources only in those cases the defined contract constraints are met by both parties. As of beginning 2017, Ethereum provides a mean to guard resources through the decentralized Swarm-mechanism[22] which plays together with the rest of the Ethereum blockchain and also implements message transfer security.

⇒ The technical details of the DMA blockchain requirements are further specified in D5.1, "Blockchain"

The DMA data indexer. The data indexer will start to harvest defined data sources and services and extract relevant metadata from these resources.

- The extracted metadata will be sent to a central DMA service to be centrally queryable by other users and broker services.
- The actual data itself will only be sent to central storage if the user agrees to centrally

[22] https://swarm-guide.readthedocs.io/en/latest/introduction.html

upload his data.

Once the DMA data ingestor indexes a new resource it will have to create a DMA resource ID to be used by other parties to unambiguously identify a service or data set. These IDs will also be used by the private Ethereum DMA network to define a protected resource and reference them in smart contracts.

⇒ The DMA data and service indexer will have a large overlap to the DMA recommender engine described in D7.1 "Broker and Assessment Technology Specification and Development Road" as recommendations will be based on user provided metadata and data derived metadata as well as with D5.1, section "Data set ingest" described mechanisms to announce the existence of data, metadata and services to the DMA.

The DMA message bus component. The DMA access component will have to communicate with the rest of the DMA network to centrally store indexed metadata and to respond to data and service requests. The DMA access component shall be implemented in a way to auto-discover its required services (central data and metadata storage, DMA blockchain, …). The message bus will transparently encrypt and sign data so that no other component can inspect the data while being transported to the recipient and will sign its messages so that the fact of message delivery can not be repudiated.

The message bus is an integral part of the DMA network access connector, operating on OSI level 4 (transport layer) as well as OSI level 5 (session layer). It is not a service which will be directly exposed to DMA users but is required to support the operation of the DMA backend services, especially the DMA Network Access Connector. The message bus component assures that:

- All content (data assets) sent via the message bus is (optionally) signed: The provider of data assets must be identifiable;
- All content (data assets) sent via the message bus is (optionally) encrypted: Data while on transport must not be inspected by any other party but the intended receiver.

The message bus is compulsory only in the sense that using it will provide the added value to ensure data transport security. If data is going to be exchanged off the DMA communication network infrastructure, these added benefits can not be guaranteed.

The plugin-mechanism. Certain services will not be able to be provided by the DMA or will be provided as a service only to specific users. As such, the DMA access component should provide a plugin-mechanism to extend its functionality. For example the described functionality to access cloud-hosted data could already be implemented as a plugin which would allow it to be extended and developed independently from the DMA core components.

DMA network access infrastructure - The Browser as a general purpose computing environment?

During the last ten years, many technologies have been conceived to bring large-scale computing to the client. The first mostly compatible approach was by means of Java Applets. Unfortunately subtle differences between Java VM implementations as well as Browser sandboxing led to a slow but steady decline of using Java in the web browser and 2016 to the final shutdown of Java Applets [23]. Another .NET-ecosystem-centric solution is (MS-)Silverlight. Silverlight was following a similar approach as Java Applets, but was more UI-centric than being a general computing platform provided by the Java VM. MS-Silverlight has always been encumbered by being tied to the .NET-framework and the platforms .NET supports well. Its primary contender in the rich Internet

[23] https://blogs.oracle.com/java-platform-group/moving-to-a-plugin-free-web

domain was Adobe Flash, which is gradually declining in usage[24] and foreseen to be practically dead by 2018[25].

The reason for the decline of browser-based (mostly) general purpose computing environments is the market and user pressure towards truly standardized solutions, delivered by an open body. HTML5 in conjunction with JavaScript and SVG[26] has already largely replaced Flash even for complex user interface requirements. The browser-based Javascript programming language got many substantial overhauls, and is nowadays capable to serve front-end as well as backend requirements[27]. The WHATWG[28]-led initiative standardises Web workers[29] as a means to extend Javascript based browser computing functionality by concurrency primitives. Web Sockets[30] in conjunction with the overhauled HTTP/2[31] -protocol makes long running web-based client-server connections a naturally supported thing.

