H2020-INFRADEV-2019-2
Grant Agreement No 871043

DiSSCo Prepare WP D6.2

Implementation and construction plan of the DiSSCo core architecture

Work package lead: Claus Weiland

Authors:

Sam Leeflang	0000-0002-5669-2769	Naturalis
Claus Weiland	0000-0003-0351-6523	Senckenberg Nature Research Society
Jonas Grieb	0000-0002-8876-1722	Senckenberg Nature Research Society
Mathias Dillen	0000-0002-3973-1252	Meise Botanic Garden
Sharif Islam	0000-0001-8050-0299	Naturalis
David Fichtmüller	0000-0002-0829-5849	Botanischer Garten und Botanisches Museum Berlin-Dahlem
Wouter Addink	0000-0002-3090-1761	Naturalis
Elspeth Haston	0000-0001-9144-2848	Royal Botanic Garden Edinburgh

Version:

Version	Date	Contributors	Comment
0.2	14/04/2022	Sam Leeflang	Foundational draft of the technical setup for the DiSSCo infrastructure
0.4	14/05/2022	T6.2 members	Revised version, added introduction of openDS
1.0	(till) 19/05/2022	T6.2 members Claus Weiland Wouter Addink	Elaborated status of openDS and preliminary work from ICEDIG (CW/WA), iterative revision by T6.2 members
1.1	19/05/2022	Claus Weiland	Final editing, submitted to project management as V1.1
1.2	24/06/2022	Sam Leeflang, Wouter Addink, Claus Weiland	Resolved last comments

Preface

This report details the system architecture for DiSSCo's concept of the Digital Specimen, which acts as a *surrogate in cyberspace for a specific physical specimen*[1]. This surrogate or *digital twin* encompasses and provides persistent linking to relevant information artefacts, derived or related to the physical specimen. The Digital Specimens are embedded in a FAIR ecosystem of system architecture and services described in the following sections, since specimen data cannot be FAIR by itself (as any data) if there isn't an infrastructure which implements policies, rules and procedures for FAIR.
Our achievements outlined in this report are based on earlier fundamental research work, architectural planning and implementation work led by Alex Hardisty within the frameworks of ICEDIG and DiSSCo Prepare, and also on input from the DiSSCo technical team and

[1] https://dissco.tech/2020/03/31/what-is-a-digital-specimen/

Dimitris Koureas. In particular our joint discussions between the technical team and Alex are - besides his groundbreaking works on FAIR Digital Objects for specimen data - visible in many parts of this text and have provided a clear guidance for the implementation and construction plan of the core architecture outlined in this report. The architecture as described underpins the nine protected characteristics for data management as described by Alex Hardisty et. all in the DiSSCo provisional data management plan (Hardisty 2019).

Abstract

Work Package Deliverable 6.2 aims to propose an architectural overview of the core components of the DiSSCo data infrastructure. This overview should provide clarity to stakeholders and guidance for the development team. The architecture is based on the earlier work done in the EC funded ICEDIG project in work package 6, discussions in the DiSSCo Prepare Work Package 5 and 6 meetings, DiSSCo Prepare end-user services development sessions and DiSSCo Prepare Technical Team meetings. The data infrastructure aims to support mass digitisation, to provide a FAIR digital representation of specimen data based on DiSSCo's Digital Specimen concept. This approach provides rich provenance information through the full data lifecycle and enables enrichment through annotations and linking related data objects, with particular emphasis on machine actionability and support for AI.

In total 23 different components were identified, grouped in 9 containers. Each component is an independent unit built for its specific purpose. The components have been grouped in containers, each aimed at providing a specific functionality. The communication between the different components is based on events using queues as a way to decouple the components.

In this document the function of each component is described as well as its role within the broader container. We work through the architecture starting with the data providers, where the data is generated. The provided data are collected by the translators, which convert it into the generic openDS data format, based on community standards like Darwin Core. After translation, the enrichment services are a vital part of the architecture as they provide an automated way to enrich the data with additional information. Data is validated and stored with the data processing services in the data storage layer. Data exposure goes through a wide array of DiSSCo end-user services providing a range of functionalities to use and annotate the data. Last but not least are several services aimed at orchestration, providing new and updated data models or providing globally unique persistent identifiers. This wide collection of services working together in unison will provide a stable but extensible architecture for the DiSSCo Research Infrastructure.

Keywords

FAIR Digital Object, Distributed System of Scientific Collections, DiSSCo RI, openDS, Digital Specimen, FAIR Digital Object, Digital Twin, Architectural overview, Data flow, Microservices, Event driven, Auto-scalable, Stateless

INDEX

Introduction and objectives

This document aims to describe the setup for the DiSSCo's technical infrastructure. Many of the design decisions and architectural choices were influenced by the FAIR principles (Wilkinson 2016) and the more technical implementation focused approaches of FAIR Digital Objects (FDOs, De Smedt 2020) and Digital Object Architecture (DOA, Kahn 2006). At the heart of this approach stands the Digital Specimen (DS), a FDO type for the biodiversity domain acting as a digital twin in the internet for a specific physical specimen in a natural science collection. The Digital Specimen encapsulates and persistently links to information artefacts which are about the physical specimen like sequences, images and biochemical or taxonomic determinations (Fig.1).
As it applies to all FDO types, a DS can achieve FAIRness only by embedding it into a FAIR ecosystem (European Commission 2018). This ecosystem has to provide essential key components including services for minting and resolving of PIDs (Hardisty 2021), repositories to index and catalogue DS and their relations (Hardisty 2019b) as well as registries for community-endorsed type specifications for DS and other curated biodiversity objects.

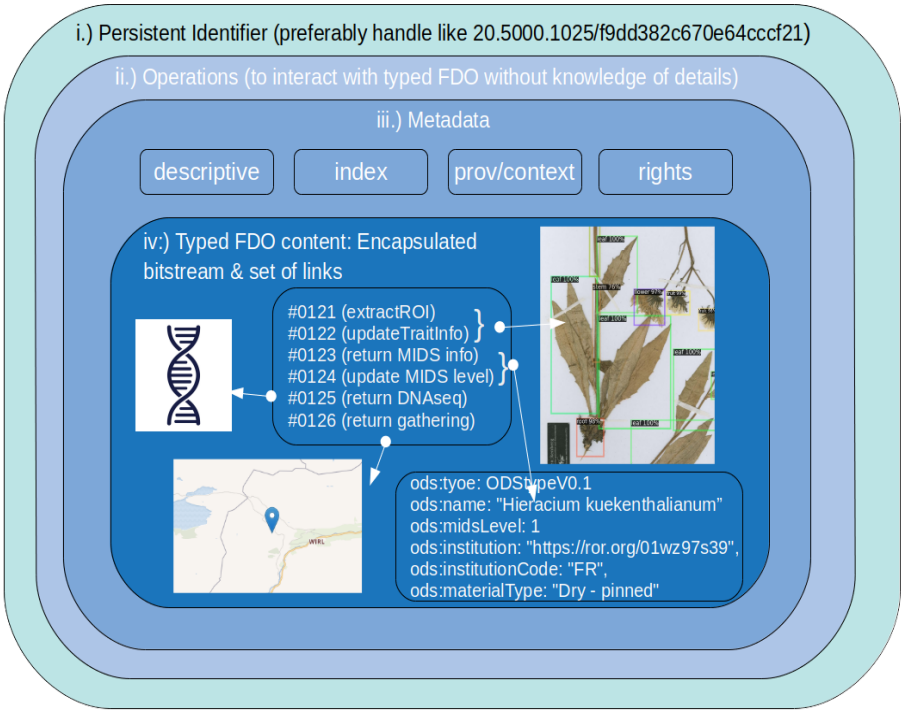


Fig. 1: A Digital Specimen substantiating a FDO type for a natural science collection object expressing the layered object structure of FDOs: (i) FDOs are identified by a PID resolvable to a PID record associated with Kernel Attributes; (ii) via type-specific operations, these attributes give access to (iii) metadata information, that enable self-contained acting of machines according to (iv) typed content and based on decision between alternative operations in a given context (machine actionability). The example shown here is based on a machine learning pipeline for feature extraction (morphological traits) from a DS containing image objects (Wittenburg 2022).

