NGI ONTO CHAIN

Blockchain for the Next Generation Internet

BABELFISH

D2. USE CASE SCENARIOS, DETAILED TECHNICAL SPECIFICATION, IMPLEMENTATION WORK PLAN, AND DEPLOYMENT PLAN (FINAL)

20/01/2022

NGI ONTO CHAIN

Grant Agreement No.: 957338
Call: H2020-ICT-2020-1

Topic: ICT-54-2020
Type of action: RIA

BABELFISH

D2. USE CASE SCENARIOS, DETAILED TECHNICAL SPECIFICATION, IMPLEMENTATION WORK PLAN, AND DEPLOYMENT PLAN (FINAL)

DUE DATE	02/02/2023
SUBMISSION DATE	02/02/2023
TEAM	OwnYourData & Kybernos
VERSION	1.1
AUTHORS	Christoph Fabianek, Sebastian Haas, Jan Lindquist

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

Table of Contents

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

LIST OF FIGURES

LIST OF TABLES

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

ABBREVIATIONS

API	Application Programming Interface
D2A	Domain specific Data Agreement
D3A	Domain specific Data Disclosure Agreement
DID	Decentralised Identifier
DISP	Data Intermediation Service Provider
DPP	Digital Product Passport
DRI	Decentralised Resource Identifier
DTLF	Digital Transport and Logistics Forum
ESG	Environmental Social Governance
FMCG	Fast Moving Consumer Goods
IP	Internet Protocol
JSON	JavaScript Object Notation
JSON-LD	JavaScript Object Notation for Linked Data
NUTS	Nomenclature des unités territoriales statistiques
OYDID	Own Your Decentralised Identifier (did:oyd method)
SCM	Supply Chain Management
SDG	Sustainable Development Goals
SOyA	Semantic Overlay Architecture
TCP	Transmission Control Protocol
TFEU	Treaty on the Functioning of the European Union
TSP	Transport Service Provider
VC	Verifiable Credential
VP	Verifiable Presentation
YAML	Yet Another Markup Language

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

1 Introduction

With the Gateway API component in the ONTOCHAIN environment we will address the challenge of interoperability in a heterogeneous environment. Interoperability by itself provides overall system benefits at different, distinct dimensions and a common approach[1] is to distinguish between the following aspects: technical (connectivity), semantic (informational), and organisational (governance, business models etc.)

- On the technical level connectivity, syntactics, and protocols for data exchange (e.g., APIs) and data storage underpin basic integration;
- the semantic level requires harmonised information with shared data models and mutually agreed consent; and
- the organisational level (usually only addressed in more mature ecosystems) encompasses shared objectives and policies between organisations.

In this document we lay out our planned design for the Gateway API. Based on identified requirements in Deliverable 1 to answer the question of WHAT to build we focus in Deliverable 2 on the HOW. It includes the final user stories, a description of all system components together with the system architecture, and an implementation and deployment plan.

This document serves as a comprehensive guide for the implementation and development of the project, and it will be followed by Deliverable D3 "Software Implementation" that will be available at the end of May 2023.

[1] 'Architecture constraints for Interoperability and composability in a smart grid', Power and Energy Society General Meeting, 2010 IEEE. https://www.researchgate.net/publication/224178883_Architecture_constraints_for_Interoperability_and_composability_in_a_smart_grid

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

2 User Stories and Use Case Analysis (final)

This chapter provides a list of user stories describing the core functionality of the components to be developed in the course of the NGI ONTOCHAIN funded Babelfish project. User stories are grouped based on the Gateway API functionalities

- Service Discovery,
- Organisation and User Accounts,
- Storage Service Results/Output,

and the three main use cases to demonstrate the overall functionality:

- Data Validation Use Case,
- Interoperability Use Case, and
- Supply Chain Management Use Case.

2.1 Service Discovery

As a service developer I want to be able to register a service in a service catalogue so that others can easily discover it.

As a service developer I want to be able to update and delete existing entries in a service catalogue so that I can keep everything up-to-date.

As a service developer I need to authenticate myself against the Gateway API using OAuth2 client credential flow to register, update, or delete entries.

As a developer I want to be able to query the service catalogue so that I'm able to discover available services.

As operator of the service catalogue I want to be able to configure one of the available storage providers to persist the service catalogue.

As an anonymous user I want to be able to retrieve the current entries in the service catalogue without authorization.

2.2 Organisation and User Accounts

As an organisation I want to be able to create an entry in a registry so that I can access services with this identity.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

As an organisation I want to be able to update and delete my organisation entry in the registry so that I can keep it up-to-date.

As an organisation I want to be able to retrieve my stored data so that I can verify the data.

As a user I want to be able to create an entry in a registry and assign it to an organisation so that I can access services with this identity.

As a user I want to be able to update and delete my user entry in the registry so that I can keep it up-to-date.

As a user I want to be able to retrieve my stored data (incl. configured chain, address, balance) so that I can verify the data and have the latest version.

As operator of the organisation and user registry I want to be able to configure one of the available storage providers to persist the registry.

As a user I need to authenticate myself against the Gateway API using OAuth2 client credential flow to access organisation and user accounts.

2.3 Storage Service Results/Output

As a developer I want to be able to create a collection so that I can store objects in the collection.
Note: a collection can be interpreted as a folder in a file system and groups together objects; many storage mechanisms provide this kind of grouping sometimes also referred to as repository or bucket

As a developer I want to be able to update and delete collection entries so that I can keep them up-to-date.

As a developer I want to be able to retrieve all information of a collection and a list of all accessible collections so that I can verify the latest state.

As a developer I want to be able to create objects in a collection so that I can have a flexible storage mechanism.

As a developer I want to be able to update and delete objects in a collection so that I can keep them up-to-date.

As a developer I want to be able to retrieve the details about an object so that I can work with the data and have the latest version.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

As a developer I want to be able to specify the backend-storage to be used for managing collections and objects. Available storage mechanisms for on-chain data will be Convex and ONTOCHAIN's ethereum based blockchain (Bellecour) and for off-chain data Semantic Container and Amazon S3 can be used.

As a user I need to authenticate myself against the Gateway API using OAuth2 client credential flow to access storage services.

2.4 Validation Use Case

As a knowledge worker I want to be able to describe data models in a simple format (e.g., YAML) that include both general and self-defined data types.

As a knowledge worker I want to manage different versions of a data model and share data models online for collaborative authoring.

As a knowledge worker I want to be able to define terms of constraints on the content, structure and meaning of a graph beyond data types.

As a knowledge worker I want to be able to transform a description of a data model into an RDF graph that complies with the semantic web standards (i.e., RDFS/OWL) for data model representations.

As a service/application I want to be able to validate a data record against a data model representation.

As an end user I want to be able to validate a data record against a data model representation on a GUI and retrieve human-readable explanations for violations in the defined constraints.

2.5 Interoperability Use Case

As a service/application I want to be able to document my API endpoints, used data model, and applicable usage policies in a Service Description so that this information can be used for integration with other services/applications.

As a service/application I want to be able to connect to other services/applications independent of the respective API endpoints so that I can exchange data without technical integration limits.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

As a service/application I want to be able to connect to other services/applications independent of the respective data models so that I can exchange data without semantic integration limits.

As a service/application I want to be able to have usage policies automatically validated upon data exchange so that I can exchange data with full governance conformance.

As services/applications that exchange data we want to be able to have this data exchange documented in a domain specific data (disclosure) agreement so that full governance conformance is unambiguously documented.
note: data agreements build on the work from ONTOCHAIN call #2 funded project PS-SDA

2.6 Supply Chain Management Use Case

For two strategically selected supply chain management purposes we specify and parametrise the use cases of service discovery (section 2.1), organisation and user account (section 2.2), storage service (section 2.3) to enable and demonstrate supply chain specific user stories. Together with both other main use cases i.e. validation (section 2.4) and interoperability (section 2.5) we are aiming at a set of supply chain functionality which we design to create a Digital Product Passport scheme as follows:

As a supply chain party I want to be able to share and receive relevant data so that this data can be used by other supply chain parties for planning, optimization and documentation purposes.

As a transport service provider I want to be able to share and receive relevant data so that this data can be used by other supply chain parties for planning, optimization and documentation purposes.

As a supply chain party I want to be able to share and receive relevant data so that this data can be used by other supply chain parties for planning, optimization and documentation purposes.

As a market maker I want to be able to provide structured food product information so that this data can be used by customers for market and product differentiation.

As a regulator I want to be able to design, co-create and mint purpose-driven data circles within the supply chain data space so that the shared data can be (re-)used for data-driven circular economy policy schemes like the Digital Product Passport and supply chain management regulations in general.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

As a region I want to be able to document and shape public procurement practices to be able to reach net zero targets for the food and transport sector.

As a data intermediation service provider I want to be able to structure and govern a product data space around the regulator-driven design artefact Digital Product Passport so that this structured and incentivised data exchange can be used by consumers, regulators, investors and insurance companies for their purposes.
note: the Digital Product Passport combines information from the complete supply chain and also includes Verifiable Presentations, i.e., attestations from third parties