The last mile towards morphing the Browser into a general purpose computing environment is to substitute Javascript with a more general approach. This is where WebAssembly[32] comes into play.

WebAssembly or ***wasm*** is a new portable, size- and load-time-efficient bytecode format suitable for compilation to the web. It is currently being designed as an open standard by a W3C Community Group that includes representatives from all major browsers. In its essence it generalizes the principles of having a dedicated web-centric programming language (Javascript) into having a virtual machine which can execute bytecode. Any programming language could target that bytecode format and thus support code execution within the Web browser.

Viability for Data Market Austria

For the DMA it is a declared goal to enable a wide audience to participate in the market in a lean, unobtrusive and safe way, also for less-technologically savvy people. This component will have to perform tasks like extracting metadata from documents declared to be shared on the DMA, announce services to be provided to the DMA, define the conditions according to which data and services are to be shared. This access component will also perform/provide DMA core infrastructure tasks such as provenance tracking on which functionalities such as billing can be build upon. Other imaginable tasks are transparent signing and encryption of messages.

For this functionality to be performed on the user side, some sort of client software needs to be installed and configured. This requires downloading the correct component, interacting with the operating systems installer infrastructure, probably answering administrative and security questions (“Install for all users of the system of just for you?”), likely to deter some users. Many computing network access infrastructure items are delivered in the described way. Examples are the Dropbox client or various Bittorrent clients: While the interaction with these services is happening via the web browser, the browser itself is incapable to provide the required services and infrastructure to participate in these networks.

The required functionality to participate in a decentralized network as the DMA is, cannot be performed entirely using web browser technology: Javascript used to be limiting (weak typing) and is perceived to be somewhat hard to maintain once the code size increases. Additionally the

[24] https://w3techs.com/technologies/details/cp-flash/all/all

[25] http://www.digitaljournal.com/technology/adobe-s-flash-expected-to-be-dead-and-gone-by-2018/article/455949

[26] https://www.w3.org/Graphics/SVG/

[27] https://nodejs.org/en/

[28] https://whatwg.org/

[29] https://developer.mozilla.org/en-US/docs/Web/API/Web_Workers_API/Using_web_workers

[30] https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API

[31] https://tools.ietf.org/html/rfc7540

[32] http://webassembly.org

sandboxing of the Javascript VM does not allow access to local resources in a way to implement the described functionality. Even more, once the browser is closed, all execution is terminated.

Starting as of now, the core infrastructure elements are in place which would allow the implementation of rich web clients which can also access local resources and can perform computationally intense tasks:

- Long-running client-server connections using Websockets and HTTP/2;
- Parallel execution threads using Web Workers;
- Programming language agnostic execution environment using WebAssembly;
- An execution environment which is not strongly tied to the concept of having to provide a User Interface, as provided eg. by "headless Chrome"[33].

Concluding Blueprint for implementation

The building blocks to implement rich web-clients are in place and, given that recent browser technology is being used, are nowadays usable to implement even big and complex back-end logic. However, WebAssembly currently is in the process of being standardized and is not yet ripe to serve as the main element upon which the DMA network user access component could be build upon. Instead, the current approach in widespread use shall be further pursued:

1. Provide a component (The "DMA connector") which acts as the access infrastructure to the DMA network at least in those cases, where data itself will not be uploaded to a central infrastructure due to data protection requirements; As of now, the technology stack for this component has not yet been fully specified, but providing Docker images seems to be a viable solution.
2. Based on the DoW and the requirements elicitation performed by D2.2 "Community-driven Data-Services Ecosystem Requirements" and D5.1 "Data Technology Specification and Development Roadmap", this DMA network access component will have to provide these services:

    a. Data source configuration: Configure the source at which data is available. This also includes granting/configuring access tokens to personal data vaults like Dropbox or Google Drive.
    b. Service announcement;
    c. Local file indexing and metadata extraction;
    d. Data encryption and message signing;
    e. Data access constraint management;
    f. Provenance tracking;