Several of those core components were informed by global discussions within the natural sciences collections and biodiversity informatics community and recommendations from the Research Data Alliance (RDA, Islam 2020).
It is important to note that this is a living document and that the current setup is a draft. Based on discussions, reviews, remarks and growing insights through pilot implementations it will change and evolve. The ICEDIG project provided the design blueprint and the provisional data management plan (DMP) for the DiSSCo architecture. While the setup of the technical infrastructure is foreseen to evolve over time, the 9 essential characteristics for data management that are described in the DMP should remain constant:

- Digital Specimen is the core component and the primary digital object type of the DiSSCo architecture
- Accuracy and authenticity of the digital specimen
- FAIRness
- Protection of data (legal regulations and community norms)
- Preserving readability and retrievability
- Traceability (provenance) of specimens
- Annotation history
- Determinability (status and trends) of digitisation
- Securability (authentication, authorization, accounting, auditing)

A detailed overview of the entire core architecture is provided in Fig. 2, subsequent sections review the infrastructure components involved and their workflow.[2]

Each component can generally be seen as a single application, although some components can be deployed as multiple instances[3], possibly with slightly different functionality[4]. This is shown in the image as the component with a postfix (X, Y, Z or 1, 2, 3). These different applications of the same component might have different implementations, but are considered at the level of this architectural overview as a single component due to their similar functionality. An example is the translator services where different instances are used for different data providers.

The lines are drawn from the action initiator to the action recipient. This means that when a user uses a frontend, the User is the actor and the frontend the recipient. The arrow will be from the user towards the frontend. When a component retrieves data from an API, the component is the actor as it initiates the request, the API is the recipient and will respond. Components which work together to provide a certain broader functionality are grouped as containers.[5] For example, in the data storage container, we have several components which ensure that the data is persistently stored and searchable. All these components work together to persist and retrieve data. Together they form the data storage container. Several containers may contain only a single component. We will still contain them in a container as more applications might be added to the container.

---

[2] Based on the components level in the C4 model for visualising software architecture, https://c4model.com/
[3] For horizontal scalability
[4] With the help of feature toggles, https://martinfowler.com/articles/feature-toggles.html
[5] Based on the container level in the C4 model for visualising software architecture, https://c4model.com/

In this document we will follow the workflow from start to finish. In the overall diagram (Fig. 2), this means from left to right. Starting at the DiSSCo facilities, through data ingestion, enrichment services, data processing and data storage towards DiSSCo services and the end users. It will briefly describe each container and within each container, each component. Below is the complete overview of all the components. It has been condensed to provide a general overview. For each container a more detailed overview is added at the appropriate chapter.

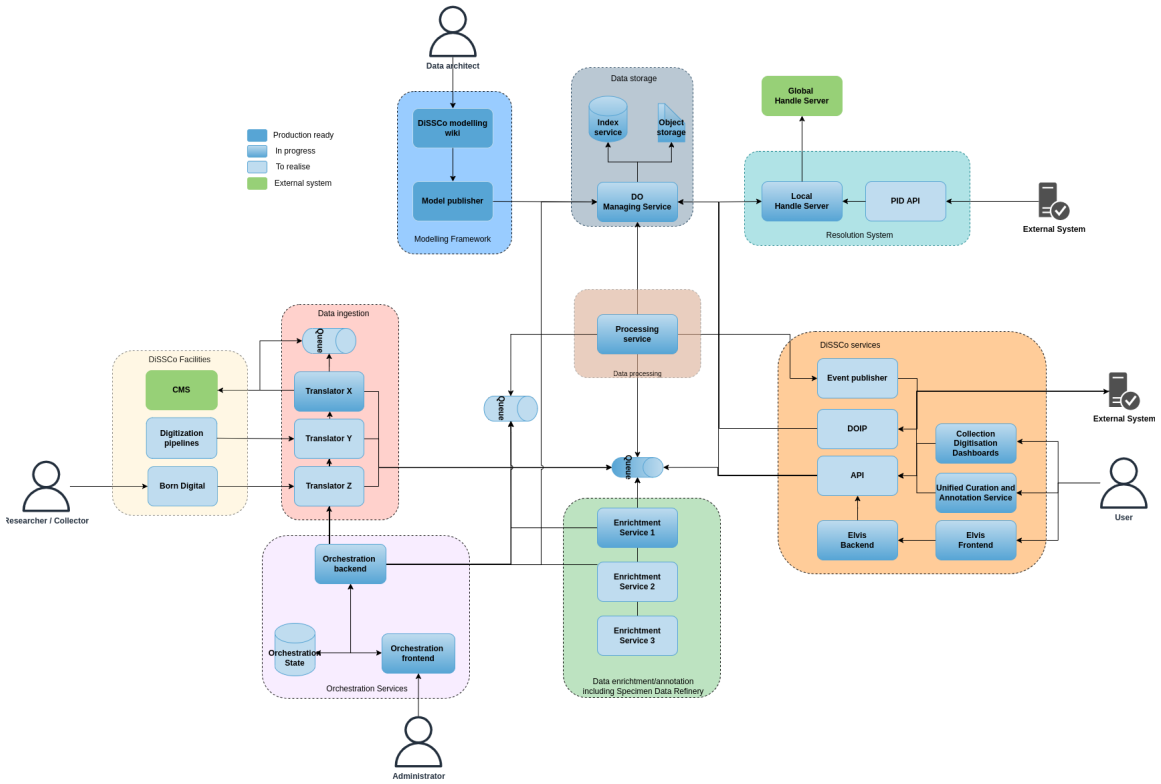


Fig. 2: Architectural overview of the DiSSCo data infrastructure

DiSSCo Facilities

Starting at the far left of Fig. 2 are the DiSSCo facilities. The DiSSCo facilities are defined by A. Hardisty (Hardisty 2019) as:

"*The geographically distributed collection-holding organisation(s) (i.e., natural science/history* collection*(s)) and related third-party organisations that deliver data and expertise to the DiSSCo Hub infrastructure, and which can be accessed by users via the DiSSCo Hub infrastructure*."

The DiSSCo facilities are the data generators that will provide DiSSCo with the necessary data. They also hold the authority over the data. There are three different types of facilities

that can generate data which are described below. All three different ways of data production will need to be supported in the DiSSCo infrastructure

Collection Management Systems (CMS)

This group of applications are used to curate and manage already digitised specimen data.

Different DiSSCo facilities will use different data formats and access protocols when making the data available and some institutions might not have a functional CMS at all, only relying on spreadsheets or other forms of simple file storage. Access to this data is often possible through data portals, aggregator ingestion endpoints such as IPT or BioCASe or less readily through personal requests to system managers.

Digitization pipelines

Digitization pipelines such as mass-scanning workflows will generate new data which needs to be stored and curated (Hardisty 2020). Rather than operating through a CMS intermediary, this data may be directly or at least independently pushed to the DiSSCo infrastructure. DiSSCo might play a role in the standardisation of this process or provide applications and tooling for mass digitization. However as this document describes only the core infrastructure of DiSSCo, the digitization process itself is seen as out of scope for this document.

Born digital specimen

Born digital specimens are specimens which are digitally registered before they are accessioned and physically added to a collection[6]. These specimens should be able to be added directly into DiSSCo whereby the link with the physical specimen is maintained. This is a particular challenge that distinguishes this data source from digitization pipelines. This way, when further information becomes available, it can be added to the digital object. The DiSSCo core infrastructure expects that the tooling used for the born digital specimen will keep it up to date and facilitate links with specimen-related data such as those of chemical analyses, DNA sequencing or annotations made in crowdsourcing platforms such as iNaturalist.

DiSSCo Data ingestion container

The Data Ingestion function group handles data retrieval/delivery, ensures that the data can be parsed to a DS and is sent to the processing services.

Translator services

Translators are the first step in the data ingestion process and the first point of entry into the DiSSCo infrastructure. They need to retrieve data from a great variety of sources (the

[6] DiSSCo will also handle destructive sampling requests (see user story) where researchers can request specific specimens for DNA, microscopic or chemical analysis. Depending on the institution's workflow, some of the digital data can be captured prior to accession and the physical specimen might be destroyed. However, the relevant digital information needs to be stored and findable.

DiSSCo facilities) in a variety of data models (Darwin Core, ABCD(EFG), local models, etc.) with a variety of data exchange formats (JSON/JSON-LD, XML, CSV, etc.) and architectures (REST, GraphQL, etc.). We will therefore design an application which can handle all these different types of data and protocols. Based on the profile the application can be configured to run the steps needed for the data retrieval and conversion. When a new protocol or data model needs to be added we will only need to create a specific step for the retrieval or parsing without the need to create a whole new application.

The main functions of the translator services are to connect to the DiSSCo Facility and retrieve/receive the data. Then translate the data to valid openDS (and finally publish the DS to a queue.

