ARPHA Preprints

Project Report

Author-formatted document posted on 30/04/2025

Published in a RIO article collection by decision of the collection editors.

DOI: https://doi.org/10.3897/arphapreprints.e157339

Digital Object Interface Protocol (DOIP) enabled Digital Object repository installation to store and provide digital specimen information

Soulaine Theocharides, **Sam Leeflang,** **Wouter Addink,** **Sharif Islam**

ARPHA Preprints *Author-formatted document posted on 30/04/2025.* DOI: https://doi.org/10.3897/arphapreprints.e157339

Digital Object Interface Protocol (DOIP) enabled Digital Object repository installation to store and provide digital specimen information

Deliverable D7.4

30 April 2024

Soulaine Theocharides, Sam Leeflang, Wouter Addink and Sharif Islam

[1]: *Naturalis Biodiversity Center, Leiden, Netherlands*

BiCIKL

BIODIVERSITY COMMUNITY INTEGRATED KNOWLEDGE LIBRARY

This project receives funding from the European Union's Horizon 2020 Research and Innovation Action under grant agreement No 101007492.

Start of the project:	May 2021
Duration:	36 months
	Prof. Lyubomir Penev Pensoft Publishers
Deliverable title:	Digital Object Interface Protocol [DOIP] enabled Digital Object repository installation to store and provide digital specimen information
Deliverable n°:	D7.4
Nature of the deliverable:	Other
Dissemination level:	Public
WP responsible:	WP7
Lead beneficiary:	Naturalis
Citation:	Theocharides, S., Leeflang, S., Islam, I., & Addink, W. (2024). *Digital Object repository installation to store and provide digital specimen information*. Deliverable D7.4. EU Horizon 2020 BiCIKL Project, Grant Agreement No 101007492.
Due date of deliverable:	Month 36
Actual submission date:	30 April 2024

Deliverable status:

Version	Status	Date	Author(s)
1.0	Draft for review	15 March 2024	Soulaine Theocharides, Sam Leeflang, Wouter Addink, Sharif Islam Naturalis
2.0	Review	24 April 2024	Joe Miller, GBIF & Quentin Groom, MeiseBG
3.0	Submission	29 April 2024	Naturalis

Table of contents

Summary

Biodiversity research relies on physical specimens stored in natural science collections, which serve as enduring reservoirs of data about organisms and their environments. However, these reservoirs remain siloed. The concept of Digital Specimen addresses the challenges posed by the vast amount of disconnected digital biodiversity data available today. The existing approach involves converting analogue records into digital replicas stored in local databases, leading to isolated and fragmented datasets that are difficult to integrate and utilise efficiently. The Digital Specimen aims to overcome this by establishing an interconnected network of digital objects on the Internet.

Digital Specimens are FAIR Digital Objects (FDOs), structured digital entities that adhere to the FAIR principles: Findable, Accessible, Interoperable, and Reusable. FDOs have the potential to enhance the accessibility and interoperability of data from natural science collections by providing unique identifiers, descriptive metadata, and defined operations. DiSSCo utilises the FDO framework to enhance the accessibility and interoperability of biodiversity research data from natural science collections. FDOs facilitate seamless data exchange by providing structured digital objects with unique identifiers, descriptive metadata, and defined operations. As part of making Digital Specimens FDOs, DiSSCO implemented FDO records, metadata records associated with a Persistent Identifier, which further enable machine actionability.

A Digital Object repository was developed for the purposes of storing and acting upon digital specimens. Three technological pillars compose the repository: a relational database stores the latest version of the digital specimen and is used for retrieving specimens by their identifier; an indexing solution provides full search capabilities on digital specimens; and a document store holds previous versions of a digital specimen for provenance purposes. There are three ways a user may interact with the digital object repository: a REST API; a user-friendly web portal; and a DOIP server.

To ingest data from multiple source systems, a harmonised data model was developed, called OpenDS. Built upon existing international standards like DarwinCore and ABCD, OpenDs accommodates complex structures necessary to capture information about multiple taxonomic identifications, events, agents, and relationships to other data sources. DiSSCo has decided to adapt the GBIF Unified Model (UM) for specimen data, ensuring interoperability and avoiding the development of potentially competing standards. By aligning with the GBIF UM, DiSSCo enhances interoperability with GBIF and promotes the establishment of a unified data modelling standard within the biodiversity community, facilitating seamless data exchange and integration with data aggregators like GBIF.

List of abbreviations

ABCD	Access to Biological Collections Data
API	Application Programming Interface
AWS	Amazon Web Services
BiCIKL	Biodiversity Community Integrated Knowledge Library
CDD	Collection Description Dashboards
DiSSCo	Distributed System of Scientific Collections
DO	Digital Object
DOI	Digital Object Identifier
DOIP	Digital Object Interface Protocol
EU	European Union
FAIR	Findable, Accessible, Interoperable, Reusable
FDO	FAIR Digital Object
GBIF	Global Biodiversity Information Facility
GeoCASe	Geoscience Collections Access Service
GUI	Graphical User Interface
JSON	JavaScript Object Notation
MIDS	Minimum Information about a Digital Specimen
openDS	open Digital Specimen (data specification)
REST	Representational State Transfer
SDK	Software Development Kit
TDWG	The acronym of the Biodiversity Information Standards organisation
TRL	Technical Readiness Level

1. Introduction

This deliverable describes the work done to develop the key components that support the FAIR Digital Object infrastructure of DiSSCo. It describes the technical implementation for task 7.4: providing specimen information as FAIR Digital Objects with input from tasks 7.1-7.3 and following the guidance given by the provisional DiSSCo RI Data Management Plan[1]. The document describes how the FDO paradigm was implemented for Digital Specimens including the required PID infrastructure, the technical implementation of the repository with storage and indexing, supported data models and the interfaces to interact with the repository: REST, DOIP and GUI interfaces. Digital Specimens are new units on the internet designed to act as a

[1] https://doi.org/10.5281/zenodo.3532937

container or 'bag of links' for all digital information known about the specimen so that it can become a digital surrogate or digital twin for the physical specimen object.

The repository is designed for the future operation of DiSSCo RI, to curate and provide access to interlinked and FAIR digital specimen data, including literature, genomic, biochemical and taxonomic information as well as digital specimen images, for the estimated 1.5 billion specimen available in natural science collections in Europe. The repository is designed for interaction at industrial scale with the specimen data and supports interaction by both machines and human experts through annotations. Machine interaction is supported by persistent identifiers and machine actionable metadata descriptions of the objects implemented, through FDO. The interaction capabilities are essential to speed up digitisation, enrich and enhance specimen data to enable direct linking with derived and related data available online and to improve specimen data supply to infrastructures such as GBIF, COL, ENA and GeoCase. This fully supports the BiCIKL vision of connecting infrastructures to enable researchers to access services across the biodiversity data lifecycle.

2. FAIR Digital Object implementation

The substantial digitization of natural history collections has opened up a wealth of data for biodiversity research (Hedrick et al. 2020). Nonetheless, ensuring this data is readily accessible, usable, and interoperable across various platforms and institutions remains a challenge (Hardisty et al. 2022). To tackle this issue, DiSSCo leverages the FAIR Digital Object (FDO) framework to apply the FAIR principles (Findable, Accessible, Interoperable, Reusable) to data from natural science collections (Islam et al. 2023). Similar to how TCP/IP protocols enabled the internet to connect and exchange information, FDOs aim to facilitate seamless access and sharing of biodiversity research data.

The core principle of FDOs is the concept of a “digital object” - a structured collection of data with a unique and persistent identifier, descriptive metadata, and defined operations. These digital objects can represent various entities such as digital specimens, media files, or agents. By adhering to the FAIR principles, FDOs ensure data is easily discoverable, accessible, and usable by different platforms and applications.

In DiSSCo, a Digital Specimen is an FDO acting as a surrogate for a physical specimen online, enabling consistent identification, description, and integration with various services like machine learning workflows. Related information, such as media or genetic sequences, remains stored at the hosting institution and is linked to the Digital Specimen through a unique identifier; DiSSCo is responsible only for storing metadata about these artefacts, though these metadata entries may also themselves be considered FDOs. DiSSCo's design aims to strike a balance between flexibility and standardisation. While accommodating diverse specimen types (e.g., marine, botanical, mineral), it also provides structured descriptions crucial for integration with different tools and workflows.

One of the core metadata elements in the FDO framework is FDO records. FDO records, aside from identifying the object's location, contain structured metadata describing its attributes and characteristics. These records resemble PID records as recommended by the Research Data Alliance but emphasise the possibility of FDO implementations without relying solely on PID information. Building upon these records, FDO profiles further standardise the metadata

required for different object types within DiSSCo. This ensures consistency and simplifies management while allowing for specific attributes relevant to each type (e.g., specimen host for Digital Specimens).

In addition to Digital Specimens, DiSSCo has developed FDO Profiles for Media Objects, Annotations, Source Systems, and other digital objects used in the infrastructure. DiSSCo actively solicits community feedback on designing its FDO profiles, such as the one for Digital Specimens, through RFC (Request for Comments) documents. This fosters collaboration and ensures FDOs effectively address the needs of the biodiversity research community.

3. Repository technical implementation

One of the aims of DiSSCo is to provide functionality to search, curate, and link specimen data and build integrated services on top of the data. To achieve this goal, metadata about the specimen and its associated objects need to be gathered and stored. This storage will happen in the Digital Object Repository of DiSSCo. This chapter describes the technical implementation of this repository, which followed the implementation and construction plan of the DiSSCo core architecture (Leeflang et al. 2022) as developed in the DiSSCo Prepare project.

We will start by describing the data characteristics. Knowledge about the nature, volume, growth of the data and balance between reads and writes is essential in developing a technical implementation. In addition to exploring the data characteristics, we will also look into five of the main use cases of DiSSCo. These use cases are based on user stories gathered during Synthesys+ and DiSSCo Prepare project[2] (Fitzgerald et al. 2021).

Based on the data characteristics and main use cases, we will walk through a set of possible technical implementations. Each will be evaluated for suitability, and its benefits and drawbacks will be summarised. This will move us towards a technical implementation, which has been implemented and piloted during the BiCIKL project. It leverages the strengths of the technical solution to fulfil the main use cases while keeping an eye on the data characteristics.

3.1. Data Characteristics

During the project, we identified three key data characteristics impacting DiSSCo data architecture. Insight into these three characteristics is vital in making a calculated design decision. This does not mean that new insights during the development of DiSSCo cannot change the implementation phase. As with all systems, DiSSCo needs to be adaptable to changing needs and shifting priorities.