[33] https://developers.google.com/web/updates/2017/04/headless-chrome

6 Conclusion

This document presented the DMA Service Technology Specification and Development Roadmap regarding the ongoing work in WP6 of the DMA project. The purpose of this document was to show that basic technologies and frameworks have been selected and that the necessary technical specifications are in place to support the development, integration, and deployment process.

However, it must be noted that the technical specification is in draft status. The specifications will be continuously adapted according to new requirements coming up, especially during the integration and deployment phases. Also the metadata scheme is going to be updated.

7 References

Armstrong, T.G., Moffat, A., Webber, W., Zobel, J., 2009. EvaluatIR: An Online Tool for Evaluating and Comparing IR Systems, in: Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '09. ACM, New York, NY, USA, pp. 833–833. doi:10.1145/1571941.1572153

Azzopardi, L., Moshfeghi, Y., Halvey, M., Alkhawaldeh, R.S., Balog, K., Di Buccio, E., Ceccarelli, D., Fernández-Luna, J.M., Hull, C., Mannix, J., others, 2016. Lucene4IR: Developing Information Retrieval Evaluation Resources using Lucene. ACM SIGIR Forum 50.

Buchleitner, Christina, and Thomas Rabl. 2017. 'Blockchain und Smart Contracts'. Fachzeitschrift für Wirtschaftsrecht, Blockchain und Smart Contracts - Vom Ende der Institutionen, , no. 2017 (Jänner): 1–14.

Montjoye, Yves-Alexandre de, Laura Radaelli, Vivek Kumar Singh, and Alex 'Sandy' Pentland. 2015. 'Unique in the Shopping Mall: On the Reidentifiability of Credit Card Metadata'. Science 347 (6221): 536–39. doi:10.1126/science.1256297.

Harman, D., 1992. Overview of the First Text REtrieval Conference (TREC-1), in: Proc. of TREC.

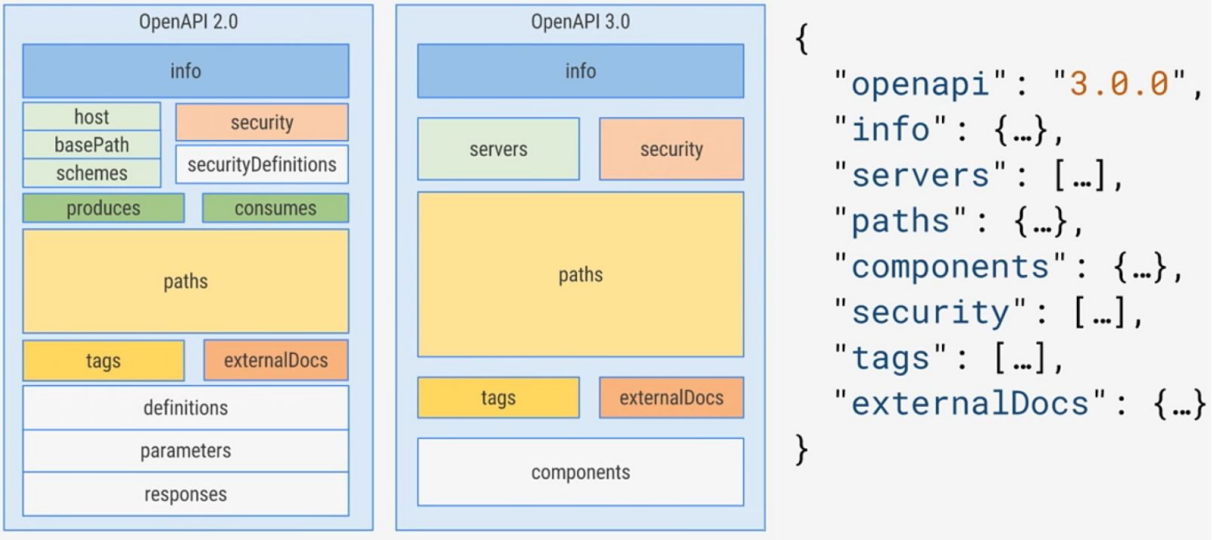
Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In Proceedings of the 2003 conference on Empirical methods in natural language processing (EMNLP '03). Association for Computational Linguistics, Stroudsburg, PA, USA, 216-223. DOI=http://dx.doi.org/10.3115/1119355.1119383