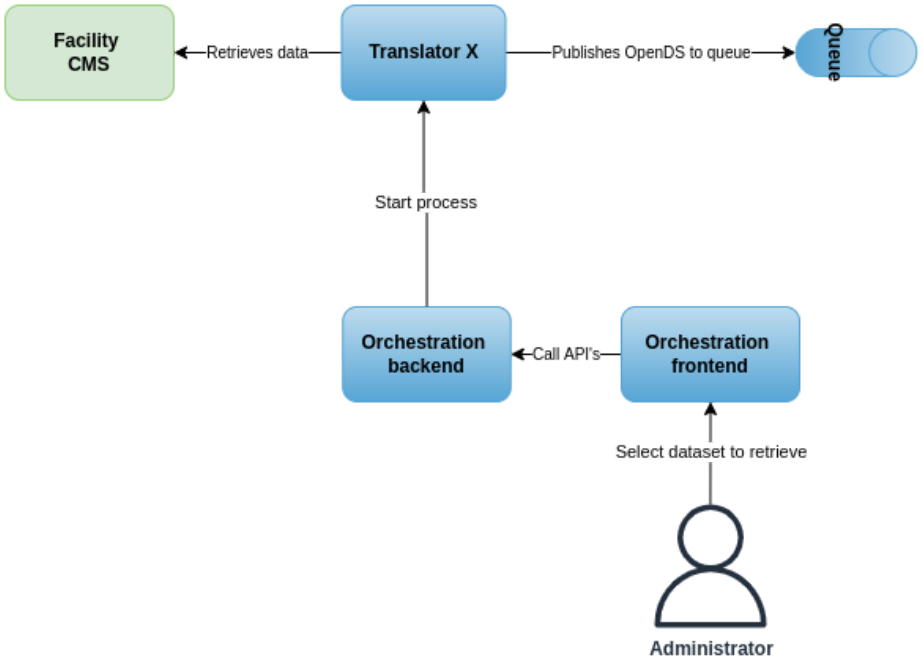


Fig. 3: Component diagram translator service

Since the result of the process should be a DS, in the rest of the DiSSCo core infrastructure all data exchanges will be in the openDS specimen format. The translators encapsulate all differences between the data sources and unites them in a generic data model. This enables us to use generic services in the rest of the DiSSCo infrastructure.

Integration with CMS based on events

In Task 6.1 a setup for deeper integration between the CMS and the DiSSCo core infrastructure has been described. This would enable the infrastructure only to act based on an event produced by the CMS. A detailed description can be found in the 6.1 Deliverable.[7] The DiSSCo core infrastructure will provide a translator which can trigger based on an event.

[7] D6.1 HARMONIZATION AND MIGRATION PLAN FOR THE INTEGRATION OF CMSs INTO T…

DiSSCo Data Processing container

The data processing container consists of services that handle inserting the data into the data storage container, after checking versioning conflicts and triggering enrichment services. At this point the data should already be a valid DS. The processing container currently consists of two services, the processing service and the data storage service. The data processing container will receive the highest load as each specimen change will flow through these services. This means that it should easily scale to the necessary throughput.

Processing service

Starting from the processing service the data is unified. The data that the processing service receives can come from different sources, for example from the translator services or the enrichment services.

The processing service will check if the data received needs to be created or updated. If the storage container already has the same specimen, (either based on DOI, other community accepted GUID or on the combination of institutionCode/collectionCode/physicalSpecimenId) data does not need to be updated. When the data is not yet available in the data storage container we need to create the DS. In the case the new data differs from the existing version we need to determine which part can and should be updated. Not all services will be able to change all parts of the DS. For example, the authoritative section of the Digital Specimen can only be updated by the authoritative source.

For new objects, the processing service might trigger enrichment services. In the orchestration frontend service, administrators can select a particular enrichment service they want to run on the newly created data.

The processing service is also tasked with validating the new or modified objects against the openDS specification. The validation checks if all required fields are filled and if the correct data types are used. This ensures that we only get consistent and reliable data in our data storage container.

When data has been checked and validated we can store the data in the data storage container by pushing it towards the digital object managing services. Handling any exceptions coming from persisting the Digital Specimen needs to be captured and if possible retried. When the processing service is unable to process the Digital Specimen we will add it to a dead letter queue from which we can manually check the issue and reschedule it if necessary. This ensures that no data is lost due to unexpected issues with either the software or the hardware.

After successfully adding data to the data storage layer we need to create provenance records of the actions. The processing service will also trigger creation of provenance records and events to external systems.

In summary, the main functions of the data processing service are: Read the DS from the queue and validate if the DS matches the JSON Schema present in the data storage

containers. Second, check if the DS is already present in the storage container and if not add the DS as a new object. If it is already present and there is new or updated information, update the DS. Third, if enrichment services need to be triggered, add the object to the enrichment service queue. Lastly persist to DS in the data storage container and trigger the creation of a provenance record.

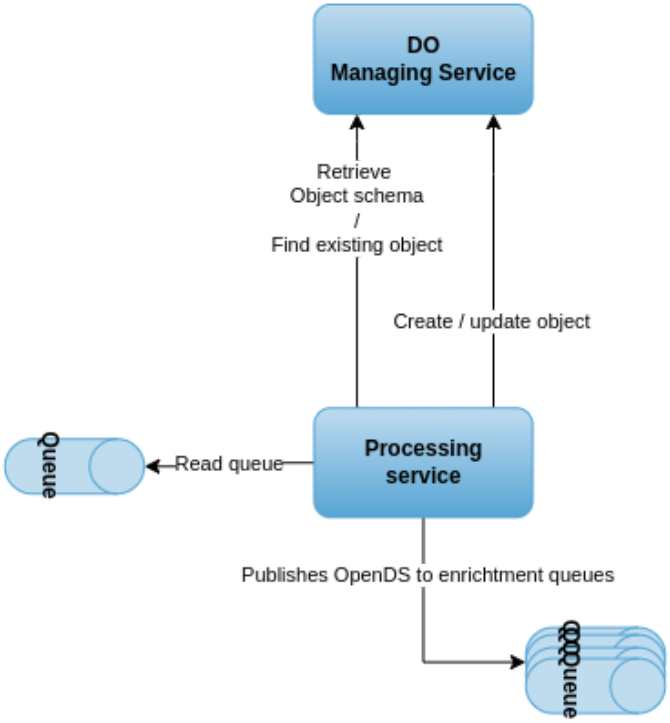


Fig. 4: Component diagram processing service.

DiSSCo Data Enrichment container

The DiSSCo Data enrichment container is a container with services which add new data to the DS. It consists of a variety of services all complying to a single rule. They have as input a DS plus (conditionally) more accompanying digital objects like images. Output is the enriched DS and, where appropriate, additional related annotation objects (Fig. 5).
What happens in between, whether it uses machine learning, manual input, community participation or anything else does not matter to the DiSSCo core infrastructure as long as they comply with the general rule DS in -> DS (plus annotation objects) out. In this way we can decouple the services from the rest of the infrastructure and provide a larger degree of freedom. A malfunctioning enrichment service will not have any impact on the rest of the DiSSCo infrastructure.

The enrichment services are triggered by a message on a queue. This message contains the DS. To help with the enrichment services the DiSSCo development team will create a

common library which will help with the retrieval, publishing and parsing of the messages. This way the development of the enrichment services will only need to focus on the addition of information.