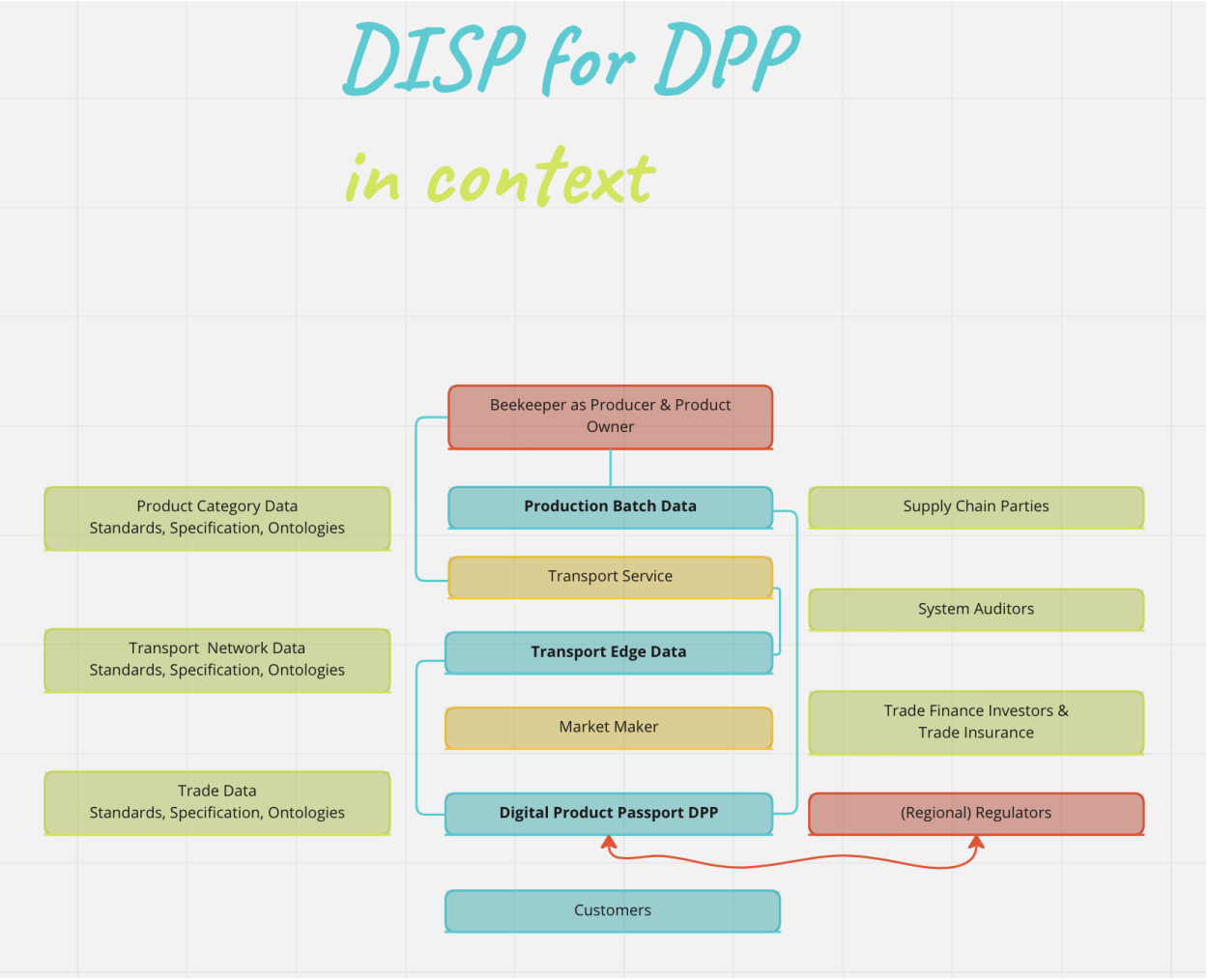


Figure 2.1: Supply Chain Management Context

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

3 Software Design and Analysis, Component Specification (final)

This chapter provides the architecture diagram and detailed technical specification of the components developed in the course of the project.

3.1 Software Modules

This section describes what will be developed in the context of ONTOCHAIN OC3 in terms of technical components.

3.1.1 Decentralised Identifiers: `did:oyd` Methode

Non-blockchain based DIDs are a type of Decentralised Identifiers that do not rely on blockchain technology to function. There are different ways to implement non-blockchain DIDs, but one common approach is to use a distributed ledger or a distributed hash table (DHT) to store and manage the identifiers.

The main difference between blockchain-based and non-blockchain based DIDs is that the latter do not rely on the consensus mechanism of the blockchain to ensure their integrity and availability. Instead, they use other methods such as digital signatures, certificate authorities, or trusted third parties to establish trust and ensure that the identifiers are authentic and accurate.

The did:oyd method[2] is an example of a non-blockchain based DID methode and provides a self-sustained environment for managing digital identifiers: it links the identifier cryptographically to the DID Document and through also cryptographically linked provenance information in a public log it ensures resolving to the latest valid version of the DID Document.

Information about DIDs, DID Documents and associated logs are stored in an OYDID repository. This repository is a centralised (online) storage but the `clone` operation allows duplicating any information to another repository and in this way provides decentralisation. An OYDID repository can be hosted by anyone and it is up to the owner of DID to select a trusted provider. Through the decentralised nature of the did:oyd method it is at any time possible to move to other repositories.

[2] W3C conform DID method specification: https://ownyourdata.github.io/oydid/

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

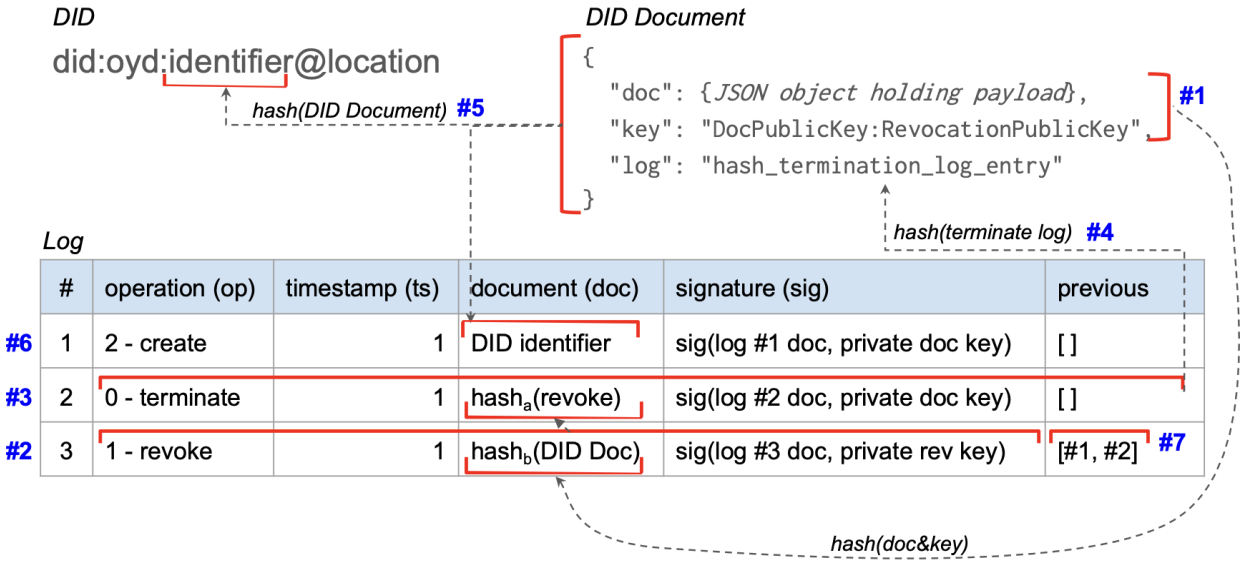


Figure 3.1: Artefacts of the did:oyd Method

In the course of the ONTOCHAIN project the did:oyd method will be extended with the following functionality to act as identifier within the Gateway API and towards services:

- **DID Delegation**
  Delegation is a mechanism that allows an entity to delegate the management of its DID to another entity. This can be useful in situations where an entity is unable or unwilling to manage its own DID. The delegator (the entity that wants to delegate control of its DID) gives the delegate (the entity that will manage the DID on behalf of the delegator) permission to perform certain actions on the DID.
  Delegation for did:oyd will allow an actor to give temporary access to others and specifically in the context of interaction between services this is a necessary functionality.

- **Verifiable Credentials and Verifiable Presentations**
  Verifiable Credentials (VCs) provide a secure, efficient, and private way to verify claims about individuals and entities, and can help to reduce the need for intermediaries, increase trust and accessibility, and automate verification processes.
  Verifiable Presentations (VPs) are a way to present one or more Verifiable Credentials to a verifier in a secure and privacy-preserving way. VPs allow an

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

individual to selectively share only the information that is necessary for a specific transaction or interaction, rather than sharing all of their personal information.
VCs and VPs for did:oyd will allow actors to seamlessly attest information, especially to be used for data agreements.

- **DID Test Suite compliance**
  The DID test suite compliance helps to ensure that the did:oyd method implementation is of high quality, compatible with other systems or applications, and compliant with the relevant standards. It also makes it easier to integrate OYDID into other systems or applications and speed up the overall development process.
  The did:oyd method is a newcomer in a series of established DID methods. By ensuring full compliance of did:oyd with existing standards and being a pioneer in developing self-sovereign data management methods, OYDID will provide a fresh breath for applications in the ONTOCHAIN environment.

3.1.2 Semantic Overlay Architecture

The Semantic Overlay Architecture (SOyA) is a lightweight, semantic-web based approach to describe data structures in simple terminology[3]. This description includes groups of data records with the same attributes, references between data records, and additional information in the form of overlays for these data structures.

At the core of the SOyA approach is the SOyA structure, a YAML-based data model for describing graph data, which consists of one or more `soya:Base`, that represent RDF classes and their properties, and zero or more `soya:Overlay`, that provides additional information and context to `soya:Base` as well as processing definitions. Furthermore, to support developers in conducting the most common data processing for graph data, we have defined a number of predefined `soya:Overlay` e.g., such as `soya:AnnotationOverlay` for data model description in human readable terms and `soya:ValidationOverlay` for constraint checking - see Figure 3.2.

It is important to note that SOyA has the same flexibility as RDF (Resource Description Framework) to describe data structures, i.e., any kind of data or documents can be

[3] W3C conform specification: https://ownyourdata.github.io/soya/

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

described. With SOyA it is possible to register any current and future data models handled by ONTOCHAIN services and applications.

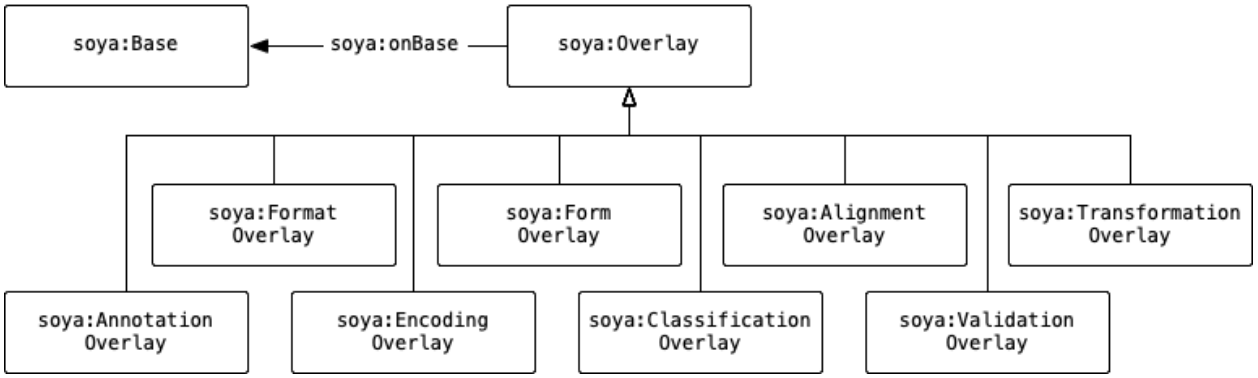


Figure 3.2: SOyA Ontology Concepts and Relations

The following tools are available to work with SOyA structures:

i. **`soya-cli`** is a JavaScript-based command line interface (CLI) which allows developers to easily interact and work with SOyA by providing functionality to deal with data modelling and various SOyA overlays
ii. **`soya-repository`** for managing SOyA structures, allowing data model creation, versioning, and storage, among others, similar to GitLab for source code management
iii. **`soya-js`** provides interfaces in JavaScript for handling SOyA structures and interacting with SOyA repositories

In the scope of the ONTOCHAIN Gateway API, SOyA is the underlying mechanism for all data management tasks. In the **Validation Use Case** (section 2.4) the Validation Overlay is used with an open source library[4] to verify conformance of a data set (i.e., an instance of a SOyA structure). This use case will be demonstrated through a specific DID Lint service that is available online with a web frontend, as well as through APIs - this service will also be a documented example for the use of a Service Description in the Service Catalogue. Further application areas are a general linting service for Usage Policies and Data Agreements used in the project.