Hasan, Kazi Saidul, and Vincent Ng. "Automatic Keyphrase Extraction: A Survey of the State of the Art." ACL (1). 2014.

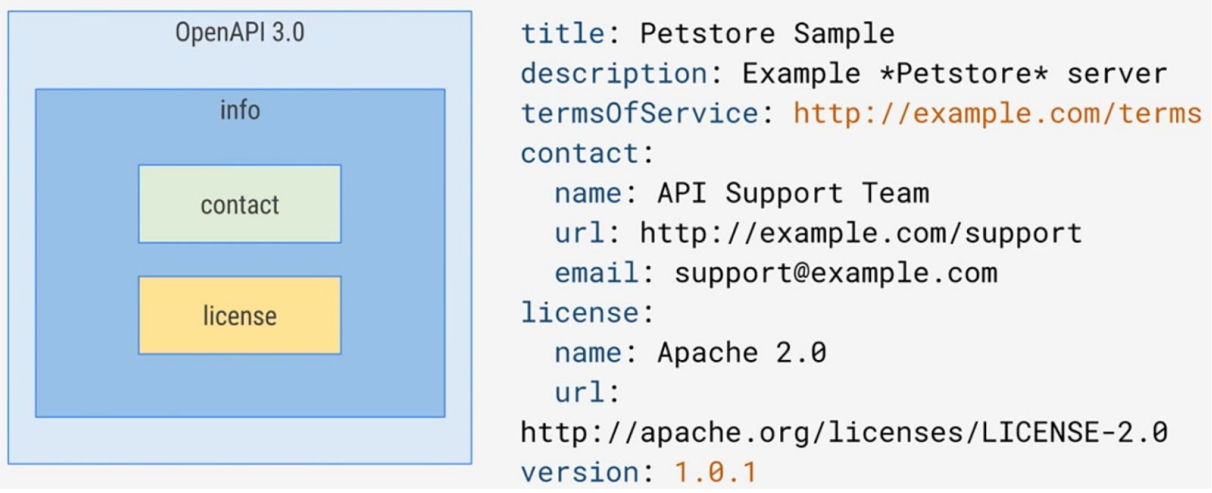
Annex

OpenAPI Specification Comparison

OpenAPI Specification V3.0.0 RC1 information at
https://github.com/OAI/OpenAPI-Specification/blob/OpenAPI.next/README.md