3.1.1. Data Volume

Estimating the volume of data gives us insight into the infrastructural needs and performance of the data architecture. However, fixed numbers for the amount of digitised collection objects

[2] https://github.com/DiSSCo/user-stories

ARPHA Preprints Author-formatted document posted on 30/04/2025. DOI: https://doi.org/10.3897/arphapreprints.e157339

are difficult to obtain. A good starting point would be the current volume of data available through data aggregators, such as GBIF and GeoCASe. If we combine these numbers for DiSSCo associated organisations the number lies around the 46 million digitised specimens[3]. However, not all digitised specimens have been made available through data aggregators. In the DiSSCo Collection Description Dashboard (CDD) of Synthesys+ it was estimated that there are around 74 million digitised specimens.[4] This number is a rough estimation, as only the larger organisations have been included in this.

The amount of specimens still waiting to be digitised is far greater than the current numbers. If we look at the CDD we see that the total number of specimens will be in the range between 282 (only large organisations) to 555 million (includes more organisations).[5] Hardisty provides further information in the DiSSCo Provisional Data Management Plan (Hardisty, A. 2019) and comes to the number of 1.5 billion specimens.[6]

If we generalise the statement about the data volume, we can conclude that at the start of DiSSCo we can expect roughly between 50–75 million objects. This will grow to around 1–2 billion specimens. The next paragraph will focus on the expected data velocity at which we can expect these newly digitised objects.

3.1.2. Data Velocity

The expected total data volume of 1-2 billion specimens will not be available from the start of DiSSCo.[7] As digitisation efforts proceed, the amount of digitised objects will grow. Predicting the speed of digitisation will depend on different factors, which are difficult to determine. Regardless, it is clear that the data velocity will not be linear. DiSSCo will (initially) not receive new data real time or near real time. Most data providers will provide the data in datasets. New datasets can be added, containing thousands or even millions of new specimens. Existing datasets can also be updated by removing or adding specimens.

A rough estimate for the data velocity can be gleaned by looking at the growth within GBIF (Figure 1). This shows that the growth globally of online published specimen records is relatively stable at 12 million new records a year. If we added geological specimens to the equation, we could expect an average data velocity of around 15 million new digital specimens per year globally or 5 million for Europe, which hosts about ⅓ of the specimen collections.

---

3 https://www.gbif.org/occurrence/search?basis_of_record=PRESERVED_SPECIMEN&basis_of_record=FOSSIL_SPECIMEN&basis_of_record=LIVING_SPECIMEN&basis_of_record=MATERIAL_SAMPLE&advanced=1&network_key=17abcf75-2f1e-46dd-bf75-a5b21dd02655 and for https://geocase.eu/search

4 https://app.powerbi.com/view?r=eyJrIjoiM2QyNDlmNjEtOTk3Ni00NGYwLThiMDctZTA5MjY0Mjc0MjUwIiwidCI6IjczYTI5YzAxLTRlNzgtNDM3Zi1hMGQ0LWM4NTUzZTE5NjBjMSIsImMiOjh9

5 https://app.powerbi.com/view?r=eyJrIjoiM2QyNDlmNjEtOTk3Ni00NGYwLThiMDctZTA5MjY0Mjc0MjUwIiwidCI6IjczYTI5YzAxLTRlNzgtNDM3Zi1hMGQ0LWM4NTUzZTE5NjBjMSIsImMiOjh9
https://app.powerbi.com/view?r=eyJrIjoiNTVlYjM2ZWUtNjI4Ny00MDA5LWEzNTgtOTgzOWI1N2Q1NmNhIiwidCI6IjhjZDI0OTg0LTBhYTMtNGZjNS1iMDliLTRkNmVjZmFhNThmYiIsImMiOjI9

6 Hardisty A, 2019 p.23 + Appendix D

7 Hardisty A, 2019 p.23-24

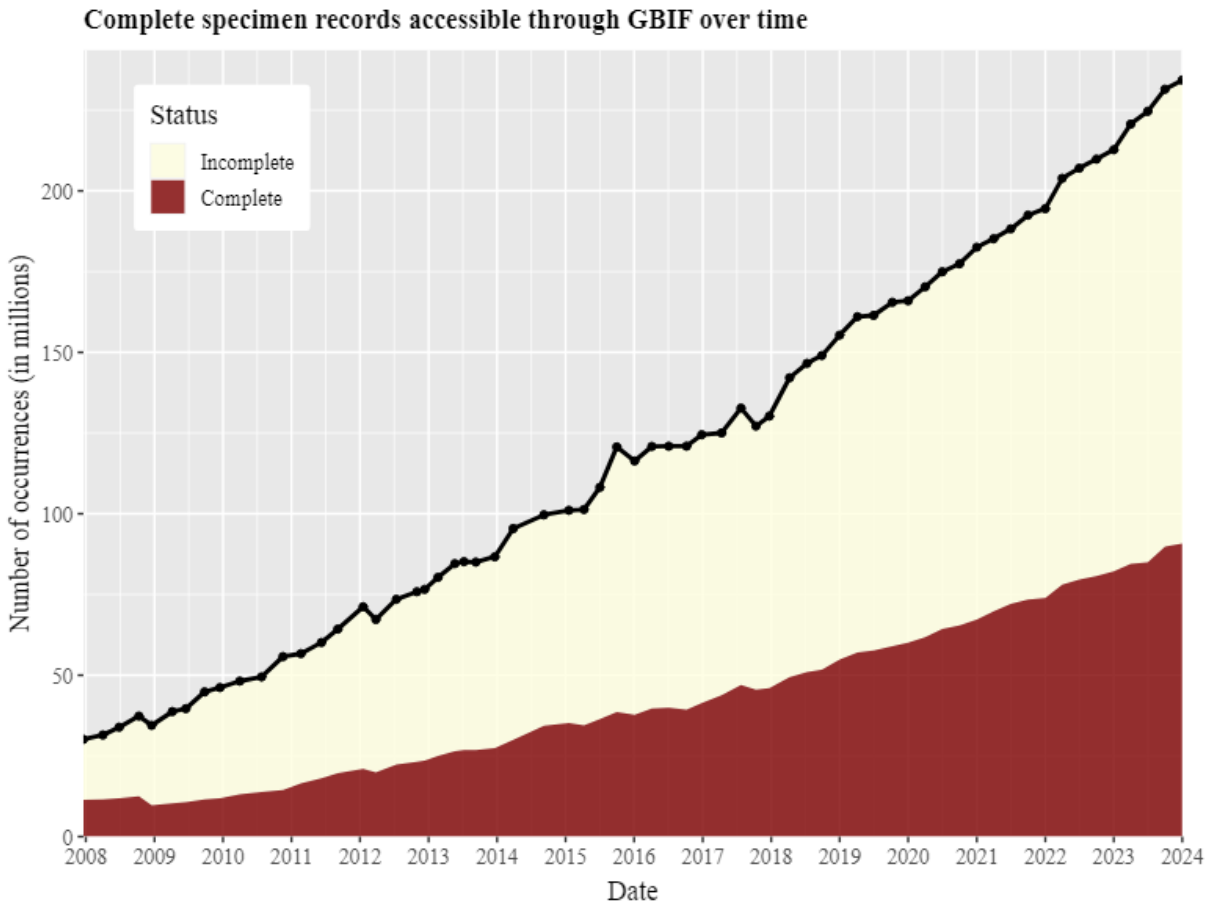

Figure 1. Growth in specimen records in GBIF over time.
Source: https://www.gbif.org/analytics/global

3.1.3. Data Variety

The data that DiSSCo receives is highly varied. Within the ingestion process, DiSSCo will harmonise all data into the open Digital Specimen (openDS) data specification. A detailed approach about the specification will be given in chapter five of this deliverable.

This means that all data will be standardised as much as possible into a single specification. However, the completeness and complexity of the data will vary greatly. openDS is a flexible model in which most possibilities are optional. This ensures that we can handle simple specimens with little information available, as well as large complex specimens incorporating several smaller specimens.

The complexity of the data specification influences the choice in the data architecture. As will be discussed in the next section, the balance between schema-upon-read and schema-upon-write is very important in DiSSCo.

3.2. Design Choices

3.2.1. Data Discoverability

One of the most critical features of DiSSCo is the ability to find a specific digital object or group of digital objects. DiSSCo will provide data discoverability functionality, a way to search and find specimen data. This means DiSSCo will fully index each object. This will provide DiSSCo with the functionality to both search for individual objects on particular fields and provide aggregated results on each field.

3.2.2. Automatic and Manual Data Annotations

Annotating data is one of the key features of DiSSCo. An annotation is a supplement to the digital specimen and is stored as a separate digital object, linked via the specimen's identifier. Annotations may come from human or machine actors. A human expert may manually add comments, corrections, or supplements to the data, while machine services may automatically provide additional curation. "Accepting" annotations, meaning to reconcile the additional information back into the specimen record, is to be subject to a trust model that is in development at the time of writing, and is beyond the scope of this document.

To avoid duplicate annotations, the creator, target, and content of incoming annotations are hashed and compared against existing annotations. Duplicates are ignored, and new annotations are processed, with the hash being stored alongside the annotation. The entire annotation is not hashed because some fields, such as timestamps, will likely always be unique, even if the incoming annotation does not contain new information.

By automating otherwise manual actions, we can speed up the digitization process, improve and enrich the data. Within the DiSSCo core infrastructure, this will be done by a range of machine annotation services (MAS). These services will receive a digital object, run some action over it, and provide the newly generated information back as an annotation. As an example for digitization, MASs can extract/create a region of interest for the label and do the text recognition on it, generate segmentation images or do species identification.

We expect that these machine agents will publish numerous annotations on the digital objects. Each annotation is seen as a separate digital object which needs to receive a PID and be stored and indexed into the digital object repository. Fast and scalable storage solutions will be essential for successfully fulfilling this use case.

In addition to facilitating automatic annotations, DiSSCo must support users' manual curation of digital specimens, multimedia objects or annotations. Curating, extending and linking the digital objects is the core functionality of DiSSCo. In addition to machines, the user needs to be able to take action on the digital objects. Users should be able to propose changes to the digital object or add new information to it. They need to be able to work together online on the same specimen.

As with the machine annotation services, new information will initially be captured as an annotation object. This object will receive a handle, and is stored and indexed in the digital specimen repository. Once the object has been created a CreateUpdateDelete event will be

ARPHA Preprints Author-formatted document posted on 30/04/2025. DOI: https://doi.org/10.3897/arphapreprints.e157339

triggered.[8] This event will be caught by DiSSCo's provenance service to store a first version of the object. It will also be possible to publish this event to any other consumers. A consumer could be a data pusher which publishes the newly created annotation object to a Content Management System. This way, we could notify data managers that an object in their dataset has received an annotation. More discussion is needed with CMS providers to see what the possibilities are.

3.2.3. Provenance

Provenance should capture who, when, and what changes were made to a digital object, providing users with the information on how the object came into being and what the trustworthiness of the information is. It also allows users to see change through time, providing a measurement for the digitisation effort.

To make sure each change in a digital object is captured and traceable, the digital object repository will store versions of each object. Each change in a digital object will generate a new version of the object. Both the new version and the difference with the old version will be stored. This way it will always be possible to see when, who and what information of the digital object changed between versions.