In the **Interoperability Use Case** (section 2.5) Alignment and Transformation Overlays will be used to transform compatible datasets between different representations. An

[4] https://github.com/zazuko/rdf-validate-shacl

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

Alignment Overlay references classes and properties within a SOyA structure to existing ontologies using class and property subsumption relations. Through transitive mapping between SOyA structures a Translation Overlay can be derived that transforms instances between different ontologies.

Note: A transitive mapping is a mathematical concept used in set theory and abstract algebra. It refers to a function or a mapping from one set to another set, where if an element from the first set is mapped to an element in the second set, and that second element is mapped to a third element in the second set, then the original element must also be mapped to that third element. In other words, a transitive mapping preserves the relationship between elements in the first set and elements in the second set, such that if element A maps to element B, and element B maps to element C, then element A must also map to element C. This relationship is known as transitivity.

The **Supply Chain Management Use Case** (section 2.6) finally brings together validation and transformation using SOyA to demonstrate data exchanges along the value chain for Honey around Vienna as a fast moving consumer good (FMCG). Data exchange is a crucial aspect of supply chain management because it enables the smooth flow of information between different organisations and systems. This in turn allows better coordination, communication, automation and performance tracking, thus improving efficiency, reducing costs and increasing customer satisfaction.

3.1.3 Domain Specific Data Agreements

The exchange of data has two main parties, the *data consumer* of the data or sometimes called data user, and the *data source*, the one who generates the data. In order to enable the sharing of the data, metadata needs to be added that helps set the *usage policies* of use of the data.

The following diagram depicts how organisation A and B communicate through the intermediary. The data source, organisation A, informs the intermediary of the usage policy to access the data as the data consumer, organisation B. Neither need to communicate directly and the intermediary has all the information to match and validate the usage policy in order to create a *Domain specific Data Disclosure Agreement (D3A)* - more in next section.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

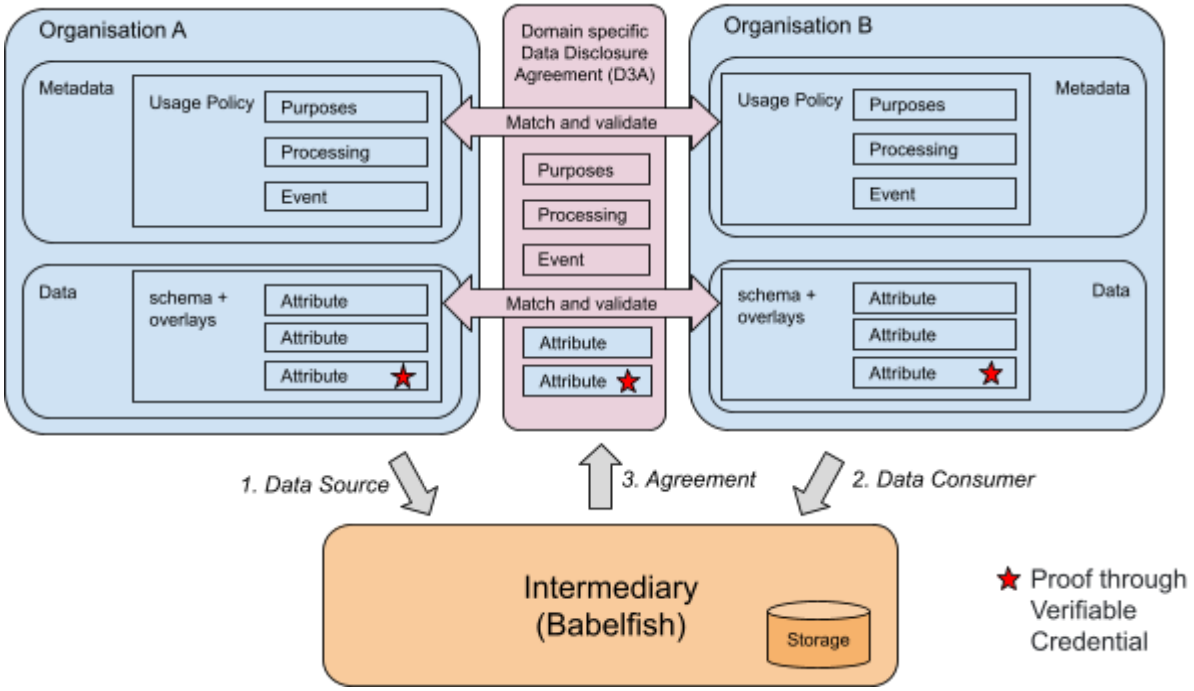


Figure 3.3: Data Agreements

Babelfish is hosted as a central service in the ONTOCHAIN environment but still follows decentralised design principles by applying content-based addressing for all data and metadata stored on it. In case of a malicious hoster it will always be possible for the community to select another intermediary and migrate any content without any loss of information or functionality.

3.1.3.1 Intermediary Data Exchange Overview

The agreements that are setup between parties need to be distinguished based on the type of relationship. Individuals and organisations have similarities when setting up these agreements. Both have data relating to their activity. Individuals may have health data or activity data from smart watches and organisations generate data relating to production or a service they provide. The challenge for both is how to perform data portability. The intermediary has a central role to facilitate the sharing of data and acts on behalf of the organisation. Babelfish can act as such an intermediary and agreements will be the basis for creating the metadata that facilitates the exchange of the data.

The following diagram compares data exchange for individuals which iGrant.io in previous ONTOCHAIN call #2 developed a Data Disclosure Agreement (PS-SDA) with

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

the Babelfish data exchange developed in this project for any domain specific application. PS-SDA is considered as one of these domains, namely personal data domain.

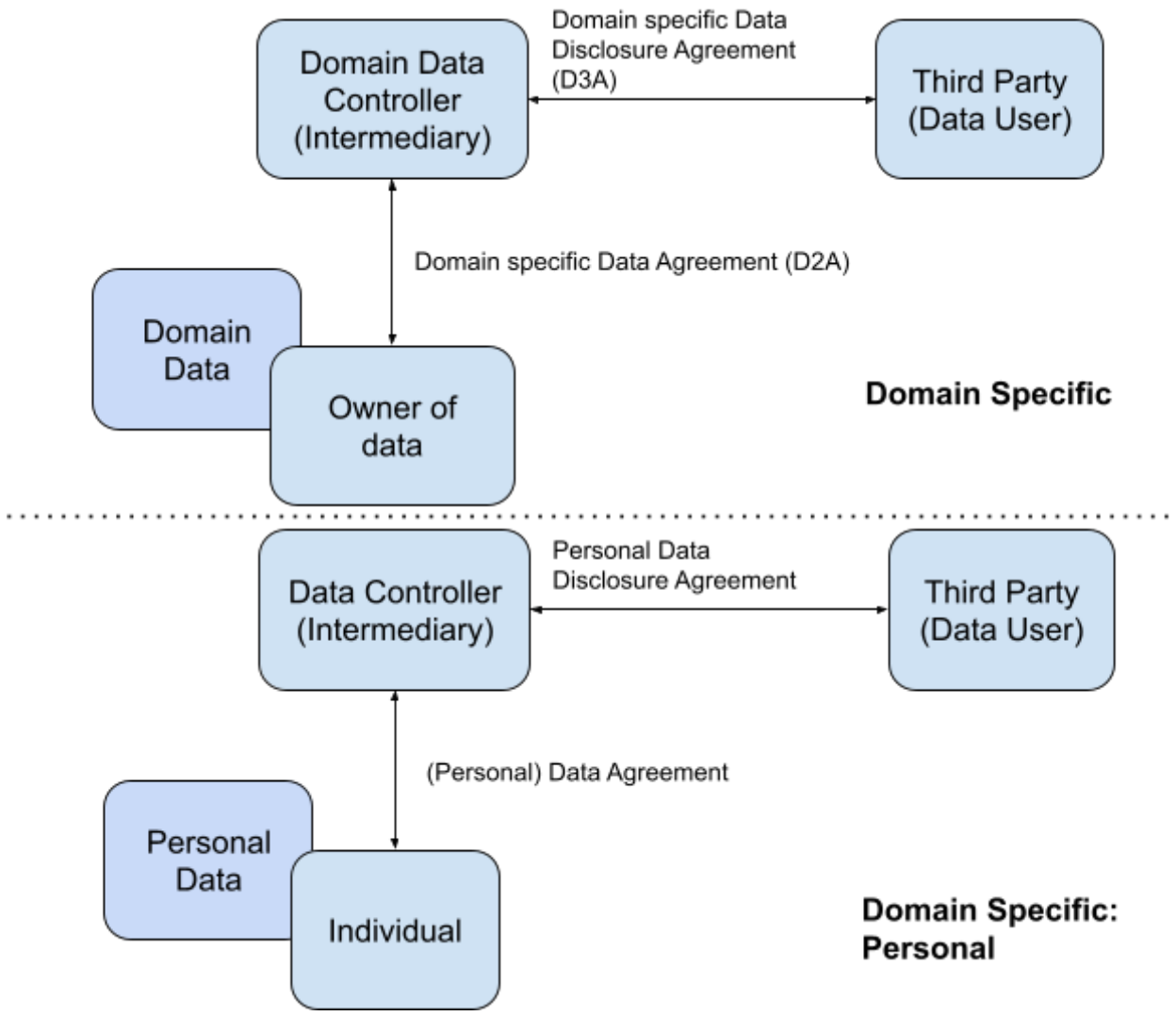


Figure 3.4: Domain Specific Data Agreements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

The relation between the two is similar and the following table describes the matching of the type of agreements.

Relation	Individual	Organisation
owner of data with intermediary	Personal Data Agreement (DA)	Domain Data Agreement (D2A)
intermediary with third party	Data Disclosure Agreement (DDA)	Domain specific Data Disclosure Agreement (D3A)

Table 3.1: Data Agreement Types Comparison

Note: due to the size of a Data Agreement it is always stored off-chain on the configured Storage Provider; nevertheless, the hash-value of the Data Agreement is stored on-chain to guarantee immutability and timeliness

3.1.3.2 Agreement Schema

This section describes the agreement schema and matches the terms associated with individuals data sharing and with organisation data sharing in the context of supply chain.

The following table gives an overview of the schema and is based on the Data Disclosure Agreement (Appendix A) from iGrant.io. The schema is based on the Kantara Consent Notice [3], as well as ongoing ISO standardisation under ISO project 27560, Consent record information structure.

Purpose(s)	• Purpose - describe (highest risk first) • Lawful basis • Domain Data Information, DDI (categories, sensitivity or optional)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

Processing	• Processing method (copy, combined or automated decision making) • Storage location • Retention (how long data kept) • Service name applicable for purpose • Jurisdiction data stored • Third parties • Exercise privacy rights (link to withdraw) • Code of conduct • Assessment performed • Privacy policy url
Event	• Event time • Event type (implicit, explicit or regular) • Event state (ex. notice, consent or termination) • Validity duration (how long consent valid) • Signing Data Controller • Signing individual

Table 3.2: Data Agreement Schema Overview

The following table is a mapping of the agreement template between individual and organisation. Purpose of the mapping is to identify if anything is missing or if field names should be different.