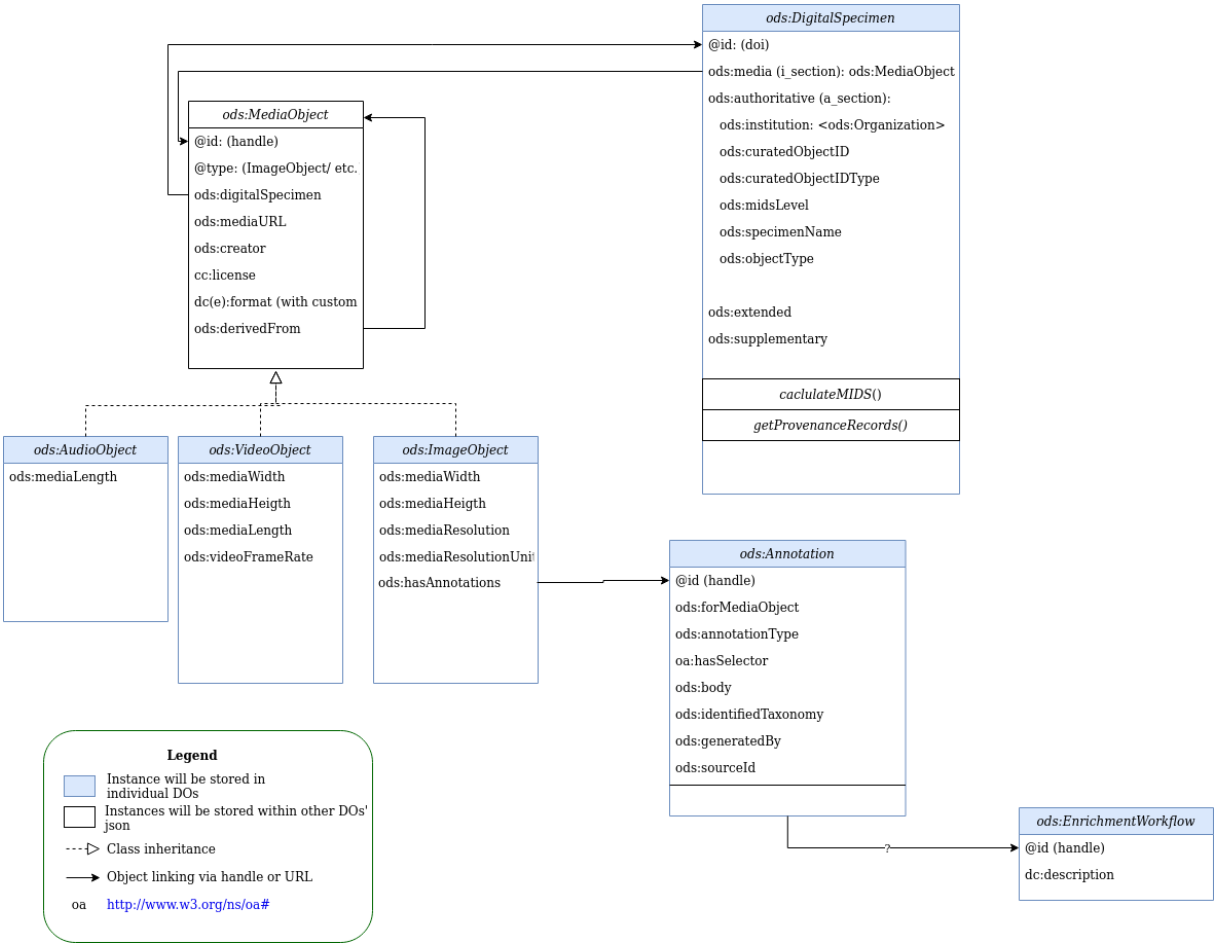


Fig. 5: Excerpt of the openDS class tree focussing on classes and properties relevant for annotations. Annotation or enrichment operations like the outlined trait feature extraction pipeline (Fig. 1) update either elements of the DS or trigger - in the context of subclasses of ods:MediaObject - the creation of one or more annotation objects.

The main functions of the enrichment services are: Collecting messages from the queue (library function), running the enrichment service, adding data to the DS and finally publishing the enriched DS as a message to the queue (library function).

The result of the enrichment services will be a DS with additional information. This DS will be published to the processing service queue (just as DS from the translator services) where it will be validated and stored in the data storage container.

Two exemplary enrichment services have been created as a proof-of-concept of this architecture. Both example services are enriching DS with images. The first enrichment

service reads the image binary (usually provided via a URL) and adds certain image metadata information to the digital object (eg. image size, mime type, etc.).

Beforehand, the processing service has to check whether the DS has an image and whether the data fields that the enrichment service can add do not exist already, and based on this information add the DS to the specific queue where this enrichment service is triggered.

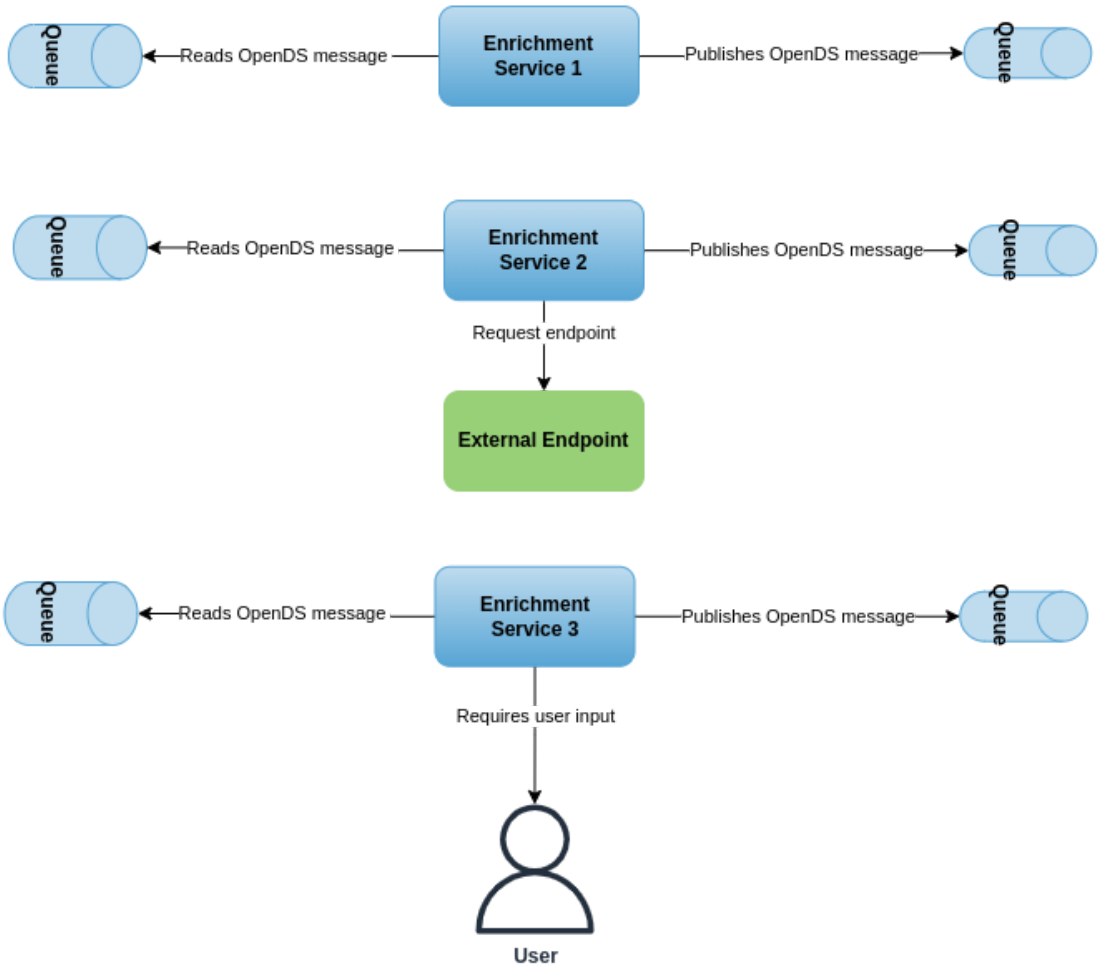


Fig. 6: Component diagram enrichment services. Three different types, from top to bottom. Internal enrichment service, enrichment service calling external endpoint, enrichment service requiring user input (human in the loop).

The second exemplary enrichment service runs a previously trained convolutional neural network created by Younis et al. (2020) on herbarium scans and detects 6 categories of plant organs on the image. The detected plant organ types are returned as annotations together with the DS to the queue. The handling of the annotations will most likely change over the course of the development of the Unified Curation and Annotation System (UCAS, see section below).

DiSSCo Data storage container

The data storage container is responsible to persistently store, index and retrieve data. Within the DiSSCo core infrastructure we will use existing Open Source projects to provide these functionalities.

DO managing service

The digital object managing service is responsible for coordinating the persistent storage and indexing. As the hearth of the system it is important that this application is always available. It should therefore be redundantly deployed to ensure that one instance is always up and running, even during deployments, just as availability of this component should also be horizontally scalable to handle a possible increased workload.

Persistent store

To store data for longer periods, a persistent store solution needs to be used. Both the metadata (digital object) and the data (images, sounds, etc.) may be stored.