Not only DiSSCo can store new versions of the Digital Objects. Through the event-driven setup we are able to publish each change, including the creation/tombstoning, to external parties. DiSSCo can notify these parties of the change or that a change we made to a particular object. This way interested parties, such as the original data supplier, do not need to have to harvest the DiSSCo API but can be actively notified of the changes. An event-based approach has the advantages that they don't need to check unchanged information but can focus their resources on handling actual changed data. This idea was positively received by several major CMS providers.

3.2.4. Media

While related media are not stored within DiSSCo and remain with the hosted institution, a "digital entity" object is created for each media, containing metadata information about the media object, such as format or licence. This digital entity record is stored in the relational database and is linked to the related specimen via its identifier. The digital entity is also stored in the document storage and is subject to the same provenance as digital specimens.

3.3. Storage Implementation

In the previous section, we looked at the data characteristics and the main use cases. In this section, we will present the implementation of DiSSCo storage layer and discuss how all components function together. Each has a specific role to play in the fulfilment of the use cases mentioned above.

[8] See for more information Glöckler et al. 2022.

3.3.1. Relational database

A relational database is part of the storage implementation. In the relational database, we have a copy of the latest version of each digital object. This is used for quickly retrieving the objects as well as indicating relationships between objects. For each digital object, there is a row in the database, using the PID as primary key.

Within DiSSCo, we do not denormalize the objects, as this would create a complex and large set of tables. The storage might be more efficient but recreating an object from this multitude of tables will be difficult. We see the digital object as a self-contained object, from which we only normalise fields which are essential for creating relationships to other objects within the digital object repository.

The majority of the data will be stored in unstructured columns. This hybrid variant between a relational database and an object store provides us with a lot of flexibility. The relational database will be the backbone of the digital object repository without forcing us into strict database schema management. The data model can change significantly without requiring any database schema changes.

The relational database ensures persistent storage of the data and covers data queries which query directly on the PID of the digital object. It stores the relationships between the digital objects in the digital object repository and is able to combine them when a query requests this relationship. However, by not denormalizing the full object, querying on other fields than the PID is tricky. This is where the indexing solution comes into play.

For the implementation, we used PostgreSQL as a solution for our relational database. PostgreSQL is an open source database which provides an extensive set of features and is supported by a broad community. One of the features especially useful for DiSSCo is the JSONB data type, allowing the hybrid usage of the relational database. As the database is the core storage solution, we want to ensure that there is no data loss. Therefore, we use an AWS managed Amazon Relational Database Service (RDS). This means that AWS is responsible for providing sufficient uptime and regular backups for the database.

3.3.2. Indexing solution

The indexing solution provides full search capabilities for the complete digital object. For each digital object type, an index in an indexing solution is created. This index contains all attributes of the object. Each time a record is added to the relational database, it is also indexed by the indexing solution.

By indexing each field in the digital object, we can do complex searches and aggregations. Data queries other than directly on the PID are handled by the indexing solution. In addition to full match queries, the indexing solution also provides possibilities for fuzzy searches and prefix searches. These can become important when we need to make suggestions to the user based on their input and create auto-complete functionality.

Besides providing ample possibilities for searches, the indexing solution also provides functionality for aggregations. Aggregations are vital for providing good discovery mechanisms, such as filtering. It will also help in generating a dashboard regarding the state of the digitisation

effort, providing the data for the Collection Description Dashboards (CDD), one of DiSSCo's services (Tilley et al. 2024).

DiSSCo uses Elasticsearch as its indexing solution. Elasticsearch is an industry standard and provides highly performant scalable solutions. DiSSCo will run as a high availability distributed solution on their cluster. As Elasticsearch is not aimed at persistently storing the data, data loss is acceptable and can be remedied by synchronisation with the relational database.

3.3.3. Document storage

For almost all use cases, the combination of the above two storage solutions is sufficient. However, both have one limitation: they do not scale sufficiently to handle all versions of a digital object. On each digital object change, a new version is created. This means that we could have tens or even hundreds of versions of a single digital object. For a relational database, the initial volume of digital objects may be feasible. However, it does not scale sufficiently when this number is ten- or hundred folded. Next to scaling issues, the complexity of the queries when multiple versions of the object exist in the same tables or index would slow performance and complicate development.

This is why we decided to create a separate storage solution for historical versions of the digital objects. This storage solution should be very scalable, as the number of objects would quickly accumulate to billions. We require little query capacities from it, as we will only need to perform searches on the PID and version.

With these requirements, we looked into document storage solutions. This group of No-SQL storage solutions scales horizontally, so it can efficiently handle the growing amount of versions. It still provides query functionality, which we use minimally and can persistently store the data. We use the PID as the key for the document and store the full JSON of the digital object as the value.

To keep performance in the storage of the digital specimen, we implemented the concept of eventual consistency for the historical storage. This means that the storage of the historical version is not synchronised with the storage of the new version. When a new version is created, the system will trigger a change event which an event consumer will pick up. This consumer stores the record in the document storage solution. There is a possibility that the new version might be available before the old version is persisted.

As implementation of the document storage solution, we use Amazon's DocumentDB service. This MongoDB compliant service provides a fast, scalable, highly available, and fully managed document database service. This should cover our needs for keeping historic records of the data, fulfilling our needs surrounding provenance and traceability.

3.4. PID Infrastructure

Persistent Identifier (PID) systems are the foundation for achieving the FAIR Guiding Principles[9]. As FAIR data and connecting different data classes (i.e. specimens, molecular sequences,

[9] https://doi.org/10.1038/sdata.2016.18

observations, taxonomy and publications) are essential aspects of the BiCIKL project, A global PID system was developed to create and maintain identifiers for the digital representation of specimens and samples. The PID system provides the mechanism to ensure that identifiers are globally unique, persistent and resolvable.

Persistent Identifiers (PIDs) are globally unique identifiers of objects, providing a stable reference if the referent is relocated or undergoes changes. By ensuring persistence, PIDs eliminate broken links and guarantee the accessibility of digital resources, regardless of any future changes in storage infrastructure or hosting platforms. Every specimen ingested into DiSSCo is assigned a PID.

A PID can be thought of as containing two components: a PID name, a unique string that identifies the resource, and a PID record, an extensible record that contains the location of the resource. The PID record may be extended to become a FDO record, which are PID records with structured metadata. This metadata may include information such as title, creator, date, identifiers for related objects, access rights, and more. This information allows machines to make decisions regarding the Digital Object without needing to resolve the PID.

Different Types of Digital Objects have different FDO Record metadata requirements, and thus the actions a machine can take on a PID record is defined by the object Type. FDO Profiles standardise which FDO record attributes should be associated with each Type of object. Within DiSSCo alone, we expect to assign PIDs to a diverse array of object Types, including media objects, annotations, and of course, Digital Specimens. Each of these objects Types have their own FDO Profile[10].

Providing metadata-rich FDO records to all Digital Specimens requires a robust, scalable, reliable PID infrastructures. This section gives an overview of the PID infrastructure developed under this project. First, we give an overview of the resolution system chosen for this infrastructure; then, we discuss the infrastructure developed to facilitate creating FDO records at scale; finally, we discuss the next steps for specimen PIDs: DOIs.

3.4.1. Handle Resolution System

The Handle Resolution System is a global, distributed infrastructure developed by Corporation for National Research Initiatives (CNRI) for minting and managing Persistent Identifiers called Handles. As a distributed system, each organisation that mints Handles is responsible for its own Local Handle Server (LHS), distinguished by a unique prefix. Handle names must be unique within a LHS.

The Global Handle Registry registers the IP addresses of each LHS, and redirects PID resolution requests to the appropriate one, based on the prefix of the PID. The LHS then retrieves the PID record based on the locally unique suffix of the PID, and redirects the user to the location stored in the PID record (Figure 2).

[10] https://schemas.dissco.tech/schemas/fdo-profiles/0.1.0/

ARPHA Preprints Author-formatted document posted on 30/04/2025. DOI: https://doi.org/10.3897/arphapreprints.e157339

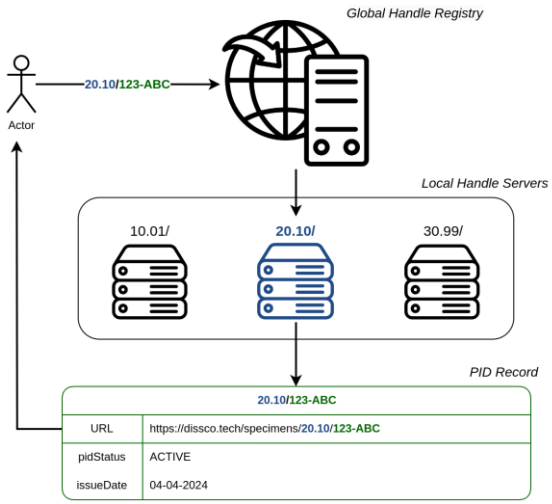

Figure 2[11]. The distributed architecture of the Handle system. When a resolution request is made, the Global Handle Registry, managed by the Corporation for National Research Initiatives, redirects it to the appropriate, institutionally managed LHS based on the prefix of the Handle. The Local Handle Server then identifies the correct record based on the suffix of the Handle.

3.4.2. Custom Handle Infrastructure

Though CNRI provides an API for LHS managers to mint and manage Handles one at a time, this is not an appropriate tool to mint Handles at scale. Instead, the LHS for DiSSCo was modified to read from a PostgreSQL database. The contents of the database can be modified more easily, while still being globally resolvable.

In using custom database storage for the DiSSCo LHS, it was possible to develop a REST API that interacted directly with the Handle database. The custom Handle Manager API[12] writes to the Handle database in batches of up to 1000, with a throughput of 300 Handles per second. The Handle Manager can mint Handles for several different Types of objects, such as annotations, media objects, and of course, Digital Specimens. The Handle Manager is responsible for building and publishing the FDO records based on the Type of the object and the FDO profiles designed by DISSCo.

When a specimen is being created, the ingestion process (called Specimen Processing) calls the Handle API, which writes the specimen's FDO record to the Handle Database. The Handle API then returns the newly minted PID to the Specimen Processor, which uses the PID throughout the rest of the ingestion pipeline (Figure 3).