Individual		**Organisation (domain specific)**		**Comments**
Field	**Type**	**Field**	**Type**	
Purpose(s)		Purpose(s)		
- Purpose description	string	- Purpose description	string	
- Purpose type	dpv:Purpose	- Purpose type	sc:Purpose	
- lawful basis	string	- lawful basis	string	
- Data information	list	- Data information	list	refer to supply chain data information section

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

Individual		Organisation (domain specific)		Comments
Field	**Type**	**Field**	**Type**	
Processing		Processing		
- processing method	dpv:Processing	- processing method	string	
- retention period	ISO 8201 duration	- retention period	ISO 8201	
- geographic restrictions	string	- restrictions		any restrictions or conditions for sharing data
- third party	list	- third party	list	refer to organisation
- storage location	string	- storage location	string	
- services	list	- services	list	
- jurisdiction	string	- jurisdiction	string	
- privacy rights	reference	- exercise rights	reference	type is a reference pointing to information to exercise rights like withdrawal
- code of conduct	object	- code of conduct	object	content is not defined; can simply be a URL
- impact assessment	object	- impact assessment	object	content is not defined; can simply be a URL
- privacy policy	URL	- policy	URL	terms of use
Event		Event		

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

Individual		Organisation (domain specific)		Comments
Field	**Type**	**Field**	**Type**	
- event time	ISO 8201	- event time	ISO 8201	
- event type	string	- event type	string	
- event state	string	- event state	string	
- duration	ISO 8201 duration	- duration	ISO 8201 duration	only if binding to an agreement like consent and how long it is valid.
- entity id	DID	- entity id	DID	the one that triggered the event
Agreement signers		Agreement signers		
- individual ID	DID	organisation id	DID	
- data controllers	list	organisation id	DID	

Table 3.3: Data Agreement Template Mapping between Individuals and Organisations

Note: a Data Agreement can be initiated by any party participating in a data exchange but is usually requested by the receiving actor to document legal compliance; it is also up to the receiver to accept data only with a suitable Data Agreement

3.1.3.3 Domain specific Data Agreement And Domain specific Data Disclosure Agreement

This section explains the supply chain related agreements and which fields are mandatory. Literally all mandatory fields match.

Field	D2A Mandatory	D3A Mandatory	Description
- Purpose description	M	M	Purpose description for processing data

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

Field	D2A Mandatory	D3A Mandatory	Description
- Purpose type	M	M	Purpose category/type using ontology language
- lawful basis			GDPR requirement to indicate reason for processing data: consent, legal obligation, contract, vital interest, public task or legitimate interest
- Data information	M	M	List of data attributes for this specific purpose
- processing method			Processing method using ontology language
- retention period	M	M	how long data is stored before being deleted; mainly a privacy requirement and data minimisation
- restrictions			Restrictions set for processing of data like geographic restrictions, cannot leave the EU
- third party	M	M	List of third parties that receive a copy of the data
- storage location			Location data is stored after processing
- services			Service or services the purpose applies to
- jurisdiction	M	M	Jurisdiction the controller is registered
- exercise rights	(M)	(M)	Reference to information on how to exercise rights. Mainly used for privacy and ability to withdraw
- code of conduct			Indicate any code of conduct followed by data controller
- impact assessment			Indicate any assessment and identified risks with processing of the data
- policy	(M)	(M)	Mainly for privacy policy URL

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

Field	D2A Mandatory	D3A Mandatory	Description
- event time	M	M	Following fields used to register events in the lifecycle of an agreement; time the event took place
- event type	M	M	Was event implicit or explicit (ex consent given) or regular
- event state	M	M	State in lifecycle; for example for privacy, notice, consent and termination; for domain specific the state has to be specified
- duration	M	M	If event is binding like a consent how long is valid be it expires
- entity id	M	M	Indicate which entity issued event

( ) mean it is mandatory for privacy domain and not necessary for other domains like supply chain

Table 3.4: Data Agreement Schema for Supply Chains

Note: Data Agreements and Smart Contracts share some properties but Data Agreements have an actual wider scope and also require a dedicated negotiation phase between participating parties that could be cumbersome to implement to the full extend on a distributed ledger; upon the actual data exchange the content of a Data Agreement could be documented as a Smart Contract

3.1.3.4 Domain Specific Data Information

The data that is collected requires a data model to validate data sources and also match with data consumers. A template is created for evaluating each attribute generated by an organisation. This is the information collected on each attribute:

1. Format: used to perform validation and translation
2. Type: data category type
3. Sensitivity: any restrictions to access attribute and are Competitively Sensitive Information (CSI)
4. Proof with VC: indicate if attribute has a proof in the form of a verifiable credential (VC)

This is an example of the template to be filled out by each organisation.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

Attribute	Format	Type	Sensitivity	Proof with VC	Description

Table 3.5: Template for Supply Chain Data Information

Organisation

Attribute	Format	Type	Sensitivity	Proof with VC	Description
id	DID	identifier	yes		Reference is internal to intermediary
organisation id	string (1)	identifier	no	yes	Organisation registration number based on GS1 ID or national ID (ex Austrian eID)
address	string	tracking	no	no	
membership	string		yes	yes	sector associations

Note (1) If a specific schema is used it is indicated in type. Information is used to validate transfer of data.

Table 3.6: Organisation Data Information

Beekeeper / Producer

Attribute	Format	Type	Sensitivity	Proof with VC	Description
membership			yes	yes	e.g., chamber of agriculture
id	DID	identifier	yes		Reference is internal to intermediary
bee keeper id	string	identifier	yes		external reference format not set
produced by	DID	identifier	no		ID is reference to organisation

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

Attribute	Format	Type	Sensitivity	Proof with VC	Description
					record
production batch volume	float	characteristic	yes		
production batch date	ISO8201	characteristic	yes		
bee collection area	string (coordinates)	characteristic	no		Term used is “wanderkarte”.

Table 3.7: Beekeeper/Producer Data Information

Transport Service Provider (TSP)

Attribute	Format	Type	Sensitivity	Proof with VC	Description
membership	string	characteristic	yes	yes	transport system roles (eg Infrastructure Manager or Railway Undertaking), transport networks, TEN-T corridors
id	DID	identifier	yes		Reference is internal to intermediary
transported company	DID	identifier	no		ID is reference to organisation record
vehicle priority	integer	characteristic	no		Note 1
vehicle label	string	characteristic	no		Designation of vehicle or vehicles used
optimization time	ISO8201	characteristic	yes		Note 2
optimization	string	characte	yes		Note 3

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 957338

algorithm		ristic			
optimization metadata	object	characte ristic	yes		Metadata required for optimization

Table 3.8: Transport Service Provider Data Information

Note 1: Vehicles should be annotated with a priority to rank their importance with regards to their respective keeping time constraints, i.e. when valid working hours (from earliestDepartureTime until latestArrivalTime) are exceeded the penalty for not finishing the tour on time is multiplied by the factor priority. In this way vehicles with a higher number are more likely to have tours that are still within their specified working hours. For instance, the optimization would evaluate the meeting of the time constraints of a vehicle with the priority of 50 as high as meeting the time constraints of five vehicles with a priority of 10. That also applies when secondsToPenaltyRatioForOutOfWorkingHours is set.

Note 2: optimizationTime specifies the desired maximum run time in seconds this request can use for the optimization. The actual run time can be lower if a solution is found quicker, or exceeded when, for example other parts of the routing request, e.g. routing, parsing, serialising of the request response, take longer.

Note 3: optimizationAlgorithm specifies which optimization algorithm is to be used. The following options are available:

- NO_OPTIMIZATION – no optimization is carried out, i.e. the algorithm will generate a tour for one vehicle per depot without carrying out routing and without trying to fulfil the constraints
- GREEDY_TSP – Greedy Algorithm: A route is created, by always travelling to the closest order from the current point of view until all orders have been served. This does not consider order or vehicle priorities.
- BRUTE_FORCE_TSP – All delivery order and transport combinations are generated and the one with the least consumed travel time is selected. This does not consider order or vehicle priorities.
- CONSTRAINT_SATISFACTION – Powerful Algorithm based on constraint solving that tries to find an optimal solution that meets all the soft/hard constraints and optimises for the travel time that is used for the routes. This algorithm should preferably be used.

Market Maker / Shop / Store

Attribute	Format	Type	Sensitivity	Proof with VC	Description
membership	string	characte ristic	yes	yes	trade groups (eg “Wiener Detailmärkte" as regulated by the “Wiener Marktamt”

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 957338

Attribute	Format	Type	Sensitivity	Proof with VC	Description
id	DID	identifier	yes	yes	Reference is internal to intermediary
Market / shop credentials like	string	characteristic	no	no	address or polygon for market space
Transport Label basic (1)	string	D3A attribute	yes	yes	gr:DeliveryMethod
Transport Label advanced (1)	string	D3A attribute	yes	yes	gr:DeliveryChargeSpecification
Designation of Origin basic (1)	string	D3A attribute	yes	yes	gr:hasGlobalLocationNumber
Designation of Origin (1) advanced	string	D3A attribute	yes	yes	Polygon for production site

Note - (1) Data consumption by Market Maker

Table 3.9: Market Maker Data Information

3.1.4 Semantic Container

A Semantic Container is a Docker-based data storage used to store and manage information in a semantically meaningful way. It provides a way to represent data as a collection of entities and relationships, making it easier to understand, process and query. This type of storage can be used in the context of semantic web and linked data, where the focus is on using structured data to enable more intelligent and automated processing of information on the web.

In the ONTOCHAIN environment Semantic Container can be used as a low-cost local storage provider. It natively uses DIDs and DRIs (decentralised resource identifiers) for addressing data and has basic functions for generating provenance information and validating compliant usage policies between data controller and data subject. The main benefits of using a local storage provider (in contrast to an online storage

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

provider like Amazon S3) are speed (data can be accessed and processed much faster), offline availability (access even when the device is not connected to the internet), security (using the device's security mechanisms and making it less susceptible to security breaches than online storage), and cost (does not incur the costs of server hosting and network bandwidth).

In the course of the ONTOCHAIN project the existing Semantic Container implementation will be extended with the following functionalities:

- **Storage Backend for Gateway API**
  extend Semantic Containers to cover all functionalities required by the Storage Provider APIs (section 4.2.3)
- **W3C conform specification**
  write and publish a complete specification for interacting with Semantic Container

3.1.5 Babelfish Component

The ONTOCHAIN Gateway API is the entry point for developers and users of ONTOCHAIN services and is composed of three main modules. This section gives a short description for each of the modules, references the API endpoints described in chapter 4, and provides the planned documentation and use-cases.