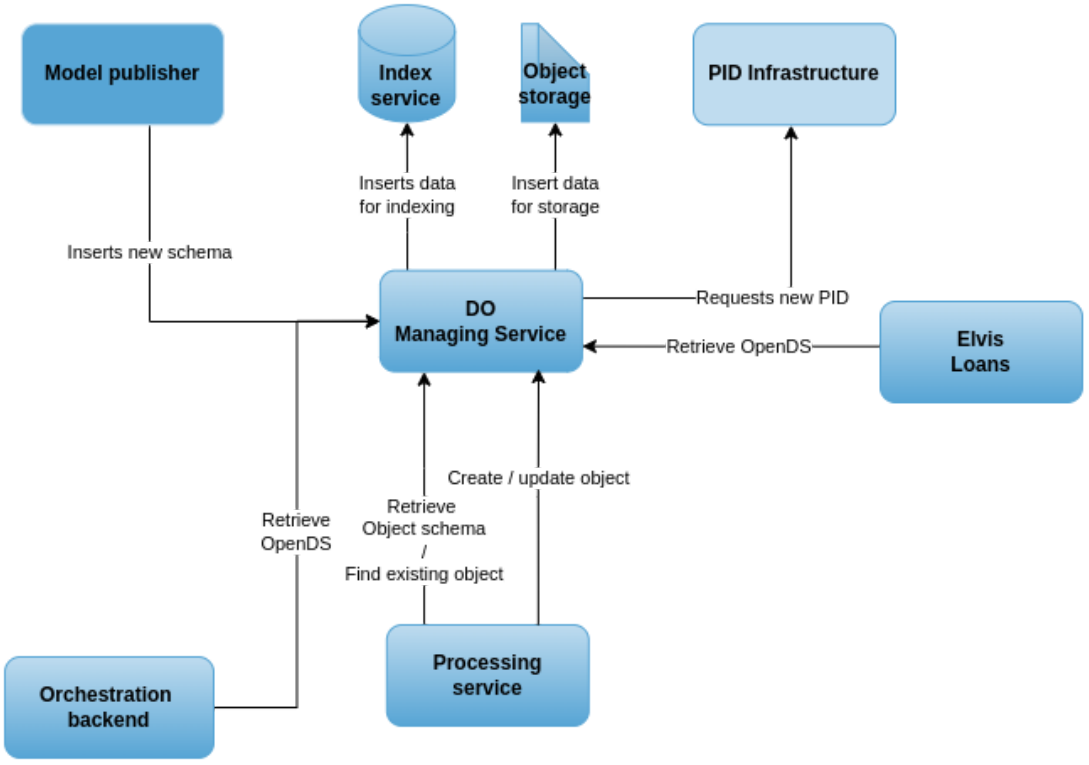


Fig. 7: Component diagram data storage.

Potentially the amount of data could grow to large dimensions which means that the storage solution needs to be able to grow with the data. Besides storage capacity the uses of the data will also grow, meaning the solution needs to be scalable based on the incoming requests.

Indexing solution

For rapid searchability and data retrieval a indexing solution is used. This will ensure that all the data and the fields are fully searchable. Even with large amounts of data, the time to find specific items based on their properties should be minimal. A query language should be available to run exact queries.

DiSSCo services

DiSSCo services will create a unique access point for integrated data analysis and interpretation through a wide array of tools. These services will retrieve data from the system and data access can be through different interfaces such as human activities, user interface (UI), machine interactions, or an application programming interface (API). These services and access points are containerised and can expand as new user requirements and access methods are requested.

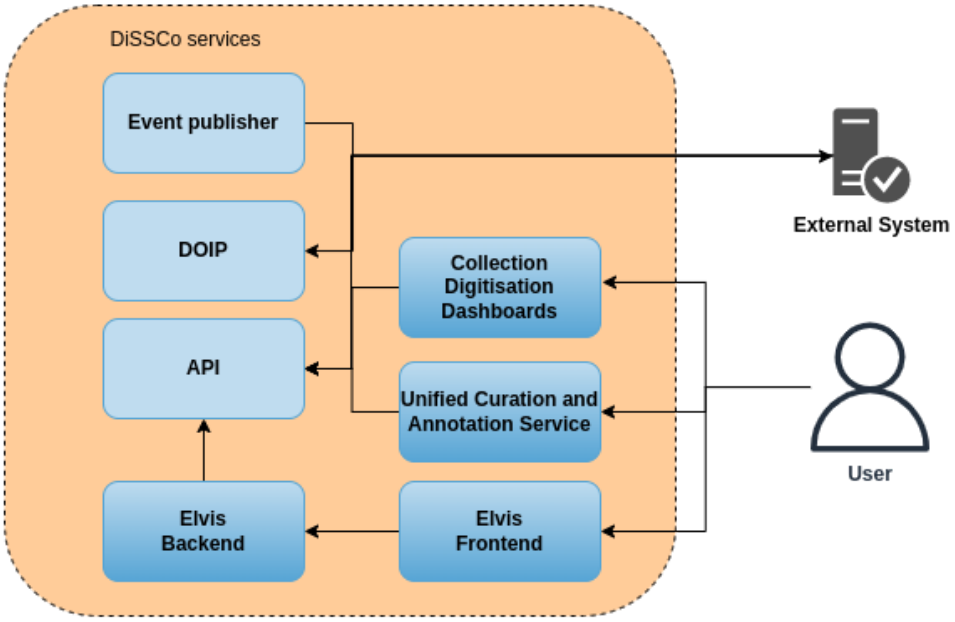


Fig. 8: Component diagram DiSSCo services.

Application Programming Interfaces (API)

API's are essential for data accessibility and usability. The DiSSCo core infrastructure will expose its data via an API. This API is used both internally, for DiSSCo's own data visualisation, as externally. This enables external users to access the data programmatically but also enables them to build their own data visualisation tools on top of the DiSSCo infrastructure.

The API will provide several endpoints, each based on the best practices coming from the JSON:API specification.[8] An important set of endpoints will be the retrieval of the data,

[8] https://jsonapi.org/

including bulk operations.[9] However as several end user services will provide additional annotations, the API will also have endpoints for data annotation or suggested modifications.[10]

European Loan and Visits System (ELViS)

ELViS is another end-user aimed service. It provides a unified way to request visits, loans and virtual access. Virtual access (VA) requests through ELViS provide digitisation on demand as a new type of access, including support for collaborating on VA ideas and proposal submission.

In line with other end-user services ELViS will also make use of the DiSSCo API. It will also have the capability to annotate a Digital Specimen. When an item is on loan to another organisation this information will be added to the Digital Specimen as an annotation. In this way ELViS provides new data to the data storage container. The new data will be published to the processing queue so it can be picked up by the processing service. For further information on Elvis see the Elvis system design.[11]

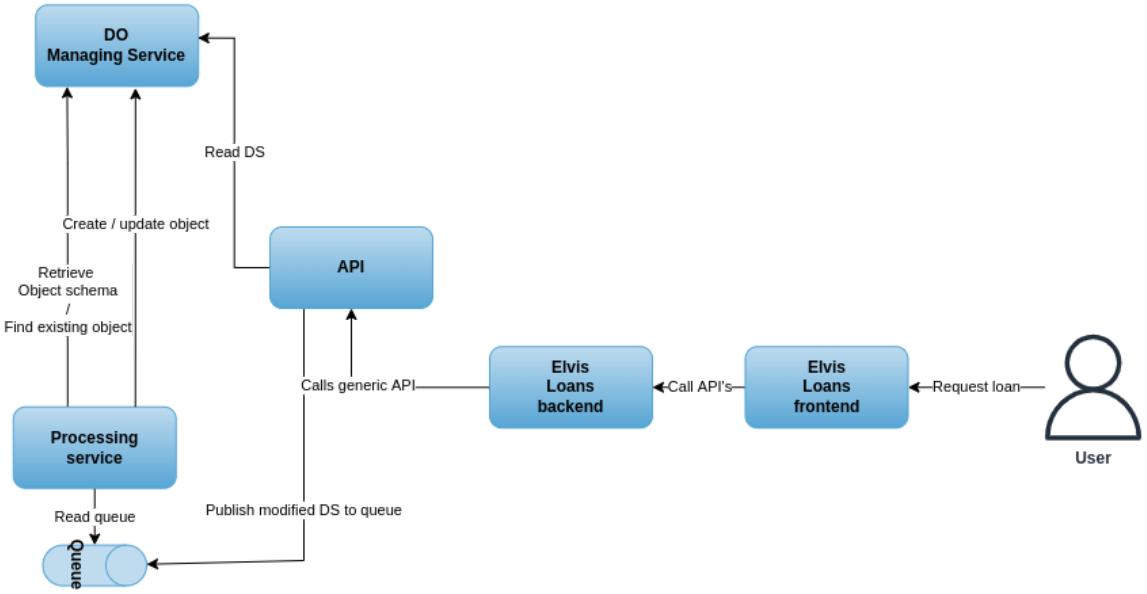


Fig. 9: Component diagram ELViS.