[11] Web server icons created by Uniconlabs - Flaticon; Server icons created by Pixel perfect - Flaticon
[12] https://github.com/DiSSCo/handle-manager

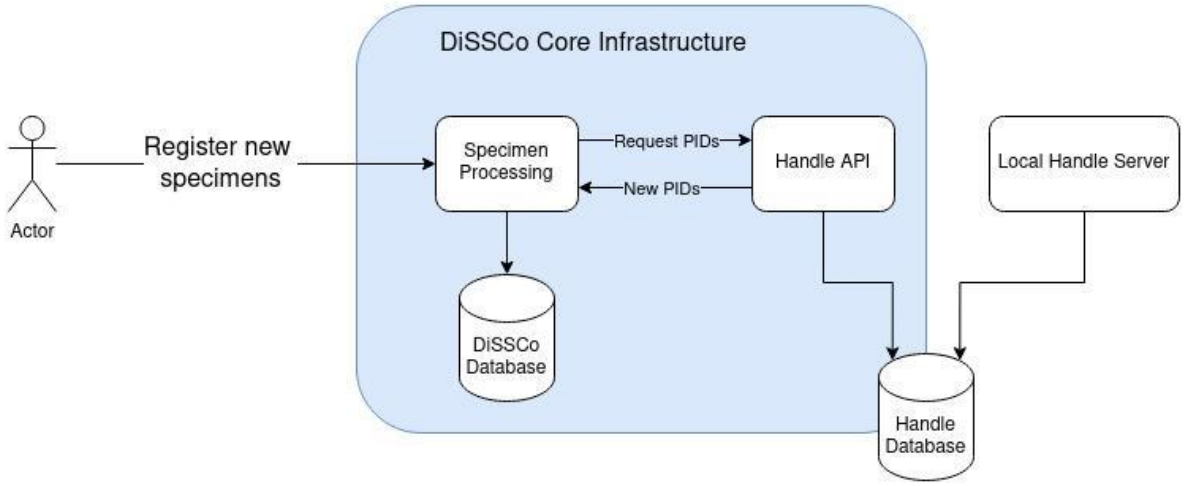

Figure 3. Situating the Handle Manager API within the specimen ingestion pipeline.

3.4.3. DOIs

The Handle System is a powerful tool to create globally resolvable identifiers, but Handles alone aren't persistent. As we've discussed, they are only records in a database. Digital Object Identifiers (DOIs) address this issue. DOIs are Handles with guaranteed persistence, which makes them a reliable tool for citation and provenance. A DOI may be recognized by its prefix beginning with "10.".

The DOI Foundation governs the DOI system, ensuring once a DOI is minted, it remains resolvable, regardless of whether or not the original institution exists or not. DiSSCo aims to use DOIs as identifiers for Digital Specimens and Media Objects to facilitate citation and persistent referencing of our resources. However, only trusted organisations, called Registration Agencies (RAs), can mint DOIs. These organisations form a community dedicated to persistent, reliable identifiers.

DataCite is one such organisation. It is a nonprofit organisation that provides open infrastructure and facilities registration of DOIs for research outputs and resources, and creates links between these. Because of their aligned mission statements, DiSSCo and DataCite are partnering to explore registration of FDO-compliant DOIs at scale for potentially billions of Digital Specimens. Over the course of this partnership, a mapping has been developed between the DiSSCo PID profiles and the DataCite metadata schema, so the FDO record may be exposed more broadly and comply with DataCite requirements.

For the BiCIKL project, 24 digital specimens were added manually to the DiSSCo Acceptance Environment[13] to allow these specimens to get a DOI and be cited with their Digital Specimen DOI in a publication made by Jeremy Miller. These represent spider specimens from Naturalis Biodiversity Center and the Manchester Museum (University of Manchester).

The DOIs are:

1. https://doi.org/10.3535/M42-Z4P-DRD

[13] https://sandbox.dissco.tech

2. https://doi.org/10.3535/1CE-SXA-2BC
3. https://doi.org/10.3535/SVV-BR5-KGE
4. https://doi.org/10.3535/3NW-1BX-8BK
5. https://doi.org/10.3535/7WH-VHP-M1K
6. https://doi.org/10.3535/B59-03B-FWV
7. https://doi.org/10.3535/C69-M7K-VWC
8. https://doi.org/10.3535/6H9-R1R-330
9. https://doi.org/10.3535/85R-G3E-4M0
10. https://doi.org/10.3535/FEE-JQY-GA4
11. https://doi.org/10.3535/0RA-FVV-2DL
12. https://doi.org/10.3535/Z2J-WMP-FDH
13. https://doi.org/10.3535/SGZ-EFZ-VRK
14. https://doi.org/10.3535/67X-9R9-YCM
15. https://doi.org/10.3535/MDR-6FG-49E
16. https://doi.org/10.3535/WL8-0R1-42B
17. https://doi.org/10.3535/5SG-PLB-MHT
18. https://doi.org/10.3535/SZV-FJV-MRM
19. https://doi.org/10.3535/5MR-J6N-26M
20. https://doi.org/10.3535/Q6C-91C-BS5
21. https://doi.org/10.3535/VYQ-YW1-AGE
22. https://doi.org/10.3535/G0G-G7D-N5J
23. https://doi.org/10.3535/PER-LNE-HEW
24. https://doi.org/10.3535/HS2-8W8-F23

3.5. Conclusion

The storage components described above together form the digital object repository (Figure 4). Each component is a vital part in this system. The data storage solutions each have a different task in which they excel. By combining the strengths of each component we can create a robust and efficient data architecture capable of handling a large volume of digital objects as well as their variable nature. Data changes should require minimal impact, providing the necessary flexibility. By using AWS managed solutions for our persistent data storage and therefore decoupling it from the compute layer, we ensure data safety and persistence.

Using different storage solutions does require additional attention to the application. It is important that the storage solutions are synchronised with each other. This is done by creating a transaction in the code. If any of the actions fails, we need to be able to rollback any successful actions. For example, if the storage in the relational database is successful, but the storage in the indexing solution is not, we need to roll back the relational database and PID database. Due to the microservice architecture, an error in one service may require a rollback in another, so these functions need to be created manually.

Overall, we can create a generic diagram where we show that for each digital object we require a digital object processor which orchestrates the storage transaction. This component forms a key piece in the DiSSCo architecture. After the data has been securely stored, we can have several components reading the data, for example the DiSSCo backend, which provides an API for retrieving data from DiSSCo.

ARPHA Preprints *Author-formatted document posted on 30/04/2025.* DOI: https://doi.org/10.3897/arphapreprints.e157339

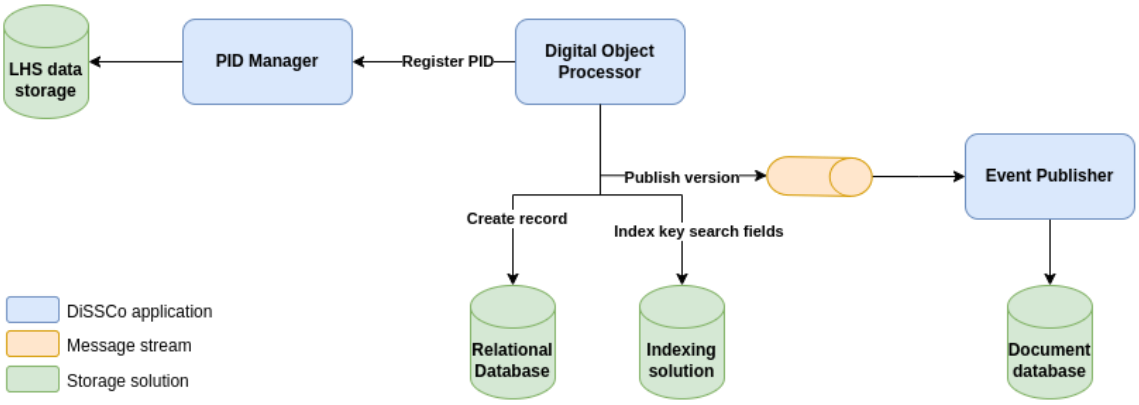

Figure 4. Storage solutions used within DiSSCo. For clarity, the translator and processing services have been merged into the “Digital Object Processor” block.

4. Repository Interfacing

This section gives an overview of different ways clients may interact with Digital Specimen data: through REST APIs, a web portal, or a DOIP server. Recall from the previous section that there are three storage solutions: A relational database that holds the latest version of a specimen, an Elasticsearch instance that facilitates rapid searching and aggregation, and a document store for provenance. These different solutions can be leveraged using different tools.

4.1. REST APIs

The DiSSCo infrastructure provides several REST APIs, which power the DiSSCover platform, or can be used to interface with the Digital Object repository directly. In this section, we discuss what is a REST API and the different applications these REST APIs can be used.

4.1.1. What is REST?

REST (Representational State Transfer) is an architectural style for designing networked applications. REST APIs are a way of enabling communication between different software systems over the Internet by using HTTP requests to perform actions on resources. REST APIs follow a client-server architecture (Figure 5), where the client (such as a web browser or a mobile app) sends requests to the server, and the server processes those requests and sends back appropriate responses.

Figure 5. REST API Client-Server Architecture.[14]

REST APIs use HTTP methods to indicate the type of operation being performed on a resource. The most commonly used methods are: GET (retrieving a resource), POST (creating a resource), PUT (update an existing resource), and DELETE (remove an existing resource). These HTTP methods provide a uniform interface for interacting with resources on the web, allowing clients and servers to understand each other's intentions without prior agreement, making it easier to build interoperable systems. Additionally, these methods are widely accepted, and this familiarity makes it easier for developers to work with REST APIs, regardless of the programming language or platform they are using.

DiSSCo REST API communicates with clients according to the JSON:API specification, providing a predictable response format for clients. The JSON:API specification is a standard for building APIs in JSON format. JSON:API builds upon the principles of REST, including the use of HTTP methods, while also providing additional conventions for structuring API responses. By following these conventions, the DiSSCo API is more consistent and easier to use for outside developers, enabling interoperability with external systems.

4.1.2. Core APIs

Two backend services enable interfacing with the Digital Object repository. For read operations, functionality is powered by the Core Backend service, which provides searching and retrieval operations. New specimens are created and updated in a separate pipeline: the translator and specimen processing services handle these write and update operations.

The Core Backend Service[15] [16] is useful for reading and searching for Digital Specimens. There are three storage solutions behind this API: a PostgreSQL database, an Elasticsearch[17] instance, and a MongoDB database, which are described in section 3.3 Storage Implementation. These endpoints are open, meaning any client can read or search for (non-sensitive) Digital Specimens.

Ingesting data is done through the Translator and Processing Services, which work together to ensure data quality, consistency, findability, and provenance. The Translator Service[18] is the first step in the ingestion process. It accepts specimen data in DarwinCore and ABCD formats and translates it into the OpenDS format used within DiSSCo. When data is properly formatted, the Translator sends the specimen data to the Processing Service via REST API. The Processing Service obtains a new Identifier for each specimen from the Handle API, and then registers

[14] Cloud icons created by kosonicon - Flaticon; Computer icons created by xnimrodx - Flaticon; Server icons created by Pixel Perfect - Flaticon; Api icons created by Tanah Basah - Flaticon

[15] GitHub: https://github.com/DiSSCo/dissco-core-backend

[16] The API is documented with the OpenAPI specification: https://sandbox.dissco.tech/api/swagger-ui/index.html#/

[17] https://www.elastic.co/

[18] GitHub: https://github.com/DiSSCo/dissco-core-translator/

ARPHA Preprints *Author-formatted document posted on 30/04/2025.* DOI: https://doi.org/10.3897/arphapreprints.e157339

specimens in Elasticsearch, MongoDB, and Postgres. Authenticated users may also create Digital Specimens directly through the Processing Service API.