- **Service Discovery:** search deployed ONTOCHAIN services with structured queries and access them directly
  APIs: section 4.2.1
  tutorial: registering the validation service

- **Accounts Management:** functions related to organisation and user accounts and access rights
  APIs: section 4.2.2
  documentation: curl statements and Postman collection to interact with APIs

- **Data Storage:** high-level interfaces that are common to all of the supported backend storage services: Semantic Container (off-chain data), Amazon S3 (off-chain data), Convex (on-chain data), ONTOCHAIN Bellecour (on-chain data)
  API: section 4.23
  tutorial: supply chain management use case

The relevant data structures used in the Babelfish component are described below.

Service Description
the service description is a JSON object with three required keys:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

Service Description Structure

```
{
  "interface": { Open API Specification },
  "data": " SOyA structure ",
  "governance": { Usage Policy }
}
```

Details:

- `service-id`: unique id of the service for referencing purpose provided by the system
- `interface`: describes the interface of the service (specifically API endpoints) and general aspects of the entity using the Open API Specification v3 (also known as Swagger documentation)
- `data`: describes the expected data structure using SOyA (Semantic Overlay Architecture) structures; if the data structure does not exist, this entry shall provide `null`; it is important to note that the content of this field is optional and services in ONTOCHAIN are not required to use SOYA -however, if those services decide to use SOyA, they benefit from the additional functionality in the integration helper
- `governance`: describes the usage policy based on the structure from the Data Privacy Vocabulary[5] - this information is used as input for Data Agreements (section 3.1.3) and relevant attributes are described there

Service Description Example

```
{
  "service-id": 1,
  "interface": {
    "info": { "title": "DID Lint" },
    "servers": [{"url": "https://didlint.ownyourdata.eu"}],
    "party": "data_consumer",
    "paths": {
      "/api/validate": {
        "post": {
          "requestBody": {
            "content": {
              "application/json": {
```

[5] https://w3c.github.io/dpv/dpv/

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

```
                "schema": {}
              }
            }
          }
        }
      }
    }
  },
  "data": "zQmc76XfAkKxjFhHUANsq2yLxvt1FNkwHULChENskd9PJ9T",
  "governance": {
    "sc:hasProcessing": ["sc:Use"],
    "sc:hasPurpose": "sc:Purpose",
    "sc:hasExpiryTime": "6 months"
  }
}
```

Organisation Record
an organisation record is a JSON object with a minimum of the following attributes:

Organisation Record Structure

```
{
  "name": " organisation name ",
  "description": " information about organisation ",
  "address": " street number, zip code, city, country ",
  "governance": { Usage Policy }
}
```

Details:

- `organization-id`: unique id of the organisation for referencing purpose provided by the system
- `name, description, address:` mandatory information about the organisation
- `governance`: optional field to provide a default governance for data provided by this organisation; if governance is provided in the service description the information there prevails

Organisation Record Example

```
{
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

```
  "organization-id": 1,
  "name": "ACME Inc.",
  "description": "generic business in instructional settings",
  "address": "1000 Main St, 37917, Knoxville, US",
  "governance": {
    "sc:hasProcessing": "sc:Processing",
    "sc:hasPurpose": "sc:Purpose",
    "sc:hasExpiryTime": "2 months"
  }
}
```

User Record
a user record is a JSON object with a minimum of the following attributes:

User Record Structure

```
{
  "name": " user name ",
  "email": " email of user ",
  "organization-id": long,
  "dlt": [ on-chain data storage information ]
}
```

Details:

- `user-id`: unique id of the user for referencing purpose provided by the system
- `name, email`: mandatory information about user
- `organization-id`: mandatory link to existing organisation
- `dlt`: optional information about the distributed ledger and required details that are used when this user performs write operations for on-chain data; available options are:
    - `Convex`
    - `Bellecour`

User Record Example

```
{
  "user-id": 1,
  "name": "John Doe",
  "email": "john@doe.com",
  "phone": "(555) 123 4567",
  "organization-id": 1,
  "dlt": [{
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

```
    "type": "Convex",
    "network": "testnet",
    "address": 48
  }]
}
```

Collection Information
a collection information is a JSON object with the following attributes:

Collection Information Structure

```
{
  "name": " collection name ".
  "storage": { off-chain data storage information },
  "dlt": { on-chain data storage information }
}
```

Details:

- `collection-id`: unique id of the collection for referencing purpose provided by the system
- `name`: optional name of the collection
- `storage`: mandatory information about the off-chain data storage provider used for this collection; available options are:
  - `Semantic Container`
  - `Amazon S3`
- `dlt`: optional information about the on-chain data storage provider used for this collection; available options are:
  - `Convex`
  - `Bellecour`

Collection Information Example

```
{
  "collection-id": 1,
  "name": "My Repository".
  "storage": {
    "type": "Semantic Container",
    "url": "https://honey-oc3.data-container.net" },
  "dlt": {
    "type": "Convex",
    "user-id": 1
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

```
    }
}
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

Object Metadata
object metadata is a JSON object with the following attributes:

Object Metadata Structure

```
{
  "collection-id": long,
  "name": " object name "
}
```

Details:

- `object-id`: unique id of the object for referencing purpose provided by the system
- `name`: optional name of the object
- `collection-id`: mandatory reference to the collection where the object should be persisted

Object Metadata Example

```
{
  "object-id": 1,
  "collection-id": 1,
  "name": "My Object"
}
```

3.2 Architecture Diagram

This section focuses on the architecture diagram, a visual representation of the system's components, relationships, and interactions. The diagram provides an overview of the system's design, which helps stakeholders understand the system's structure and key elements.

Figure 3.5 depicts the Gateway API in the context of the overall ONTOCHAIN environment. It spans a data space for internal as well as external services through a well-defined entry point for users. This includes service discovery in the Service Catalogue (public access), requires authentication as a user registered in the accounts management component, allows applications to enlist themselves and interact with other applications through service descriptions, and provides high-level APIs to read and write from available storage providers. The Integration Helper functionality in the Gateway API provides specific interoperability features to assist developers in

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

integrating their applications and services with other - already existing - functionality in the ONTOCHAIN environment.

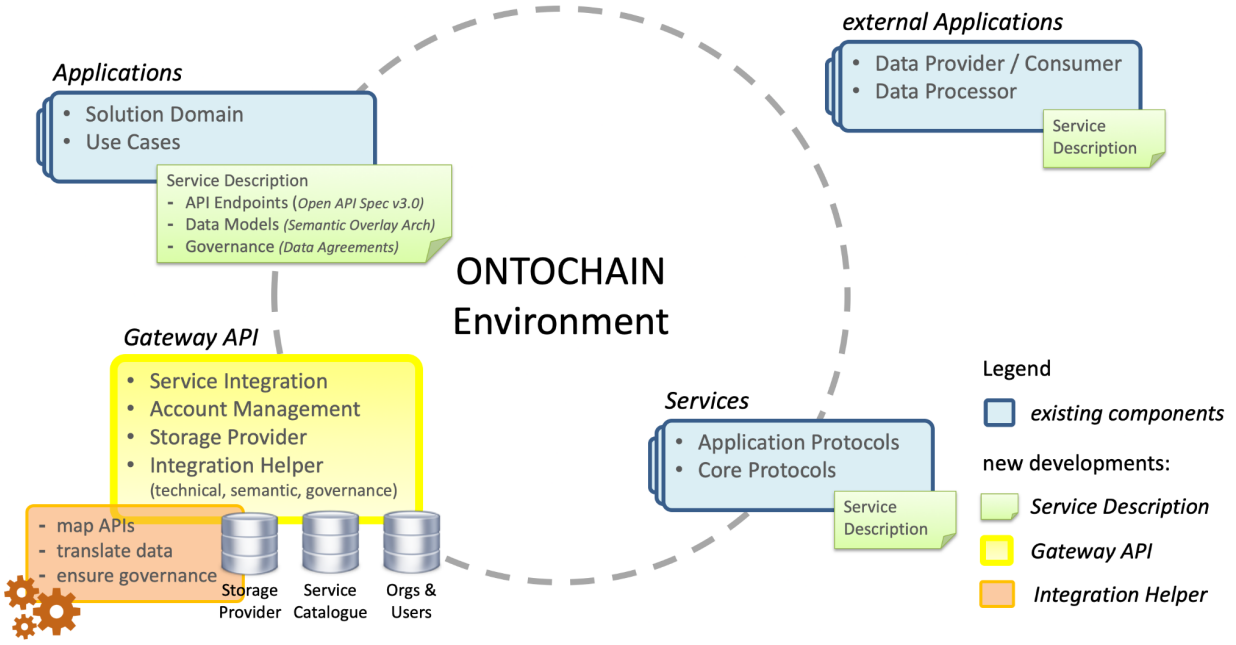


Figure 3.5: Architecture Overview

Figure 3.6 provides an overview of the logical components in the Gateway API. It acts as a Data Controller / Intermediary between different organisations that want to exchange data. It consists of

- a well-defined API endpoints (described in section 4.2) for read and write access,
- uses internally and - where applicable - also externally DIDs for identifying users, organisations, and records
- manages datasets through data models defined by SOyA (Semantic Overlay Architecture),
- ensures governance through the use of Data Agreements that document any data exchange, and
- persists any information in configurable stores (Domain Specific Data Stores), that can be further sub-classified into
    - Service Catalogue: a list of available services
    - Orgs & Users: account management data
    - Storage Provider: configurable general purpose data store that supports off-chain and on-chain storage

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

Note: only hash-values of datasets are stored on-chain to take into account the limited storage capabilities of blockchains; for storing this hash-values Smart Contracts will be deployed on Bellecour and Convex