Unified Curation and Annotation System (UCAS)

UCAS is the main user interface on the DiSSCo data. It will enable the user to search and find specific Digital Specimens. For authenticated users it will also provide functionality aimed at curating and annotating the data. These will be sent to the authoritative party for

---

[9] https://github.com/DiSSCo/user-stories/issues/213: *"As a scientist I want to gather, compare, reuse data from individual small scale e.g. single species studies so that I can do meta-analyses and find large scale general patterns for this I need bulk data in a comparable format, a selective search and export tool, links to other databases e.g. GenBank, vegetation databases."*

[10] See some integration related user stories.

[11] ELViS design document

evaluation. UCAS is built on top of the DiSSCo API, using the API for data retrieval and modification actions.

Collection Digitisation Dashboards (CDD)

The collection digitisation dashboards are an example where collection information from the data storage container is retrieved and presented in a dashboard to a user.[12] The user can gain insights into the status of the collection and the level of digitisation based on MIDS.[13] When new data is ingested through the data ingestion container and added to the data storage container, the dashboards will be updated accordingly. The dashboards will only read data from the DiSSCo API which exposes DiSSCo's data. This data is then used to present aggregated data in the form of charts and tables.

Event Publisher

To notify external systems of changes in the digital objects we will publish events. These external systems can be CMS's which can update their information with the new information in DiSSCo. Together with the event based translator this will create synchronisation between the CMS and the DiSSCo infrastructure.

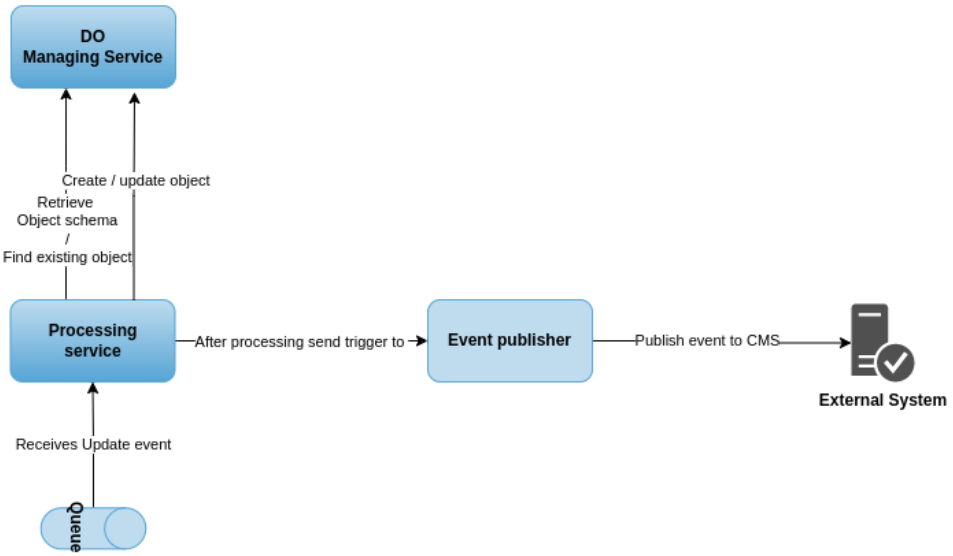


Fig. 10: Component diagram event publisher.

However, the functionality is not limited to the CMS's. Other external systems, such as data aggregators like GBIF and GeoCase, can subscribe to events in the DiSSCo infrastructure. This means they will be notified when information within the DiSSCo infrastructure changes and react accordingly.

[12] Currently two dashboards are running but not yet part of the infrastructure: https://rebrand.ly/synth-cdd https://icedig.eu/content/policy-analysis
[13] https://www.tdwg.org/community/cd/mids/

Digital Object Interface Protocol (DOIP)

The DiSSCo services will offer a DOIP interface for external systems wanting to use this protocol.[14] Users should be able to add information to a free section in the DS. They can use DOIP for retrieving and modifying DS. As we want to prevent open access to our data storage container we will have a separate DOIP interface handling this communication. All modified DS will be run through the data processing container before they are inserted into the data storage container.

DiSSCo Resolution system

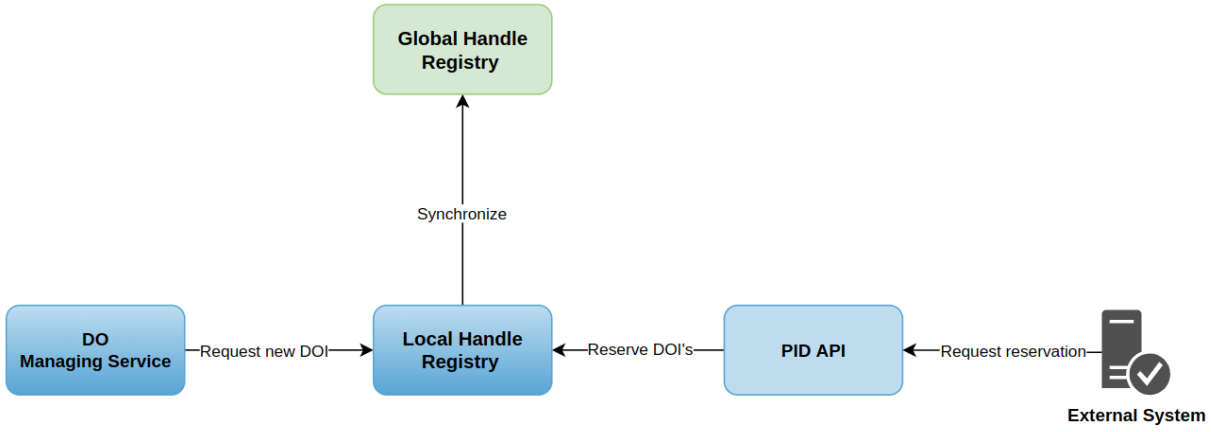
The resolution system is aimed at creating persistent identifiers for the digital objects within the DiSSCo infrastructure.[15] It is built based on the requirements provided by the DONA foundation.[16] It will consist of several components which will work together to manage, create and update the Digital Object Identifiers (DOI).

Local Handle Registry

The Local Handle Registry (LHS) will create, update and store the DOI's. This will be an instance of the local handle registry software provided by the Corporation for National Research Initiatives (CNRI).[17] The DO managing service will communicate with the LHS to register and update the handles. The Local Handle Registry access specifications, such as network address, will be recorded and kept current in the Global Handle Registry, which accordingly redirects clients to the appropriate LHS.

PID API

To enable users or external applications to request to mint a DOI we will create a PID API. This API will enforce authentication and authorization to ensure that the user or application has the correct rights to request to reserve or mint DOI's. The PID API will also validate and add the required metadata that will be necessary for registering the DOI.



[14] Digital Object Interface Protocol Specification v2.0
https://www.dona.net/sites/default/files/2018-11/DOIPv2Spec_1.pdf
[15] https://riojournal.com/article/67379/instance/6662391/
[16] https://www.dona.net/handle-system
[17] https://handle.net/

Fig. 11: Component diagram resolution system.

DiSSCo Modelling Framework

The DiSSCo Prepare Deliverable 5.2 is focused on the setup of a DiSSCo Modelling Framework (DMF). This framework aims to create a workspace in which coherent and documented data models can be created and reviewed. When a new data model has been finalised it will automatically be published as a Schema in the data storage container. Other applications needing the schema can retrieve it from the data storage container.

DiSSCo modelling Wikibase

The DiSSCo modelling Wikibase is the heart of the modelling framework. It is based on the Wikibase setup which provides tooling such as a user interface, authentication & authorisation management and persistent storage. For a complete overview see the deliverable regarding the modelling framework (Fichtmueller 2022).

Model publisher

Besides the Wikibase the modelling framework ensures that new versions of the data models are automatically published to the data storage container. This ensures that the rest of the system will be able to use the new models in the validation and creation of the Digital Objects.

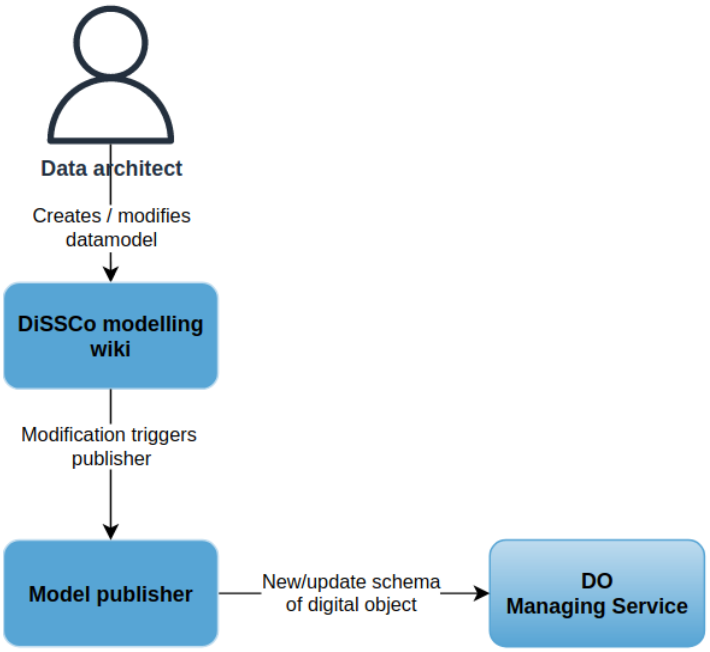


Fig. 12: Component diagram DiSSCo modelling framework.