4.2. GUI Interfaces

While the described services are useful for building larger infrastructures, they are less accessible to the general public. The DiSSCover Platform (Figure 6, Figure 7) and the DiSSCo Orchestration Service (Figure 8) are two public-facing interfaces to achieve the reading and writing operations described in the previous section.

The DiSSCover platform's main purpose is to search and explore Digital Specimen and related data stored in DiSSCo. It is powered by the Core Backend API. Users may filter on different fields, such as discipline or institution, or they may conduct a search using the search bar. These features utilise the Elasticsearch indexing.

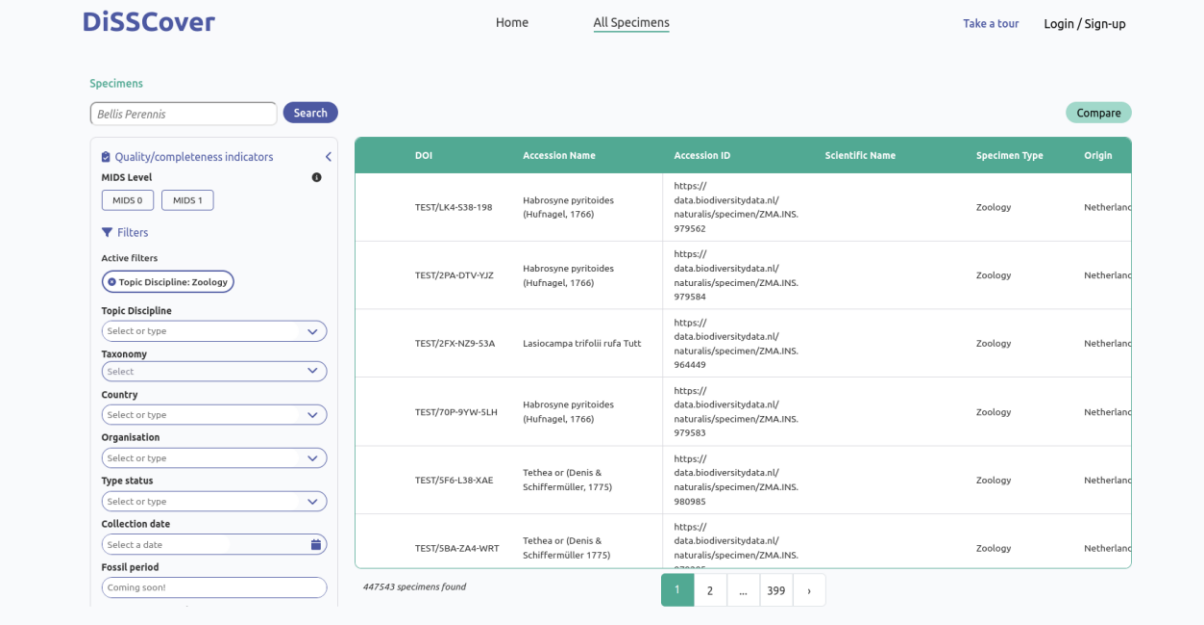

Figure 6. Main search page of DiSSCover platform: https://dev.dissco.tech

The latest version of each specimen is visible by default, but the user may choose to view older versions by selecting the "version" tab. This uses the document store to retrieve previous versions. Previous versions are thus always retained.

ARPHA Preprints *Author-formatted document posted on 30/04/2025.* DOI: https://doi.org/10.3897/arphapreprints.e157339

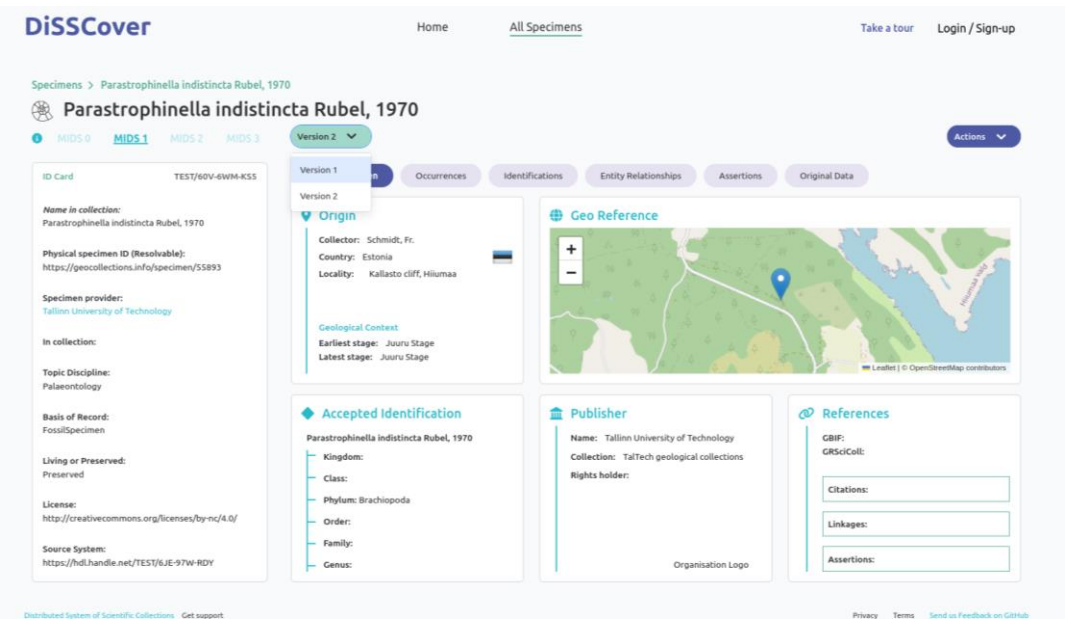

Figure 7. Specimen page on the DiSSCover Platform. The most recent version of data is obtained from the relational database. When users request earlier versions of the specimen, data are retrieved from the document store.

Ingestion of specimen data is aimed towards collection managers and other trusted users. To ingest specimens data into the database, authenticated users may register a “source system” within the DiSSCo Orchestration Service. The source system provides the Translation Service with an endpoint to pull data from, and begin the ingestion process described in the previous section. Once a source system is registered, ingestion is scheduled to occur automatically once a week, though users may also trigger an ingestion manually. Through the orchestration service, trusted community members are able to write to the repository and create Digital Specimens.

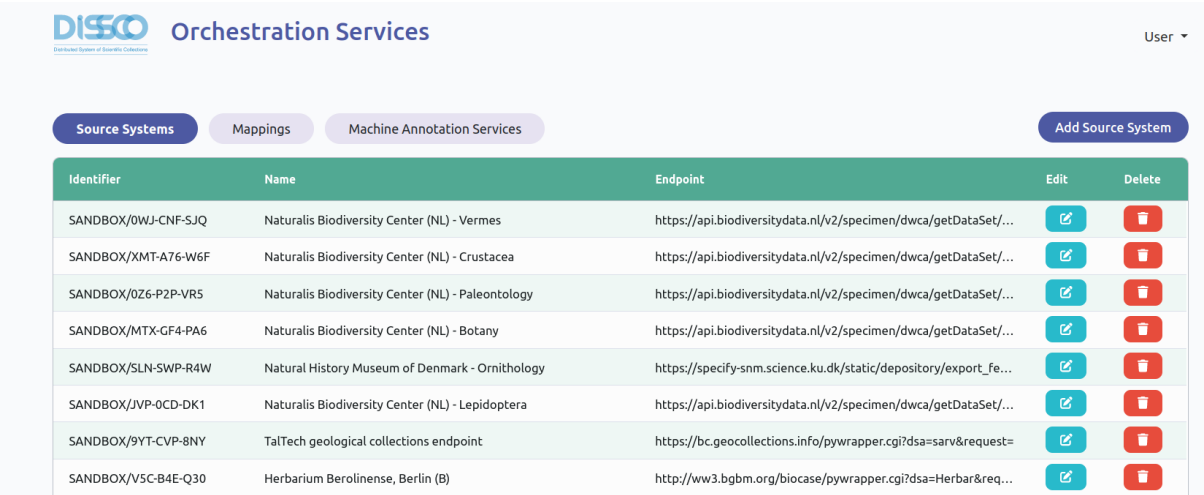

Figure 8. Source systems in the Orchestration Service. This service allows the Translator to pull information from a specified endpoint (e.g. a GBIF Integrated Publishing Toolkit node), triggering the ingestion process.

4.3. DOIP Server

In addition to the REST endpoints and front-end platform, a novel way to interface with Digital Specimen data was developed for this project: A Digital Object Interface Protocol (DOIP) server, connected to the DiSSCo Infrastructure.

Digital Object Architecture (DOA) provides a technology-agnostic approach to interacting with Digital Objects. Developed and maintained by the DONA Foundation, DOA relies on a globally unique and resolvable identifier, with an associated record of "state information", and Types, which inform machines what kind of information is associated with the object and what kind of operations may be performed. Digital Object Interface Protocol (DOIP)[19] is the protocol through which machines can use DOA. For this deliverable, a demonstrative DOIP server was deployed on AWS.

4.3.1. Serialisation

Any DO communicated through DOIP must be serialised according to the DOIP specification to ensure consistent serialisation and interaction. DOIP serialisation consists of multiple segments, which are structured units of information that collectively encode a digital object. Segments include JSON serialisation segments for metadata and attributes, as well as bytes segments for raw data.

Both "attributes" and "elements" are components that provide additional information about a digital object. However, they serve different purposes and are used in distinct ways within the protocol. Attributes provide metadata fields that offer contextual information about a digital object, while elements represent discrete units of content that can be included within the object. A serialisation for Digital Specimens was developed for MS33 *Documentation on how to interface with FAIR Digital Object via Digital Object Interface Protocol (DOIP*), wherein specimen-specific data were represented as attributes, and related information, such as annotations or media objects, were represented as elements. For Digital Media Objects, the media-related metadata stored in DiSSCo were stored in attributes, and the annotations were represented as elements.

4.3.2. Infrastructure and Deployment

A DOIP Server[20] was developed for demonstrative purposes using the DONA Foundation's libraries and deployed over Amazon Web Services. Clients may connect through OpenSSL and request specific operations be performed on Digital Specimens and Media Objects in the DiSSCo Acceptance Environment[21]. Requests and responses are formatted according to the DOIP specification. The server adheres to the DOIP specification of response codes, as illustrated in Table 1.