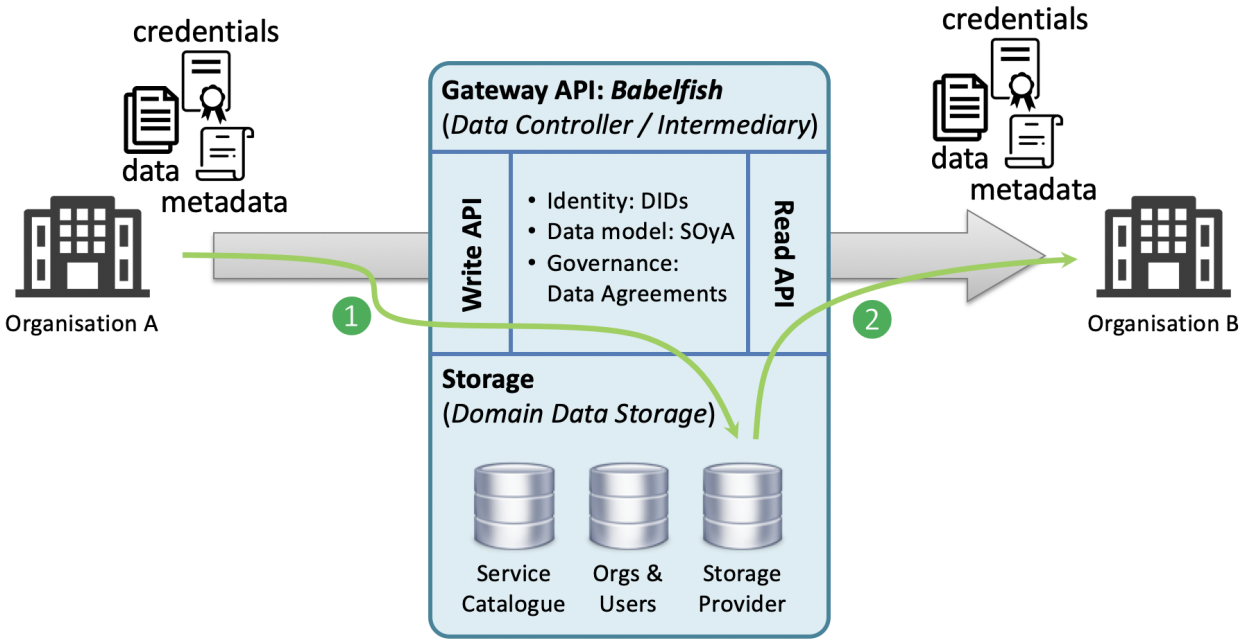


Figure 3.6: Data Flow Between Two Organisations

It is important to note that a data exchange between two organisations through the Gateway API is actually a 2-step process: the intermediary first receives data from Organisation A and persists it (step 1) and upon request for the data from Organisation B the intermediary provides the data. The concrete process that happens in a single step is shown in the sequence diagram depicted in Figure 3.7. Using the example from above in the first step Organisation A is Entity A and the Gateway API acts as Entity B. In the second step the Gateway API is Entity A and Organisation B is Entity B.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

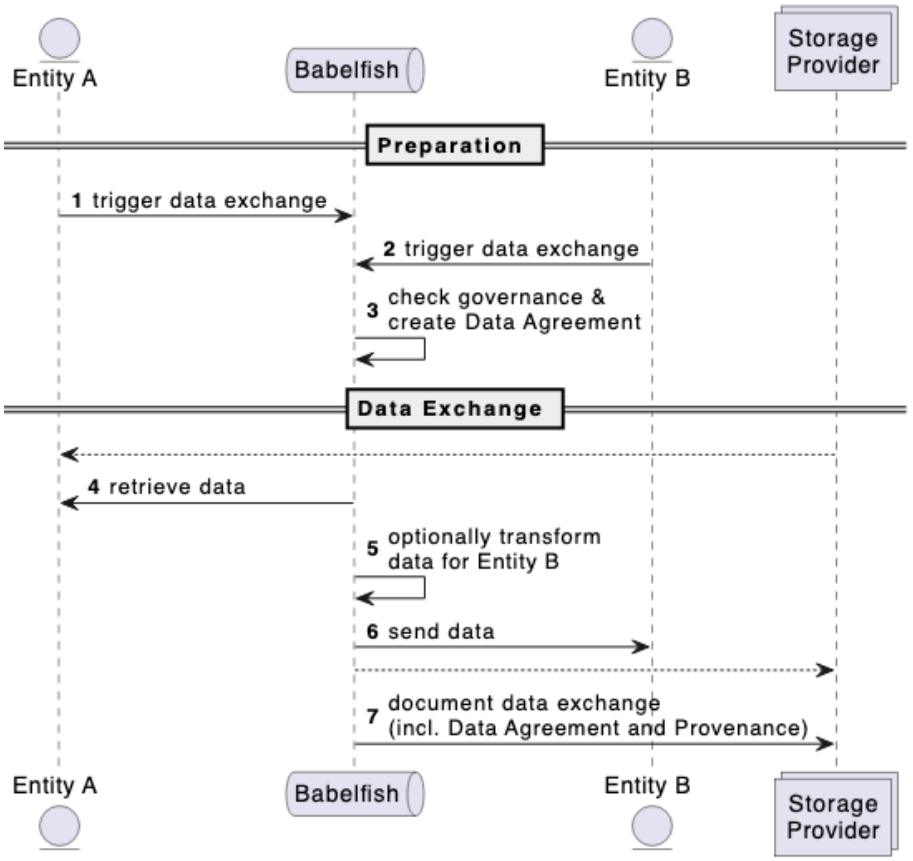


Figure 3.7: Detailed Sequence Diagram for Data Exchange

Step 1 or 2: a data exchange can be triggered by both of the participating entities

Step 3: upon starting a data exchange the Usage Policies (if present) for the participating entities are evaluated and checked for compliance; if successful, a Data Agreement is compiled based on the Usage Policies documenting the governance aspects of this data exchange

Step 4: data is retrieved from the data providing entity

Step 5: in case the applicable Service Descriptions provide information about transforming data between the two entities, this steps performs this transformation

Step 6: the payload is forwarded to the data consuming service

Step 7: the complete data exchange is documented in the configured Storage Provider (configuration is documented on-chain)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

4 Detailed API Specification (preliminary)

4.1 API Specification for SDK Modules

No SDK in this implementation

4.2 API Specification for REST Services

This section describes the REST APIs for the services developed in the course of the project. Details and examples can also be found on the following public HackMD: https://hackmd.io/faNBTCUcSRyQsLOf_Jhdag.

4.2.1 Service Discovery APIs

Service Discovery APIs provide a programmable way of discovering the services provided through the ONTOCHAIN ecosystem. Access to the functions will be authenticated and regular users will only have access to read operations. Write operations will be accessible only to administrators, unless noted in the function's description.

HTTP method	URI	Arguments	Return value	Description
GET	/list?page=X&items=X	page: selected page (default: 1) items: number of items per page (default: 20)	(array of json objects) services matching query	**Retrieve Service Catalogue List (public)** obtain a paged list of all available services; the filter can contain every field from resource description schema
GET	/service?query_field=query_value&field2=value2	service catalogue query in the format: field, value	(array of JSON objects) services matching query	**Query Service Catalogue List (public)** obtain a list of services which name or description match the provided search terms

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

HTTP method	URI	Arguments	Return value	Description
GET	/service/ SERVICE_ID	service_id: numerical identifier of service	JSON object with service description	**Read Service Description (public)** obtain details of a service description
POST	/service	body: JSON object with service description	JSON object with name of the service and assigned service-id	**Create Service Description** provide details of a service description and persist on the configured storage
PUT	/service/ SERVICE_ID	service_id: numerical identifier of service; body: JSON object with service description	JSON object with name of the service and assigned service-id	**Update Service Description** provide details of a service description and update on the configured storage
DELETE	/service/ SERVICE_ID	service_id: numerical identifier of service	JSON object with name of the service and assigned service-id	**Delete Service Description** mark service as deleted on the configured storage

Table 4.1: Service Discovery APIs

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

4.2.2 Organisation and User Accounts APIs

Organisation and User Accounts APIs provide interfaces for creating users and organisations, and basic interactions with wallets. All API calls must be authenticated (i.e., non-public) and write operations are accessible only to administrators.

HTTP method	URI	Arguments	Return value	Description
GET	/organization/ORGANIZATION_ID	organization_id: numerical identifier of organisation	JSON object with organisation details	**Read Organisation** obtain details of an organisation
POST	/organization/	body: JSON object with organisation details	JSON object with name of the organisation and assigned organization-id	**Create Organisation** provide details of an organisation and persist on the configured storage
PUT	/organization/ORGANIZATION_ID	organization_id: numerical identifier of organisation; body: JSON object with organisation details	JSON object with name of the organisation and assigned organization-id	**Update Organisation** provide details of an organisation and update on the configured storage
DELETE	/organization/ORGANIZATION_ID	organization_id: numerical identifier of organisation	JSON object with name of the organisation and assigned organization-id	**Delete Organisation** mark organisation as deleted on the configured storage
GET	/organization/ORGANIZATION_ID/list	organization_id: numerical identifier of organisation	(array of JSON objects) users of the organisation	**Retrieve User List for Organisation** obtain a list of users for given organisation
GET	/user/USER_ID	user_id: numerical identifier of user	JSON object with user details	**Read User** obtain details of a user

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

HTTP method	URI	Arguments	Return value	Description
GET	/user/USER_ID/wallet	user_id: numerical identifier of user	JSON object with user wallet details	**Read User Wallet** obtain wallet details of a user
POST	/user	body: JSON object with user details	JSON object with name of user and assigned user-id	**Create User** provide details of a user and persist on the configured storage
PUT	/user/USER_ID	user_id: numerical identifier of user; body: JSON object with user details	JSON object with name of user and assigned user-id	**Update User** provide details of a user and update on the configured storage
DELETE	/user/USER_ID	user_id: numerical identifier of user	JSON object with name of user and assigned user-id	**Delete User** mark user as deleted on the configured storage and remove all personally identifiable information

Table 4.2: Organisation and User Accounts APIs

4.2.3 Storage Provider APIs

Storage Provider APIs provide users and applications with high-level interfaces that are common to all of the supported backend storage services, both members of the ONTOCHAIN ecosystem, and external services. Storage providers are also registered as services and can be searched in the Service catalogue like any other service.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

HTTP method	URI	Arguments	Return value	Description
GET	/collection/list	none	(array of JSON objects) collection-id and name of collection	**Retrieve Collection List** obtain the list of collection of objects
GET	/collection/ COLLECTION_ID	collection_id: numerical identifier of collection	JSON object with collection details	**Read Collection** obtain details of a collection
POST	/collection	body: JSON object with collection details	JSON object with name of collection and assigned collection-id	**Create Collection** provide details of a collection and persist on the configured storage
PUT	/collection/ COLLECTION_ID	collection_id: numerical identifier of collection; body: JSON object with collection details	JSON object with name of collection and assigned collection-id	**Update Collection** provide details of a collection and update on the configured storage
DELETE	/collection/ COLLECTION_ID	collection_id: numerical identifier of collection	JSON object with name of collection and assigned collection-id	**Delete Collection** mark collection as deleted on the configured storage
POST	/object/OBJECT_ID/ USER_ID	object_id: numerical identifier of object; user_id: numerical identifier of user	JSON object with name of object, assigned collection-id and object-id, and object access information	**Check Object Access** checks the access control for a particular user to access a particular object

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

HTTP method	URI	Arguments	Return value	Description
GET	/object/OBJECT_ID	object_id: numerical identifier of object	JSON object with object details (metadata)	**Read Object** (metadata) obtain details related to an object
GET	/object/OBJECT_ID/read	object_id: numerical identifier of object	JSON object (object itself)	**Read Object** (object itself) obtain the object
POST	/object	body: JSON object with object metadata	JSON object with name of object and assigned object-id & collection-id	**Create Object** (metadata) provide details of an object and persist on the configured storage
PUT	/object/OBJECT_ID/write	object_id: numerical identifier of object; body: JSON object (object itself)	JSON object with name of object and assigned object-id & collection-id	**Write Object** (object itself) provide object and persist on the configured storage
PUT	/object/OBJECT_ID	object_id: numerical identifier of object; body: JSON object with object metadata	JSON object with name of object and assigned object-id & collection-id	**Update Object** (metadata) provide details of an object and update on the configured storage
DELETE	/object/OBJECT_ID	object_id: numerical identifier of object	JSON object with name of object and assigned object-id & collection-id	**Delete Object** mark object as deleted on the configured storage

Table 4.3: Storage Provider APIs

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

4.3 ONTOLOGIES

Name	Description	Domain	New/ MyUpdate/ MyReused/ Reused	language
Data Agreement	the data agreement add metadata setting the rules for sharing of data	data intermediation	New	JSON-LD
SOyA Ontology	a data model authoring and publishing platform that also provides functionalities for validation and transformation	data model management	MyUpdated	OWL
DID Ontology	specifies the DID syntax, a common data model, core properties, serialised representations, and DID operations	identity management	Reused	OWL
SOyA DID Ontology	extension of the DID ontology for describing constraints of DID documents using SOyA	identity management	New (extended)	OWL, SHACL
DTLF / FEDeRATED	specifies the semantic model for transport sphere and enables a multi-modal transport labelling approach	data model management	Reused	RDF turtle
GS1	specifies DPP vocabulary for the production, transport and marketmaker spheres. DPP architecture “network of resolvers”	data model management	Reused	RDF turtle, JSON-LD
GoodRelations	specifies DPP vocabulary for marketmaker sphere	data model management	Reused	OWL, RDFa

Table 4.4: Ontologies

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 957338

5 Detailed Work Plan for Implementation and Deployment (final)

The work plan for implementing Babelfish during the 10-month funded project duration takes a stepwise approach and is depicted in Figure 6.1.