DiSSCo Orchestration container

To orchestrate the pipelines for data retrieval and enrichment an orchestration container will be implemented. This container will ensure that other services will be triggered and their

process will be monitored. To enable persistent storage of state a storage solution will be used.

Orchestration frontend

The orchestration frontend will provide administrators with the ability to schedule translator and/or enrichment services. For example an administrator can select to translate a certain dataset by using an application which requests its API. For this a form will be filled in which it can map elements of the API response to the appropriate fields in the openDS data model. The administrator can also select to run one or more enrichment services over the retrieved data set.
For already existing data sets the data enrichment services can be scheduled for the complete or a part of the collection (based on query parameters). This ensures that when new enrichment services are developed these can be run on existing DS.
Besides triggering services, the user will also be able to view the progress of the running services and the status of the service. This way the administrator can review if the dataset was successfully retrieved and enrichment services are completed.

Orchestration backend

The orchestration backend is responsible for triggering the requested services (coming from the frontend). It has two main paths, the scheduling of translator services and the triggering of enrichment services.
For the scheduling of translator services the orchestration backend will create a new translator and monitor its progress. The translators are all running within the DiSSCo core infrastructure and therefore under the DiSSCo control. The translators will regularly report their progress back to the orchestration service (by publishing an event) so that the administrator can monitor the progress. The orchestrator will also monitor the container and regularly check if it is still running and active.

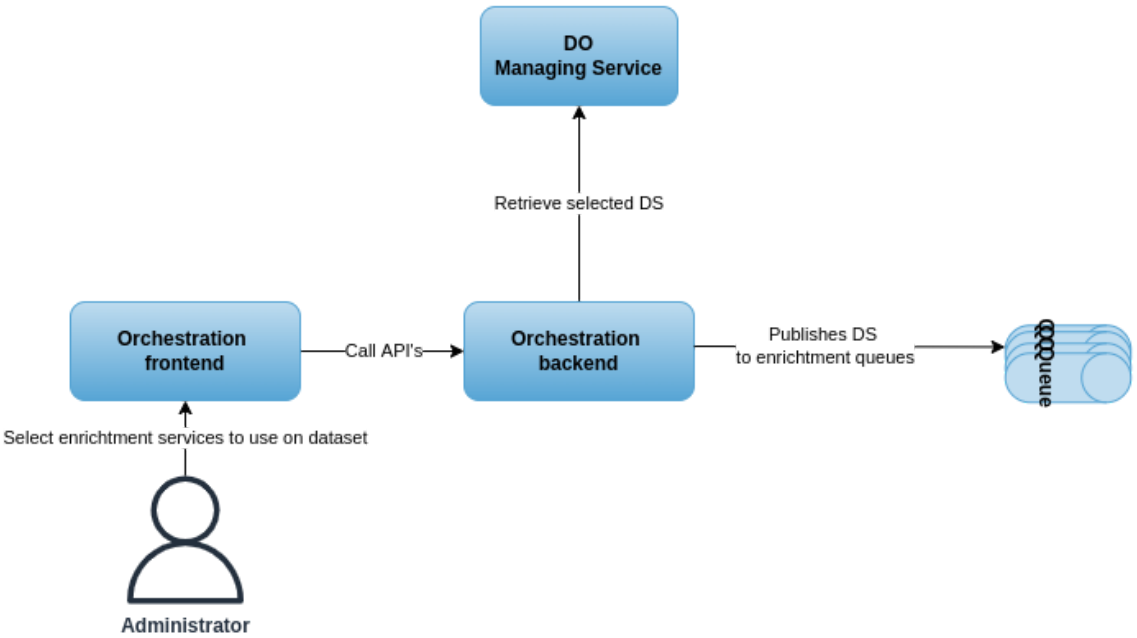


Fig. 13: Component diagram orchestration service, in specific the data enrichment trigger.

For the enrichment services the orchestrator will retrieve the requested DS and publish them to the relevant enrichment queues. The enrichment services are not in direct control and not necessarily part of the DiSSCo infrastructure. These services are more decoupled from the overall architecture. By retrieving the DS by the orchestrator (instead of the individual enrichment services) the enrichment services don't require any knowledge or connection to the data storage container. For the enrichment services retrieving the progress will be more difficult as it will be difficult for the enrichment services to report back. A solution for this will need to be devised.

Authentication and Authorization Infrastructure (AAI)

Authentication and authorization plays a key role for all services in the infrastructure. Authentication refers to the process of signing in as a registered user whereas authorization refers to the rights and permissions to perform certain actions a user has. From the user's perspective it is desired that a user can login into all DiSSCo services with the same credentials (single sign on, SSO). Another important feature is identity and access management (IAM) which means the management of a user's details and permissions within the system.

SSO and IAM provider

A single container provides both, SSO and IAM, for all other services in the DiSSCo infrastructure. Whenever a user is required to login in one of the end user services, they will be redirected to the login page of the SSO and IAM container. This can by itself make use of external SSO providers, so the user can choose to login for example with their eduGAIN, ORCID or Google account. After logging in for the first time an entry is created for every user in the DiSSCo SSO's internal database. When a user has logged in successfully, they get redirected to the website of the original service they requested, together with a signed JSON web token (JWT). The JWT is defined in RFC 7519 and is an encoded JSON object which holds the claims of the user[18] (e.g. user details, roles, issuer of the token).

The DiSSCo service that receives the JWT must previously be made aware of the public key of the issuing Keycloak server. The service uses the public key to verify the JWT and that it has not been tampered with. JWTs can be signed, encrypted or both. If it is only signed, anyone can decode the token and read its content. If the content is considered to be sensitive information, the JWT should be sent encrypted.

User roles and groups will be managed in the SSO and IAM provider container and included in the JWT which gets transferred to each end-user service. However the actual evaluation of authorization, e.g. "is user A (having role B) allowed to perform action C" will be done in each service, based on the roles and group memberships a user has. A good documentation of the user roles within DiSSCo and their associated permissions in each service is required.

[18] https://tools.ietf.org/html/rfc7519

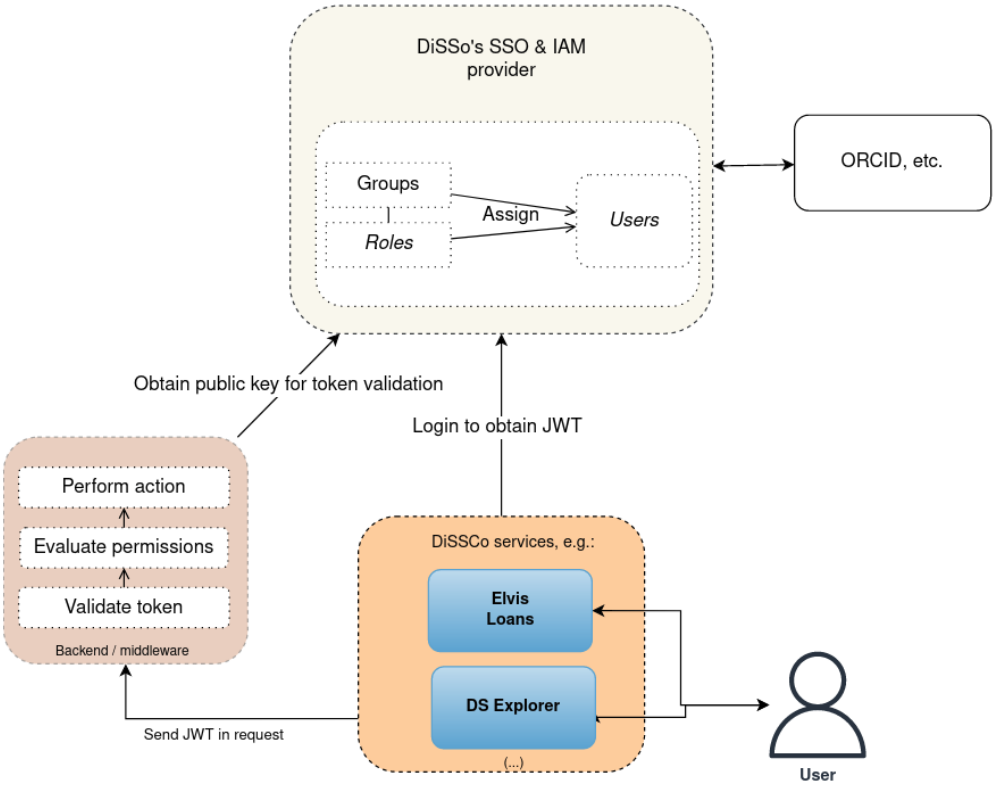


Fig. 14: AAI flow diagram.