[19] Because of their similar names, it may be possible for the reader to equate "Digital Object Identifiers" (DOIs) with "Digital Object Interface Protocol." However, the two concepts are entirely distinct. DOIs do not refer to "Digital Objects" at all. The "Digital" in "Digital Object Identifier" characterises the identifier, not the object. It may be helpful to think of DOIs as "Digital Identifiers for Objects" instead. In fact, the term DOI predates the widespread concept of a "Digital Object".

[20] Source Code: https://github.com/DiSSCo/DOIP-Demo

[21] https://sandbox.dissco.tech/

Table 1: *DOIP Response Codes.*

Code	Message
0.DOIP/Status.001	The operation was successfully processed.
0.DOIP/Status.101	The request was invalid in some way.
0.DOIP/Status.102	The client did not successfully authenticate.
0.DOIP/Status.103	The client successfully authenticated but is unauthorised to invoke the operation.
0.DOIP/Status.104	The digital object is not known to the service to exist.
0.DOIP/Status.105	The client tried to create a new digital object with an identifier already in use by an existing digital object.
0.DOIP/Status.200	The service declines to execute the extended operation.
0.DOIP/Status.500	Error other than the ones stated above occurred

From the client's perspective, the DOIP server is an independent service; however, the basic DOIP operations have already been implemented by DiSSCo as a REST API, and the DOIP server takes advantage of this. When the server receives a request, the server formats the request and calls the most appropriate DiSSCo REST endpoint (described in the previous section); it then formats the response from the DiSSCo API into a DOIP-compliant response (Figure 9). Essentially, the server acts as a DOIP translator. This process is similar to the approach Cordra[22], a well-known adopter of DOIP, takes towards DOIP.

This process is complicated by a potential weakness in the DOIP Java SDK: it is not possible to specify the Type of the object being acted upon in the DOIP request. For instance, in a basic 0.DOIP/Op.Retrieve request, the client may only specify the identifier of the requested resource. In order to determine the Type of the object requested (and thus the appropriate DiSSCo endpoint), the DOIP server must first resolve the PID record of the object. Only once the Type of object is identified can the DiSSCo REST API be called and the object be retrieved.

[22] https://www.cordra.org/

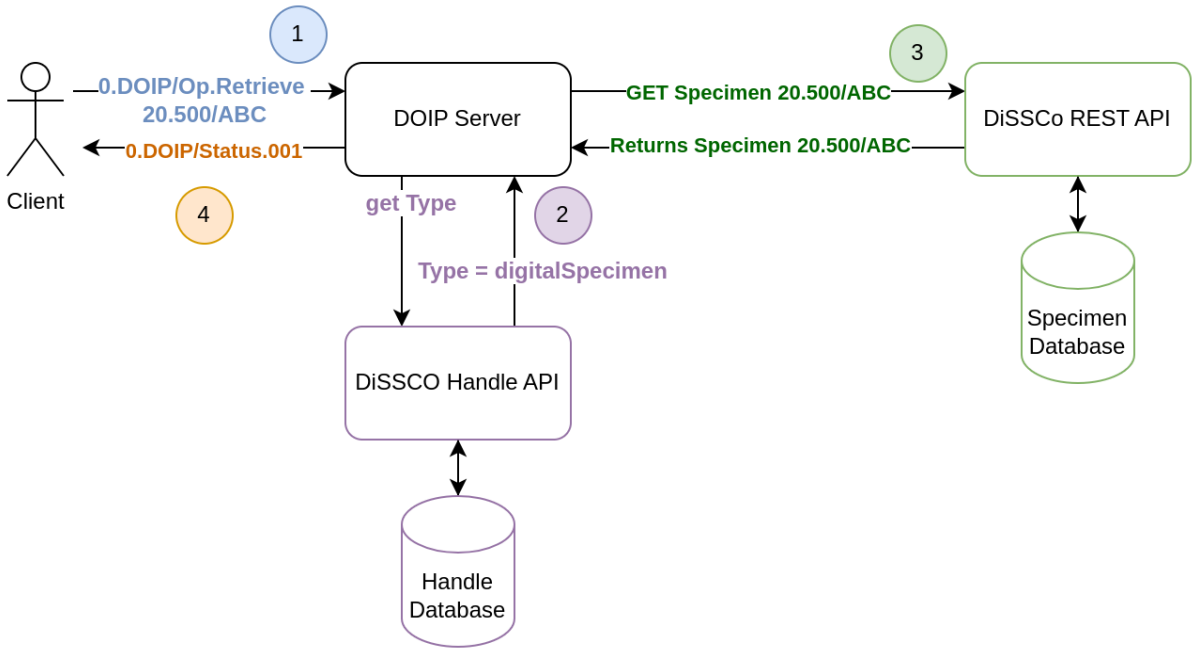

Figure 9. Demo DOIP Server Architecture. Upon receiving a Retrieve request (1), the DOIP server calls the DiSSCo Handle API to determine the Type of object requested (2). The DOIP server then uses the Type to determine which endpoint the main DiSSCo API to call (3). The DiSSCo REST API interfaces with the Digital Object repository and returns a JSON response, which the DOIP server translates to a DOIP-compliant response for the client (4).

4.3.3. Operations

Clients may connect to the DOIP server using an OpenSSL s_client[23]. Connection information can be found in the DOIP server information record: https://hdl.handle.net/20.5000.1025/SERVICE-INFO. The following section gives an overview of the available operations for the DOIP server and provides an example request.

0.DOIP/Op.Hello

The DOIP Hello operation is conducted on the Service Information itself. This operation takes two arguments: the operationId, and the identifier of the target, which in this context is the Handle of the Service Information Record. This operation can accept a Handle for any Service Information Record and return information about the server of interest.

Request:

```
{
 "operationId": "0.DOIP/Op.Hello",
 "targetId": "20.5000.1025/SERVICE-INFO"
}
```

0.DOIP/Op.ListOperations

This is an operation to list the operations that can be invoked by the DOIP server.

[23] https://www.openssl.org/docs/man1.0.2/man1/openssl-s_client.html

ARPHA Preprints Author-formatted document posted on 30/04/2025. DOI: https://doi.org/10.3897/arphapreprints.e157339

Request:

```
{
  "operationId":"0.DOIP/Op.ListOperations"
}
```

0.DOIP/Op.Retrieve

This operation allows the client to retrieve the DOIP serialisation of the latest version of a given Digital Specimen or Media Object (i.e. the digital entity record described in 3.2.6). Mandatory arguments for this operation are: the operationId, the targetId (which is the Handle of the DES). The client can request the full specimen (including annotations and media objects) by setting the “includeElementData” flag to true in the “attributes” field. The default behaviour is to not include the element data.

The latest version of the resource is always served, as the SDK does not support specifying versions.

Request:

```
{
 "operationId": "0.DOIP/Op.Retrieve",
 "targetId": "20.5000.1025/5CE-JN5-Z4H",
 "attributes": {
   "includeElementData": "true"
 }
}
```

0.DOIP/Op.Search

For searching, the query is constructed in the “query” attribute. The basic query (URL encoded) is submitted in the “q” field. Additional filters can be provided in this object as well. Searching is only supported for Digital Specimens.

Outside the “query” object, clients can specify pageNumber, pageSize, and response type (“type”). If “type” is set to “id”, only identifiers of the retrieved objects will be returned. Otherwise, the full serialised DO will be returned.

Request:

```
{
 "operationId": "0.DOIP/Op.Search",
 "attributes": {
   "query": {
      "q": "bombus",
      "midsLevel": 1
   },
   "pageSize": 2,
   "type": "id"
 }
}
```

4.4. REST vs DOIP

One notable advantage in REST lies in the familiarity and prevalence of REST services within the scientific community. As evidenced by the test server, existing REST services can effectively accommodate DOIP operations. In fact, all DOIP operations could be conducted by existing infrastructure, and the test DOIP server functioned primarily as a DOIP wrapper around REST services.

An integral component of the DO is its globally unique and resolvable identifier; each identifier is associated with a record containing relevant “state information” clients may access. At its core, a DO may be considered a sequence of bits with an identifier, with or without associated metadata. This approach is not unique to DOIP, and may be implemented using a REST approach.

Unfortunately, there is limited community uptake of the protocol. While DOIP is a generic way of interacting with DOs, as long as other infrastructures only support REST APIs, there is little benefit in this generic approach. The target audience of BiCIKL has a strong preference for REST APIs and is unfamiliar with DOIP. As long as communicating over HTTPS supports community needs, there is no strong benefit to adopting a new protocol.

By adopting DO-centric principles, REST services can be aligned with DOIP concepts to achieve similar outcomes. For instance, work is already being done by DiSSCo to assign globally resolvable identifiers to digital resources, a key concept emphasised by DOIP. Additionally, interoperability of REST services can be enhanced through the use of standardised serialisation, such as the json:api standard[24]. The adoption of established standards minimises ambiguity and promotes clearer communication across distributed networks.[25]

5. Supported Data Models

One of the goals of DiSSCo is to provide structured and harmonised data to our users. This means that we will have different data formats and serialisation coming into DiSSCo. Some of these could be part of TDWG standards, such as DwC or ABCD(EFG), while others will be in custom data models.

DiSSCo will harmonise the data from these different sources, formats, and serialisations into a single data specification which we call open Digital Specimen (openDS). OpenDS is being built on top of existing international standards such as DwC, MIDS, AC, LtC and others. DiSSCo leverages the work already done and confirms standards development under the Biodiversity Information Standards (TDWG) organisation.

[24] https://jsonapi.org/

[25] Wouter Addink, Niki Kyriakopoulou, Lyubinur Oebevm David Fichtmueller, Ben Norton, David Shorthouse. (2022). “Best practice manual for findability, re-use and accessibility of infrastructures” (BiCIKL Deliverable D1.3)

OpenDS has been developed after a series of tests with other simpler models. These tests indicated that DiSSCo needs to accommodate a wide range of information. We require complex structures if we want to capture information about multiple (competing) taxonomic identifications, events, agents, and relationships to other data sources. Being able to handle these complex objects is essential if we want to meet our goal of harmonising all specimen information and providing all relevant links between the specimen and external data sources.

5.1.1. GBIF Unified Model

DiSSCo was not the only infrastructure searching for a broad and extensive data model. The Global Biodiversity Information Facility (GBIF) was researching a new model as well. Under the guidance of John Wieczorek and Tim Robertson, they developed a new Unified model (UM).[26] This model is based on a wide range of use cases covering a broad range of biodiversity data. During the DiSSCo Prepare Project (DPP), members of the DiSSCo openDS team have had contact with Tim Robertson (Head of Informatics at GBIF) to talk about aligning the efforts.