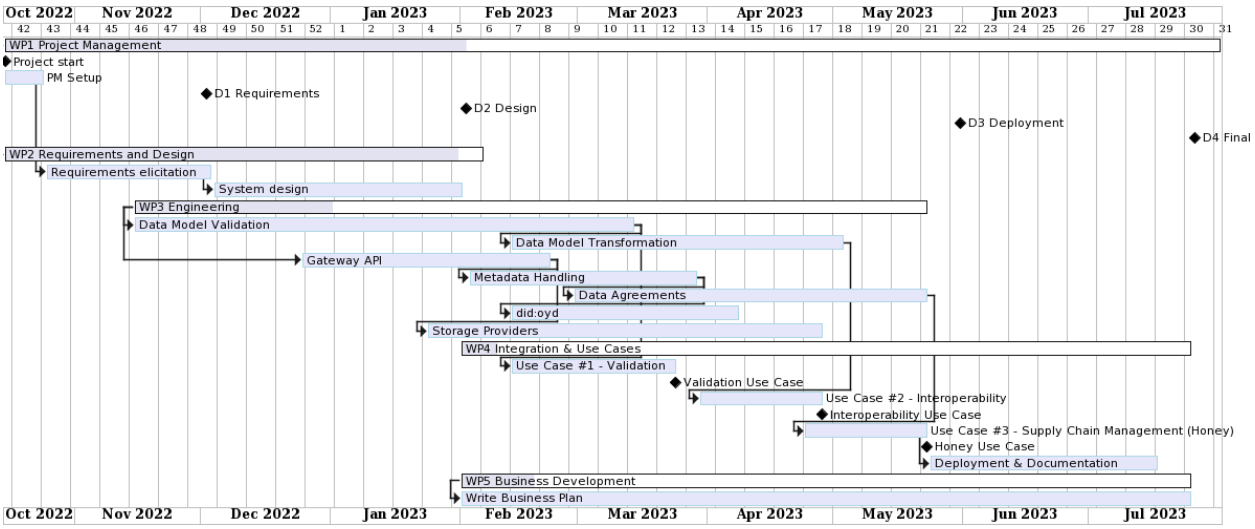


Figure 6.1: GANTT Chart

There is a dedicated time for testing - especially for integrating with other ONTOCHAIN projects - during the "Deployment & Documentation" phase in June and July 2023. We are also committed to support Babelfish and the developed services beyond the NGI ONTOCHAIN project, and are looking forward to collaborate on relevant business cases.

5.1 Work Plan for Implementation

The implementation of the Babelfish project comprises seven building blocks that are described below in the order we start working on it.

Data Model Validation

Data Model Validation builds on the Semantic Overlay Architecture functionality of verifying a dataset against given constraints. This was already demonstrated in an early technology preview for DID Document validation (online service available here: https://didlint.ownyourdata.eu) and will in the next step be extended to Usage Policy and Domain specific Data Agreement validation, and shall eventually provide a general validation service.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

This general validation service will be used as a demo case for the Service Catalogue in the Gateway API and will be the first available use case - see section 2.4.

Target date: mid March 2023

Gateway API

The Gateway API component is the single entrypoint for developers and users to interact with the results of the Babelfish project. Section 4.2 describes the exposed API endpoints for Service Catalogue, Accounts Management, and Storage Providers.

This component will be complete when a full test suite (using automated tests based on `pytest`) is available.

Target date: end of February 2023

Storage Providers

For managing collections and objects with the Gateway API it shall be possible to choose between at least two alternatives. And since data can be maintained either on-chain (i.e., data stored on a distributed ledger) and off-chain (for larger datasets where blockchain would either be too expensive or because of legal reasons not desired) the following 4 storage providers will be made available:

- Semantic Containern(off-chain data)
- Amazon S3 (off-chain data)
- Convex (on-chain data)
- Bellecour (on-chain data)

Each storage provider will include its own automated tests that will be added to the Gateway API test suite.

Target date: end of April 2023

Metadata Handling

The metadata handling in Babelfish is an important building block to collect and automate annotating datasets that pass through the Gateway API. Relevant metadata handled by Babelfish include Usage Policies, Provenance and Schema information, and handling of identifiers - this includes links between data that is stored off-chain but is also persisted through a cryptographic hash value on-chain.

Target date: end of March 2023

Data Model Transformation

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

Data Model transformation is about using alignment information between different ontologies and automatically deducing how datasets can be transformed. Concrete examples from various domains including supply chain will be used to validate the approach and demonstrate the capabilities. We plan to use this method also for generating a Digital Product Passport that summarises data from different sources along a value chain.

The results from this component will be made available as a documented use case for interoperability as described in section 2.5.

Target date: begin of May 2023

`did:oyd` Method

Identifiers are a crucial element in referencing datasets and Babelfish will heavily rely on DIDs for this. Additionally, Verifiable Credentials (VCs) and Verifiable Presentations (VPs) using did:oyd will be used for attestations from third parties. The did:oyd method will be used as the DID method and to ensure interoperability our focus is on standard compliance and community engagement. Therefore, in addition to implementing relevant features like delegation and VC & VP management, we will also release an updated detailed W3C conform specification of the did:oyd method and ensure full compliance with the DID Test Suite[6].

Target date: mid April 2023

Data Agreements

The Babelfish project will continue the work on Data Agreements from ONTOCHAIN Call #2 funded project PS-SDA and will develop it further from the personal data space domain into other domains like supply chain management. Usage Policies are used to make it easier for organisations to provide initial information for compiling Data Agreements between multiple entities and we will use Verifiable Credentials and Verifiable Presentation to demonstrate consent from all involved parties.

Based on the Supply Chain Management Use Case (described in section 2.6) a complete end-to-end example will showcase the use of Data Agreements.

Target date: mid May 2023

5.2 Work Plan for Deployment

[6] https://w3c.github.io/did-test-suite/

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

We plan regular deployments of the developed and updated software components in the course of the project. Our primary site for testing functionality and deploying off-chain storage is a Kubernetes cluster maintained by OwnYourData and services will generally be available as a sub-domain of data-container.net and ownyourdata.eu.

For on-chain data we are using Convex and Bellecour from iExec as distributed ledger technology. Deployment of smart contracts will take place during implementing the Storage providers and should be available for testing in April 2023.

5.3 Risk Analysis

The potential risks in the project that could delay implementation as stated in the work plan in this chapter are listed below.

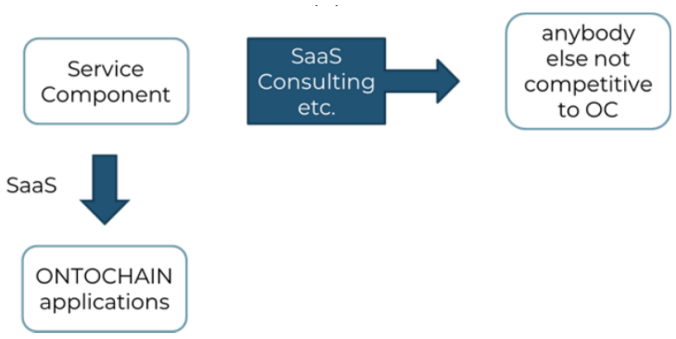
- Complexity of data model alignment and automated transformations
  Mitigation: stepwise approach and use-cases addressing different aspects of alignment
- Integration problems with existing ONTOCHAIN infrastructure and other projects
  Mitigation: close monitoring of planned milestones and regular reporting in Scrum meetings
- Resource availability - because of the small team size an unexpected absence of a team member would jeopardise project completion
  Mitigation: substitutes were established where possible and regular internal team meetings ensure to address any problems early on

The integration of ONTOCHAIN services in the Service Catalogue is crucial and Babelfish is an important aspect as a single point of entry for the ONTOCHAIN project. We will provide detailed information in the form of documents and tutorials to be shared with other projects in order to accelerate and define requirements.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

6 Business Model and Exploitation Plan (Preliminary)

6.1 Business Model Description



The business model for an ONTOCHAIN service component like Babelfish is basically described by the provision of SaaS and related Saas consulting complemented by dedicated business development consulting. Therefore, we established sales cycles with ONTOCHAIN participants (Convex, PS-SDA, TruSSIHealth, Origintrail DKG, GIMLY ID) which are positioned in ONTOCHAIN value network to experiment with loosely coupled exchange constellations and fulfil our role as Gateway API service component. The ONTOCHAIN platform is proposing to withhold a certain percentage of its application service fees paid by end-users of main business use cases.

A second business scenario for the Babelfish capabilities is to design and coordinate an ecosystem business model during our early engagement activities: For this scenario, Kybernos is acting in roles as initiator, ecosystem steward and knowledge worker for design artefacts Digital Product Passport DPP based on Domain specific Data Disclosure Agreements (D3A).

Shared narrative and mission for the ecosystem: “design and lead with purpose to transform the food supply chains towards net zero goals”. The ecosystem steward has detailed understanding of supply chain network design, strategy, standards and terminology. Regarding the strategically designed SaaS provision of a food value chain specific DPP functionality we place our business development focus on

1. the **design phase** to facilitate domain data exchange: we have designed functionality specifiable and programmable with SoyA data models on attribute level around the storage provider within an ecosystem-relevant governance element with requirements regarding data and value exchange. This data intermediation service provider DISP, the commercial provider of the DPP scheme which we design for the domain data agreement (upstream) and the domain data disclosure agreement (downstream). We experimentally expose and mint at least two concrete purposes: “transport labelling” and “geographical designations of origin”.
2. We **motivate early adopters** to share the selected purposes, to perform all needed preparatory tasks and join the ecosystem: a beekeeper, a specialised

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 957338

transport service provider (a cargo bike service) and a food shop register their organisations and their acting users via Babelfish Gateway API or a "minimum frontend" and create their credentials to fulfil their roles in our sequence diagram based on OYDIDs VC/VP functionality.