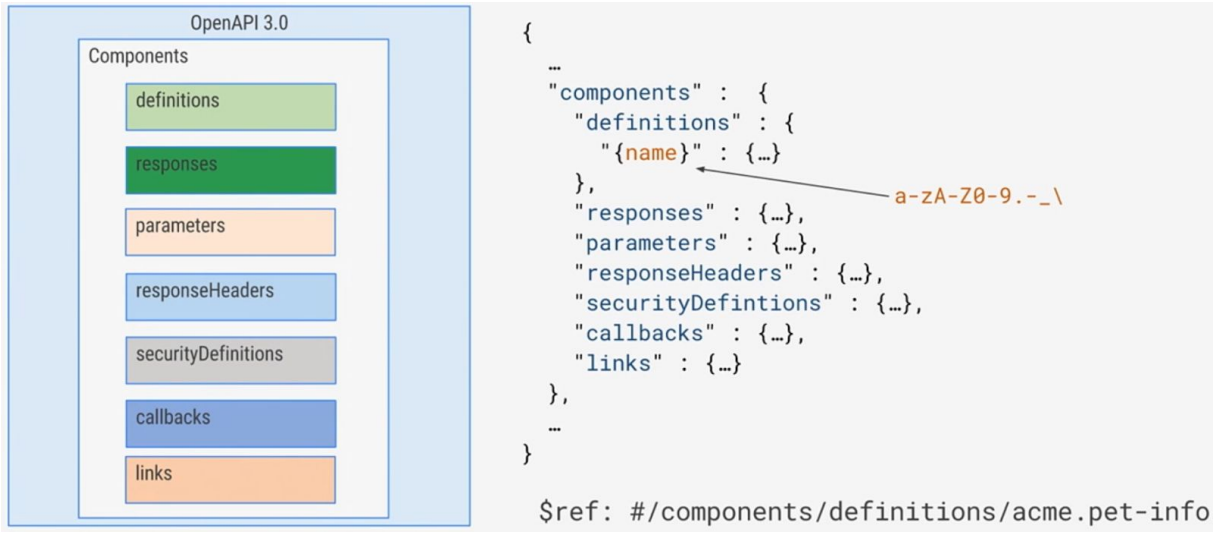


Toplevel differences between OpenAPI version 2 and 3

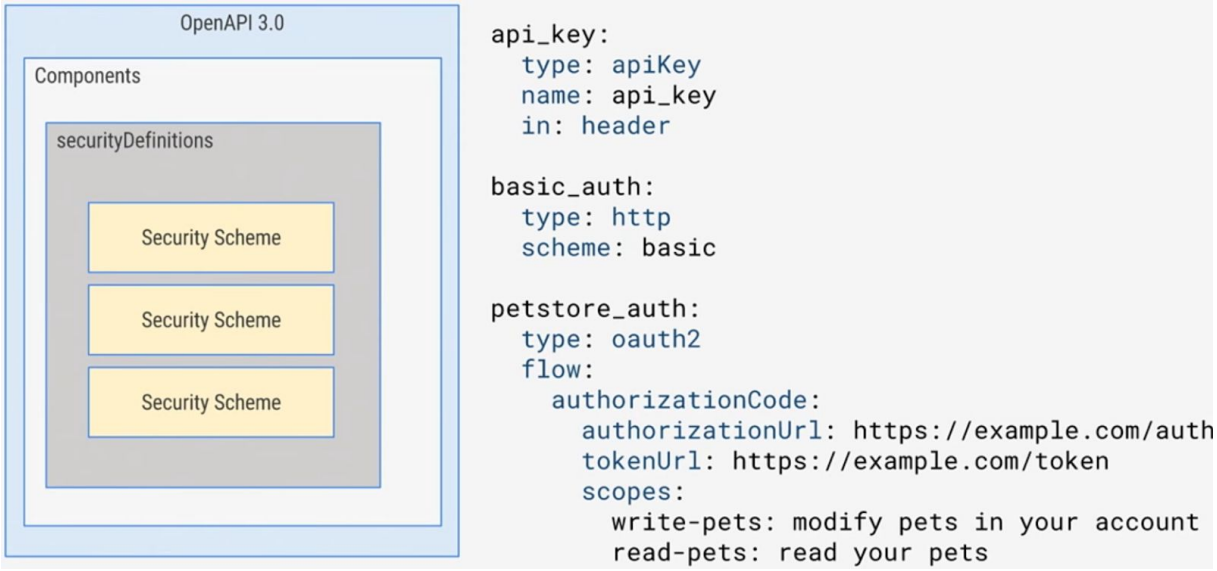


Info description in OpenAPI version 3

Multiple server description in OpenAPI version 3



Reusable components in OpenAPI version 3



Improved authentication scheme in OpenAPI version 3