5.1.2. DiSSCo adaption of the GBIF UM

DiSSCo has followed the work of John and Tim with great interest. They have proven that their UM fits the bill and might become the next standard in biodiversity data. After careful consideration, DiSSCo has decided to create an adaption of the GBIF UM which focusses on specimen data. We call it an adaptation, as we will follow the structure of the GBIF UM but will focus purely on specimen data. We also use a different serialisation and will have slight differences to cover for DiSSCo-specific information. Additionally, with DiSSCo, we want to enforce stricter control over the used values and provide additional data related to specimens, such as access information for loans and visits and a richer provenance model

The advantages of adapting the GBIF UM to DiSSCo are plentiful. DiSSCo will not have to invest time and expertise into developing its own, potentially competing data standard. As a Biodiversity community, we would like to move towards a single interoperable data standard. DiSSCo believes that due to its broadness and flexibility the GBIF UM can make an important contribution to uniting the different data models currently used in the Biodiversity community. By using and promoting this standard DiSSCo can help to establish the GBIF UM as the main form of data modelling.

Basing the openDS data specification on the GBIF UM will ensure DiSSCo's interoperability with GBIF. One of DiSSCo's goals is to provide complete, high quality data to GBIF and other data aggregators. By aligning our data structures, we ensure there won't be any issues nor will information be lost in a translation between the systems.

5.1.3. Serialisation

Within DiSSCo, most data transfer and data storage is in JavaScript Object Notation (JSON). The GBIF UM, however, is a strongly relational model. We therefore created a JSON-Schema

[26] https://www.gbif.org/composition/HjlTr705BctcnaZkcjRJq/gbif-new-data-model

based on the GBIF UM.[27] The main difference between the relational model and the JSON-Schema is that we de-normalise several tables. This will mean that information connected to the specimen will be duplicated within the specimen.

For example, suppose multiple specimens were collected by the same agent on a single occurrence. Within the GBIF UM, this will ideally be a single record, connected to a single occurrence, connected to a single collector. Within DiSSCo, we will get a record per collected specimen, which includes in the object the occurrence and the collector. This means we will duplicate the occurrence and the collector information.

The main mean reason behind this is that we see the specimen information as leading. A change in the occurrence will trigger a new version of the specimen. So if the information of the agent changes, this will force all connected specimens in the DiSSCo infrastructure to generate a change event and increment the version of the specimen. This will help us to provide full provenance for the specimen.

This also simplifies part of the model as we reduce the amount of identifiers needed to indicate relationships between the objects. Several of these relationships become implicit relationships. The challenge of creating the identifiers has also been identified by GBIF as one of the Common Challenges.[28]

5.1.4. Changes to the Unified Model

DiSSCo has made several small changes to the GBIF UM. These changes are connected with two main differences. The first difference is DiSSCo's sole focus on the specimen information. This means that we see the specimen as an entrypoint into the model and built around this. The second difference is the scope. Where GBIF only focuses on the biological information, DiSSCo will cover geological specimens as well. We believe the data model will fit for most of the information, however several classes or fields need to be added to cater this community.

We would like to change the name Organism to Digital Specimen. Organism is the main class that GBIF UM uses for a specimen. For GBIF, the name organism works as it is solely geared towards organic material. However, DiSSCo would also like to include geological specimens. This means the name would be an ill fit and might create confusion. Within DiSSCo we use the term "Digital Specimen" to indicate the digital information of a physical specimen. Therefore, the name Digital Specimen would be a better fit.

Additionally, we would like to include specimen parts. This means that one Digital Specimen could have several Material Entities. The information in the Digital Specimen will be about the complete specimen, but individual parts will be captured as a material entity. This is a specific requirement for specimen parts which didn't receive a separate registration number. Otherwise, both would be Digital Specimens with an entity relationship between them.

As DiSSCo includes earth science (e.g. geological) specimens, we would need to adopt the GBIF UM to be able to incorporate them. We believe most information about geological specimens would already fit into the model, as it is generic enough to cater for them. However,

[27] https://schemas.dissco.tech
[28] https://github.com/gbif/model-material/blob/master/lessons_learned.md

we would probably need to add additional identification classes. This means that next to the TaxonIdentification class we would add another class, such as MineralIdentification. In this way we can extend the model to include other types of identifications.
DiSSCo will have close contact with the TDWG tasks groups “Extension for Geosciences (EFG)” and Mineralogy Extension.

DiSSCo wants to add several fields specific for the DiSSCo infrastructure. Examples of these terms are “version”, “created” and “mids_level”. These fields can be separated into two different groups. One is metadata about the Digital Object. These fields provide information about the digital object such as the digital object identifier, the version, the creation date of the version, the main physical identifier and the main physical identifier type.
The second group of terms provides additional information, which helps in the digitisation effort of the collections. These terms include the MIDS level, the topic origin, the topic domain, topic discipline and several other fields. This information might be interesting for data aggregators. Discussion is needed to see how DiSSCo can provide this information.
With the openDS specification, we want to provide additional restraints. A uniform way in the interpretation of the data model will improve the interoperability and reusability of the data. This means that for some fields, we will enforce a controlled vocabulary. For others, we will remove ambiguity by choosing a specific implementation. This means that we will remove some flexibility in the GBIF UM and enforce a strict interpretation helped with controlled vocabularies and pattern matching.

5.2. Digital Specimen

In this section, we will give a short summarising overview of the open Digital Specimen data specification. We set out the main lines and point to the relevant resources for further information. OpenDS is still in active development and the model might change. In this document, we will point to our first version of the model, which hasn’t been finalized yet. New iterations of the model will be published at https://schemas.dissco.tech/schemas/ which is based on our GitHub repository which can be found at: https://github.com/DiSSCo/openDS.
The Digital Specimen is the top level object within openDS. A digital specimen is any object in a natural history collection that has a physical identifier (catalogue number, accession number, occurrence identifier, etc...). This object contains all information about the digital specimen. It consists of a series of attributes containing generic information about the specimen and a number of nested lists of objects providing specific information. For example, top level attributes are “ods:id” which contains the DOI of the specimen but also things as the version (“ods:version”), FDO type (“ods:type) or the topicDiscipline are attributes on this level.
The schema for the digital specimen can be found here:
https://schemas.dissco.tech/schemas/digitalobjects/0.1.0/digital-specimens/

ARPHA Preprints *Author-formatted document posted on 30/04/2025.* DOI: https://doi.org/10.3897/arphapreprints.e157339

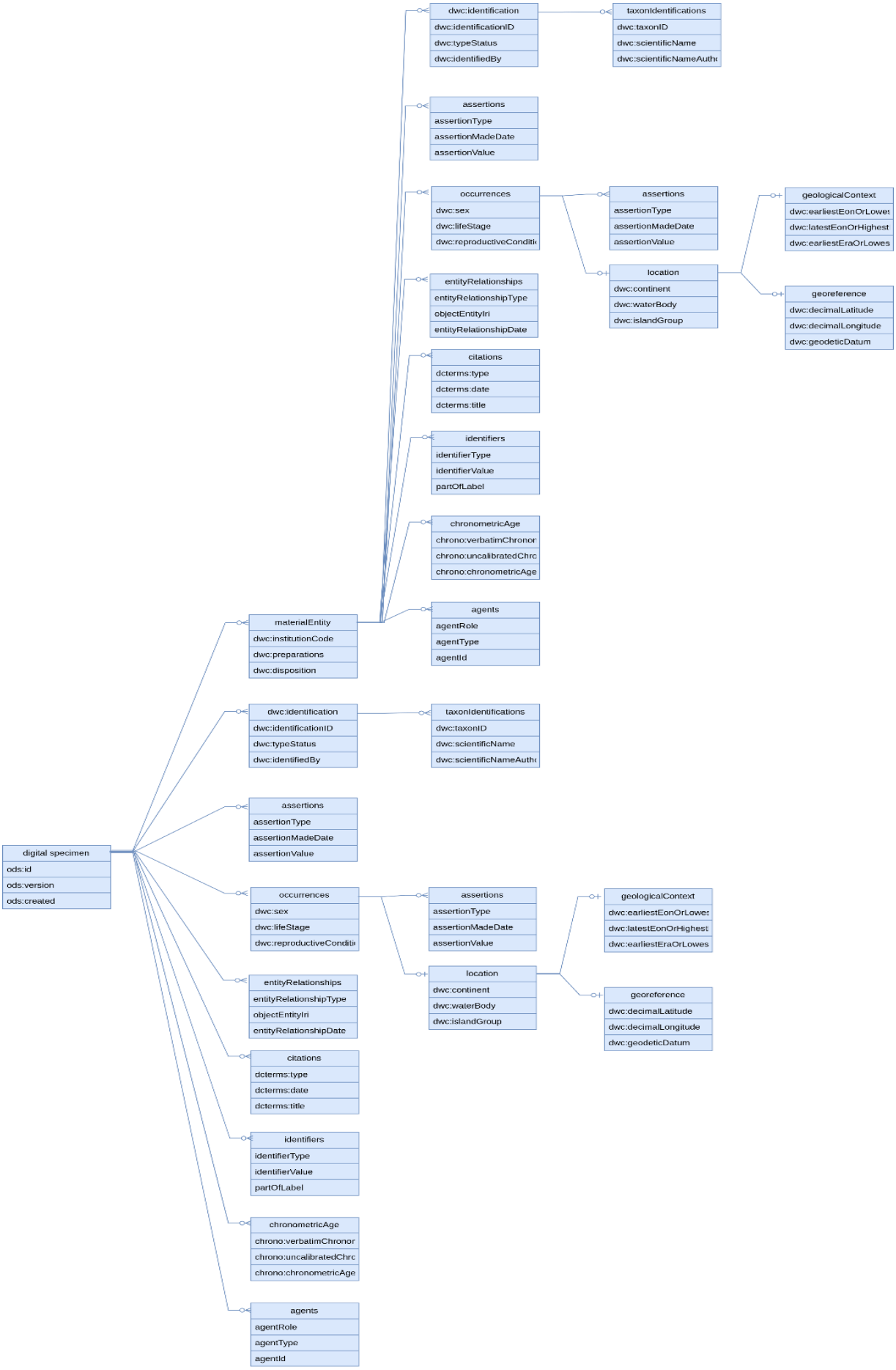

Figure 10. Simplified version of the openDS data model. For each object three attributes have been included to give an indication of the object. For all attributes see the relevant JSON Schema.