3. **Data stewardship** the data governance principles of an organisation to ensure data quality and consistency. It includes:
    a. Knowing all data that has relevance for the ecosystem
    b. Dealing with competitively sensitive data, usability, trust, and reliability
    c. Understanding the location of data
    d. Maintaining acceptance, transition transparency of implemented data models and accuracy of data
    e. Implementing usage policies, i.e., Domain specific Data Agreements through the data intermediary and Domain specific Data Disclosure Agreement (D3A) for using data in a Digital Product Passport
    f. Enabling the ecosystem to utilise domain data to gain a competitive advantage
    g. Advocating the use of reliable data in wider domain data space

Minimum design artefacts and system function:

- Purpose (sharing of competitively sensitive information)
- Data models (SoyA-Process)
- Data Intermediary concept (Intermediary, Storage Provider, Domain Data Controller, DPP scheme as ultimate business goal)
- Wallets / Front Ends / Connectors
    - for knowledge workers as system designer & shaper
    - for scheme participants
    - for customers

We use net zero policy-driven incentive structures for domain data exchange (production, transport) and we will be able to resell third-party service components (Targomo).

We will be validating our arguments and eventually the designed treatment with real demand and users according to the early adopters engagement plan.

6.2 Business Value for the Blockchain Domain in General

The Babelfish project provides data intermediation project designers some basic orchestration functionality as generically described in our generic user stories and in our ecosystem user story for the food supply chain domain. The applicability of this set of functionality within the supply chain management domain in section 2.6 is able to demonstrate two concrete purpose-driven data and therefore value exchange constellations within the dominant supply chain transparency framework which is

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

based on some crucial blockchain functionality despite the fact that some blockchain solutions of the 1st blockchain wave in supply chains (e.g., Maersk's IBM Trade Lens) have to transform or even already stopped their operations due to unsolvable governance challenges.

Using light-weighted components like Babelfish in the food supply chain has the potential to improve supply chain transparency and traceability bottom-up and create some upstream momentum as well as reduce administrative costs related to the limited existing solutions for transport labelling and designation of origin as our selected purposes. New blockchain-based mechanisms as offered by the ONTOCHAIN environment are presenting promising opportunities for the transformation of existing supply chains towards rising net zero goals of supply chain actors.

A supply chain featured with strategically designed blockchain together with non-blockchain functionality can help supply chain actors record price, date, location, quality, certification, and other for DPP-purposes relevant information to more effectively manage the supply chain.

6.3 Business Value and Relevance for ONTOCHAIN

What is the link between your business and ONTOCHAIN?

Regarding the reference document for Ontochain's business model we follow the valid scenario "Babelfish as service component" for the Ontochain software ecosystem and demonstrate the Babelfish software modules in the supply chain data domain.

What would the value exchange be?

At the end of the design phase, we see three interconnected value exchange scenarios based on given storage provider mechanisms which we plan to validate during early user engagement phase:

1. OYD-based value exchange: Demonstrate domain specific DISP functionality
2. Convex-based value exchange: Value exchange via "by juice" as DPP scheme currency
3. Origintrail-based value exchange by TRAC token; see best practice "SCAN Trusted Factory"

The potential success of DGA-compliant data intermediaries crucially rests on some factors: the accuracy of the claim that a lack of trust has been the main reason hampering the development of data intermediaries, and the regulation's ability to promote the needed levels of trust in data intermediaries.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

In order to be accredited as a provider of data intermediation services, providers must therefore play an active role in establishing direct commercial relationships between businesses..

As Recital 28 DGA clarifies, it is not sufficient to merely provide the technical tools for data sharing without the aim to establish or gather information on commercial relationships between data holders and users. Data intermediaries must actively assist in the establishment of (direct) commercial relationships for the purposes of data sharing through technical, legal or other means. In other words, data intermediaries must act as matchmakers by connecting data holders and data users with each other, thereby initiating data transactions. (von Ditfurth, Lienemann 2022 - The Data Governance Act)

We directly contribute with our Babelfish approach to

- the Specific Objective 1 ONTOCHAIN ECOSYSTEM Setup, with Gateway API and Storage Provider;
- the Specific Objective 2 ONTOCHAIN Technological Framework Design, with privacy aware and secure data exchange and value sharing and participation/contribution incentives;
- the Specific Objective 3 ONTOCHAIN Ecosystem experimentation, with supply chain knowledge workers to form a transformative supply chain DISP.

6.4 Any Other Impact

With our designed Babelfish capabilities to enable various ESG purpose-focused data exchange we support the external effects' policy stream in both directions:

- internalisation of negative external effects by design principle "accountability"
- incentivisation of positive external effects (network effects with loosely coupled operational system connecting the spheres)

We anticipate a strong demand for data-driven ESG policy innovation on EU level as well as on all multi-level governance (NUTS) levels below due to already existing "net zero" target claims.

Two business risk categories which are inherent to the second business scenario of the Babelfish project could create positive impacts towards a widely shared governance for supply chain transformations:

1. The inherent **complexity risk** of the Babelfish project
    a. mitigation by safeguarding project management structures across various stakeholders involved; understand and help drive complex

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

design and development decisions; incentivise participation by purpose-orientation across the stakeholders involved.

b. mitigation by purpose-oriented commercial risk analysis of design artefact "competitively sensitive information" as subtasks for data agreement's purpose specification.
c. mitigation by coordination with key stakeholders, other DPP system shapers, engagement with a knowledge workers of stakeholder groups

2. The inherent **governance risk**, market dynamics risk, market domination risk:
   a. description: There is always a risk that an overly rigid interpretation of Article 101 of the Treaty on the Functioning of the European Union (TFEU) can actually have negative effects on competition and innovation. It is becoming increasingly clear that the ability to collect and retain exclusive control over data could also be used as a market control mechanism. If a competitor or an innovator would require access to certain data to build or expand its own services, and this data would only be accessible with the cooperation of the data holder, then that data holder could in principle exercise significant influence over its market (Graux 2022 - Sharing Data (Anti-)Competitively)
   - mitigation: design and pre-specification of data agreements along given categories to enhance DPP's acceptance in food sector
   - mitigation: Initiate the DISP as organising entity around the storage provider functionality with conscious design loops to enhance stakeholders' acceptance
   - mitigation: to enable favourable purpose-focused supply chain cooperations – which inevitably require some data sharing – to exist, the EU supports a mechanism of so-called Horizontal Block Exemption Regulations (HBERs).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

7 Early User Engagement Plan

Our first engagement with project outcomes relevant to an external user group was with the Decentralised Identity Foundation (DIF). We presented the DID Lint service (a DID Document validator) in the DIF Identifiers & Discovery Working Group in early January and received very positive feedback to our work. We plan to regularly post updates and news about our developed services and how it relates to DIF standards. Additionally, we can participate in DIF events and conferences as speakers to further build relationships with this community.

The Data Agreements should be based on a format that is interoperable. We plan to engage closely with the DIF data agreement task force to select a method of signing the agreement. Additionally we will also be checking what fields in the data agreement are being considered and introduce some of the concepts we are introducing in this project. Namely that data agreement may be used in domain specific areas and not associated with individuals but organisations.

Another community where some of our team members are already well connected is the MyData Network: MyData is a global network of individuals, organisations, and communities working towards empowering individuals with their personal data. Building on the work from ONTOCHAIN call #2 funded project PS-SDA about Data Agreements, we are extending the concept towards domain specific Data Agreements. By actively participating in MyData events and online forums, we can connect with users who are passionate about personal data privacy and use feedback from this group.

Another way to engage with users is to host workshops and webinars that focus on the use of our services and its benefits. These events can be targeted towards specific user groups such as developers, privacy advocates, and supply chain management stakeholders. During these events, we can provide hands-on training, demos, and open discussions to further educate users on our service and its features. This will not only help us build a loyal user base but also generate positive word-of-mouth for our product.

Specifically in the domain of supply chains the demonstrations of both purposes (transport labelling and designation of origin) are planned with a short supply chain around the City of Vienna. Therefore we will directly engage the needed stakeholder expertise and traction via the Kybernos network. This network comprises the Clever Clover FMCG accelerator and its portfolio of FMCG startups.

We develop our scalability matrix along three dimensions:

- **supply chains**: we start with a short non-industrial, regional honey supply chain around Vienna. This experiment should be the basis for our sales cycle with the City of Vienna and further regional FMCG supply chains which we want to elaborate together with our strategic partner Clever Clover.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

- data sharing **purposes**: we start with designation of origin as purpose within the production sphere and transport (mode) labelling as purpose within the transport sphere; we combine both purposes in a DPP and are therefore actively demonstrating within the market maker sphere.
- **jurisdictions** (NUTS regions): we start in the eastern region (NUTS:AT-1) of Austria (NUTS:AT). Our Food Hinterland System FHS policy partner, the City of Vienna (NUTS:AT-13) is the first example of NUTS-2 basic regions which we will address for the application of regional policies. The next strategic region ist Lower Austria (NUTS:AT-12) which is a crucial system partner of Vienna's food hinterland. Our concrete honey experiment will touch NUTS-3 and small regions for specific diagnoses regarding our defined purposes in Lower Austria and/or Vienna.

To fully elaborate the market potential of a DPP we are strategically integrating with the 20+ living labs of the Federated Project and EU's Digital Transport and Logistics Forum DTLF. Additionally, Kybernos is a member of relevant expert groups of UN/CEFACT.

We are aiming at the "Marktamt", the market office authority in the City of Vienna to leverage the outcomes of our Food Hinterland System project with Babelfish. The most concrete business stream is with Clever Clover as Europe's #1 accelerator for FMCG startups. Therefore, we are connected with an Austrian food startup cluster which has some players with European scalability which are already funded by the EIC funding scheme.

Another already prepared "real option" of our business model beyond the Ontochain project: we can address and specify the designed DPP structures for industrial supply chains with soft commodities like the sugar supply chain.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338

8 Conclusions

This document outlined the design of the planned Gateway API and accompanying use cases of the Babelfish project. It addresses the key requirements and challenges identified in the project scope. The system architecture is robust, scalable and easy to maintain, making it suitable for the intended use case. It aims to make the integration with existing systems (other projects) seamless, improves interoperability, and meets the objectives of the project.

Overall, the proposed design is a comprehensive solution that caters to the needs of the stakeholders and delivers a high-quality user experience. The team is confident that the proposed design will be successful in meeting the project objectives and delivering the desired results. We are committed to providing ongoing support and maintenance to ensure the continued success of the project.

Based on this design and the implementation & deployment plan the implementation will be described in deliverable D3 Implementation scheduled for end of May 2023.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957338