Conclusion

The DiSSCo Core Infrastructure consists of several components working in close relation with each other. Each of these components has its own distinct task and goals. The glue between the services is formed by queues on which events can be published or consumed. By decoupling the different services into an event driven architecture we can easily scale with the amount of data that is generated. This scalability ensures that we can perform at peak moments, but also scale down on moments of quietness.

Preparing for the future is one of the main challenges. Developing an infrastructure which is still relevant decades from today is difficult, the future cannot be predicted. However by decoupling the applications we can easily replace a component in the ecosystem. Minimising downtime due to the stateless character of the services creates space for fast and early releases.

Through these principles, the DiSSCo Prepare Work Package 6 team has tried to identify a comprehensive architectural blueprint for the DiSSCo Research Infrastructure. Implementing the infrastructure in the coming years will provide further feedback on the choices made. Only practice makes perfect, changes should be promoted rather than prohibited. By using an agile mindset combined with an agile architecture we are ready to set up the DiSSCo Research Infrastructure.

Acronyms,terms, and definitions

Acronym	Term	Definition in the DiSSCO context
ABCD	Access to Biological Collection Data	Comprehensive standard for access to and exchange of data about specimens and observations.
ABCDEFG	Access to Biological Collection Databases Extended for Geosciences	ABCD extension that provides various terms relevant for geoscientific collection objects.
AC	Audubon Core	Set of vocabularies for the representation of metadata for biodiversity multimedia resources and collections.
BioCASe	Biological Collection Access Service	Meant in this context is the BioCASe Provider Software, a data binding middleware that allows publishing of multiple data resources of a provider with a single web service in ABCD format.
CD	Collection Descriptions	Data standard for describing collections of natural history materials including information about access and usage of specimens.
DOA	Digital Object Architecture	DOA introduces the concept of a digital object, which forms the basis for the architecture by specification of three key components: Identifier/resolution system, repository system, and registry system.
DS	Digital Specimen	Digital Twin of a physical specimen in a collection, encapsulates and persistently links to information artefacts derived from the physical specimen such as sequences, images and taxonomic determinations.
DMF	DiSSCo Modelling Framework	An instance of Wikibase where the DiSSCo data model is being developed.
DwC	Darwin Core	Framework of standards to compile and mobilize biodiversity data from varied and variable sources.
FDO	FAIR Digital Object	FDOs are abstracted data objects encapsulating content, descriptive metadata and globally resolvable and persistent identifiers in compliance with the FAIR principles.
GBIF	Global Biodiversity Information Facility	Networked data infrastructure funded by the world's governments aggregating data

		about all types of life on Earth.
ICEDIG	Innovation and Consolidation for Large Scale Digitisation of Natural Heritage	Project which provided essential blueprints and capacity enhancements to make DiSSCo operational with special emphasis on mass digitisation and subsequent access to all related data
IPT	GBIF Integrated Publishing Toolkit	A free, open source software tool used to publish and share biodiversity datasets through the GBIF network
JSON	Java Script Object Notation	Lightweight, text-based, language-independent interchange format for structured data.
JSON-LD	JSON for Linking Data	JSON-LD is a syntax to serialise Linked Data in JSON and can therefore be used as an RDF syntax.
JWT	JSON web token	An encoded JSON object which transports information about user authentication and roles.
LD	Linked Data	Set of methods to publish structured and connected data involving i.a. controlled vocabularies and ontologies to enable machine-interpretability of data.
MIDS	Minimum Information about a Digital Specimen	Specification defining information elements for graded digitization levels of physical specimens.
openDS	Open Digital Specimen	Specification providing the set of elements to model Digital Specimen (and other curated biodiversity objects) as typed data objects compliant with FDO specifications.
RDF	Resource Description Framework	Standard data model of the Semantic Web to model resources and statements about those.
OWL	Web Ontology Language	Decision Logic-based language enabling formal modelling of types of resources and their relationships.

References

De Smedt, K., Koureas, D. and Wittenburg, P., 2020. FAIR digital objects for science: from data pieces to actionable knowledge units. *Publications*, *8*(2), p.21. https://doi.org/10.3390/publications8020021

European Commission, Directorate-General for Research and Innovation, Turning FAIR into reality : final report and action plan from the European Commission expert group on FAIR data, Publications Office, (2018), https://data.europa.eu/doi/10.2777/1524

Fichtmueller D. & Güntsch A. (2022) DiSSCo Prepare Deliverable 5.2 "DiSSCo Modelling Framework". https://doi.org/10.34960/e3nv-zh69

Glöckler F, Pim Reis J, von Mering S, Petersen M, Weiland C, Dillen M, Leeflang S, Haston E, Addink W, Fichtmüller D (2021) DiSSCo Prepare report D6.1 Harmonization and migration plan for the integration of CMSs into the coherent DiSSCo Research Infrastructure - MfN WP6/T6.1. https://doi.org/10.34960/366d-sf49

Grieb J, Weiland C, Hardisty AR, Addink W, Islam S, Younis S, Schmidt M (2021) Machine Learning as a Service for DiSSCo's Digital Specimen Architecture. Biodiversity Information Science and Standards 5: e75634. https://doi.org/10.3897/biss.5.75634

Hardisty, AR. (2019). Provisional Data Management Plan for DiSSCo infrastructure. Deliverable D6.6. Helsinki: ICEDIG. http://doi.org/10.5281/zenodo.3532937

Hardisty AR, Lannom L, Koureas D, Addink W, Weiland C (2019b) 'The Last Mile': The registry behind the identifier. Biodiversity Information Science and Standards 3: e37034. https://doi.org/10.3897/biss.3.37034

Hardisty AR, Addink W, Glöckler F, Güntsch A, Islam S, Weiland C (2021) A choice of persistent identifier schemes for the Distributed System of Scientific Collections (DiSSCo). Research Ideas and Outcomes 7: e67379. https://doi.org/10.3897/rio.7.e67379

Hardisty A, Saarenmaa H, Casino A, Dillen M, Gödderz K, Groom Q, Hardy H, Koureas D, Nieva de la Hidalga A, Paul DL, Runnel V, Vermeersch X, van Walsum M, Willemse L (2020) Conceptual design blueprint for the DiSSCo digitization infrastructure - DELIVERABLE D8.1. Research Ideas and Outcomes 6: e54280. https://doi.org/10.3897/rio.6.e54280

Islam, S., Hardisty, A., Addink, W., Weiland, C. and Glöckler, F., 2020. Incorporating RDA outputs in the design of a European Research Infrastructure for natural science collections. Data Science Journal, 19(50), pp.1-14. https://doi.org/10.5334/dsj-2020-050

Kahn R, Wilensky R (2006) A framework for distributed digital object services. Int J Digit Libr 6(2): 115-123. https://doi.org/10.1007/s00799-005-0128-x

Wilkinson, M., Dumontier, M., Aalbersberg, I. . The FAIR Guiding Principles for scientific data management and stewardship. Sci Data 3, 160018 (2016). https://doi.org/10.1038/sdata.2016.18

Wittenburg, P, Anders, I, Blanchi, C, Buurman, M, Goble, C, Grieb, J, Hardisty, AR, Islam, S, Jejkal, T, Kálmán, T, Kirkpatrick, C, Lannom, L, Lauer, T, Manepalli, G, Peters-von Gehlen, K, Pfeil, A, Quick, R, van de Sanden, M, Schwardmann, U, Soiland-Reyes, S, Stotzka, R, Trautt, Z, Van Uytvanck, D, Weiland, C and Wieder, P 2022 FAIR Digital Object Demonstrators 2021. Zenodo. https://doi.org/10.5281/zenodo.5872645

Younis S, Schmidt M, Weiland C, Dressler S, Seeger B, Hickler T (2020) Detection and annotation of plant organs from digitised herbarium scans using deep learning. Biodiversity Data Journal 8: e57090. https://doi.org/10.3897/BDJ.8.e57090