5.2.1. Material Entity

A Digital Specimen can have several Material Entities. These Material Entities are very similar to a digital specimen, with one major difference. A Material Entity does not have its own physical identifier, but is part of the digital specimen. An example is a geological specimen that contains multiple fossil specimens. When only the geological specimen has a physical identifier, the individual fossil specimen can each be seen as a material entity. This means they can have their own measurements, identification, relationships, etc... This will also provide the needed flexibility when we describe containers containing multiple specimens which need to be individually described.
The schema for the material entity can be found here:
https://schemas.dissco.tech/schemas/digitalobjects/0.1.0/digital-specimens/material-entity.json

5.2.2. Identification

Each digital specimen or material entity can have one or more identifications. These identifications provide information about who identified the specimen, whether the specimen is a type specimen (and what kind of type for which name), and what the identification status is. In general, each specimen should only have one accepted identification (the one dated most recent) but can have more supporting identifications. These supporting identifications can be alternative identifications, they can also be historic identifications.
Nested in the identification object is an array of a specimen type specific identification, such as a taxonomic identification. This taxonomic identification contains information about the taxonomy of the object and contains the full scientific name. Ideally, this taxonomic information should have been resolved against a taxonomic list such as Catalogue of Life. For non-biological specimens, we will need other domain specific sub-identification objects. These have not been included in this version. More work with the specific communities is needed to develop this model.
The schema for identification object, including any nested objects, can be found here:
https://schemas.dissco.tech/schemas/digitalobjects/0.1.0/digital-specimens/identifications.json

5.2.3. Occurrence

The occurrence object contains information about the events surrounding the specimen, such as the collecting event. It is an array, as one specimen can have multiple events. It contains information such as the event date, the habitat, field numbers and field notes and the quantity of the specimen.
Within the occurrence object is nested the location object. The location object contains location about the country, city, depth, or elevation of the object. It also contains two further nested objects, the georeference and the geological context. The georeference object contains all

information about the location of a specimen to pinpoint the location on a map. The geological context contains information about the stratigraphy of the specimen.
The schema for occurrence can be found here: https://schemas.dissco.tech/schemas/digitalobjects/0.1.0/digital-specimens/occurrences.json
The schema for locations, including georeference and geological context, can be found here: https://schemas.dissco.tech/schemas/digitalobjects/0.1.0/digital-specimens/location.json

5.2.4. Entity Relationship

Each digital specimen can have multiple entity relationships to other objects. These other objects could be other digital entities. For example, when different pieces of a single organism have been registered as different digital specimens, they can be connected through an entity relationship. The entity relationship can also be used to connect the digital specimen to external resources. For example, when the digital specimen has a taxonomic identification based on Catalogue of Life, there should be an entity relationship connecting the digital specimen to the taxonomic record. When the Wikidata identifier of the collector is added to the specimen, an entity relationship to Wikidata should be created.
This helps us to build a web of relationships in which the digital specimen is the centre and connects to a multitude of external resources. From these external resources, we will only take over the crucial metadata but will otherwise refer (link) to them as the main source of information. This behaviour as "bag of links" limits unnecessary data duplication and acknowledges the external data sources in their authority for the specimen related data these serve, while supporting FAIR by including descriptive metadata about these references.
The schema for entity relationships can be found here: https://schemas.dissco.tech/schemas/digitalobjects/0.1.0/shared-models/entity-relationships.js

5.2.5. Identifier

Each specimen can have one or more identifiers. To capture the individual identifiers and the metadata about them, we use the identifier object. All specimen identifiers (locally or globally unique, current and historical) can be added and it can be indicated if they are part of the label or barcode.
The schema for identifier can be found here: https://schemas.dissco.tech/schemas/digitalobjects/0.1.0/shared-models/identifiers.json

5.2.6. Assertion

Assertions provide additional information about the specimen. A single digital specimen can have multiple assertions. This could for example be information about the mass of a specimen or specific measurements. An assertion is a generic element consisting of a key value pair, with some additional metadata. This makes it a very flexible object, able to contain any kind of additional information about the specimen. However, when there is a specific attribute for a piece of information, we prefer to use the attribute. For example, the quantity of the specimen could be captured in an assertion, but we also have an attribute in the occurrence object called “dwc:organismQuantity”. We would prefer to use the Darwin Core attribute over a generic assertion.

The schema for assertion can be found here:
https://schemas.dissco.tech/schemas/digitalobjects/0.1.0/shared-models/assertions.json

5.2.7. Citation

The Digital Specimen could be cited in a publication. These publications can be added to the specimen through the citation object. One specimen can have multiple citations. The citation object contains information about the publication. In addition to the citation object, we would also like to capture the link to the publication through an entity relationship.
The schema for citation can be found here:
https://schemas.dissco.tech/schemas/digitalobjects/0.1.0/shared-models/citations.json

5.2.8. Agent

The Agent object contains information about the agent. This object is still in development, and we are in close contact with the attribution interest group within TDWG.[29] Each specimen will have multiple agents who performant an action on the specimen. These agents can have different roles, such as collector, recorder, or preserver. The agent object captures information about the agent's role and in what period he was active. We would also like to capture a persistent identifier for the agent, such as a Wikidata identifier or an Orcid identifier. These relationships to wikidata or ORCID iD should also be captured in an entity relationship.
The schema for agent can be found here:
https://schemas.dissco.tech/schemas/digitalobjects/0.1.0/shared-models/agent.json

5.2.9. Chronometric age

The chronometric age object is specific to specimen where information about the age is important, such as anthropological or archaeological specimen. It contains information about the age of the object and how that age was determined. A specimen can have multiple chronometric age objects.
The schema for chronometric age can be found here:
https://schemas.dissco.tech/schemas/digitalobjects/0.1.0/digital-specimens/chronometric-age.json

5.3. Digital Media

Besides Digital Specimen, DiSSCo will also create digital objects for digital media. Digital media is a broad category and can contain 2D images, video, 3D images, sound recordings and more. The media object should be directly connected to the specimen[30]. Within DiSSCo we will use a one-to-many relationship for digital media, one specimen can contain more than one media object. One media object can only be connected to one specimen. This might create some

[29] https://www.tdwg.org/community/attribution/

[30] Specimens may also have physical media and other supporting materials associated with them, support for linking such materials is foreseen for the future and already taken into account in the FDO record metadata.

duplication of information when one media object holds multiple specimens. However, it will make things easier when changes are made to the media object's metadata.
The data model for the digital media object is also an adaptation of the GBIF Unified Model. The digital entity is the entry point for the digital media objects. The digital entity holds all the metadata about the media object such as the DOI, the version, the licence but also the access URI.

Within the Digital Entity there is room for nested objects. Many of the nested objects used for the digital specimen are also available for the digital entity. The digital entity can have one or more citations, identifiers, assertions, entity relationships or agents. This provides us with the flexibility to add additional metadata on the level of the digital entity, like TDWG Audubon Core metadata.

The schema for the digital entity can be found here:
https://schemas.dissco.tech/schemas/digitalobjects/0.1.0/digital-media-objects/digital-entity.json

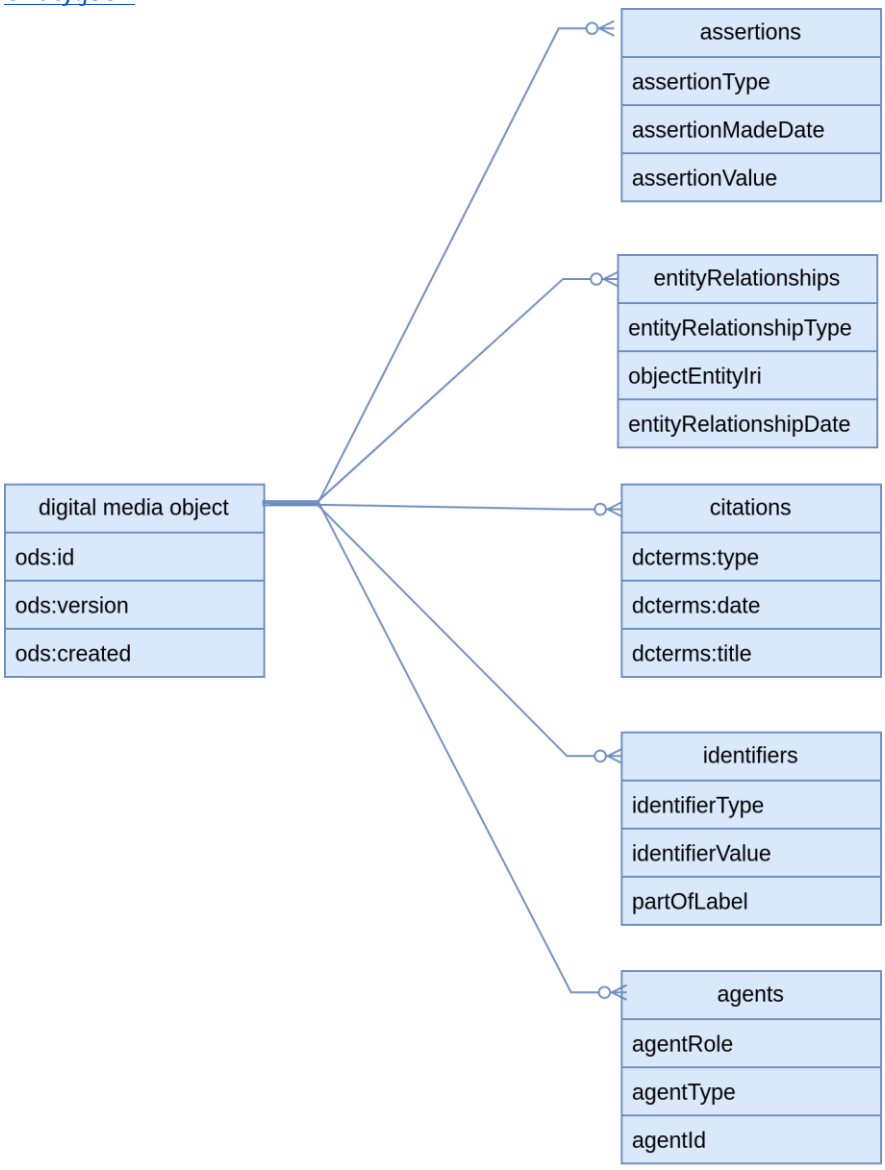

Figure 11. Simplified version of the Digital Media Object data model. For each object three attributes have been included to give an indication of the object. For all attributes see the relevant JSON Schema.

6. Conclusion

In this document, we described comprehensively the design choices and technical implementation of the Digital Object repository to store and provide digital specimen information. The repository is available as a fully functional installation in a sandbox environment: https://sandbox.dissco.tech (TRL level 6), which includes real data for testing and can be interacted with through GUIs (DiSSCover platform, Orchestration service) and APIs. Documentation for the APIs and data models is also available. Code for the repository is available as open source in GitHub under Apache 2.0 licence. Minting of Digital Specimen DOIs has been demonstrated in a TDWG Biodiversity Standards conference (fall 2023), and the first Digital Specimen DOIs have been created to be used in a scientific publication. These include the FDO record with machine-actionable metadata (see for example: https://doi.org/10.3535/1CE-SXA-2BC?noredirect) and redirect to both a HTML representation for humans and a JSON representation for machines. Further development and end-user testing is needed to deploy the repository in a production environment (TRL7).

7. Acknowledgements

The authors would like to thank the DiSSCo Technical Advisory Board, DiSSCo Technical Team, contributors to openDS, BiCIKL partners and everybody else who contributed to the work resulting in this deliverable for their valuable contributions.

8. References

Fitzgerald, H., Juslén, A., von Bonsdorff-Salminen, T., von Mering, S., Petersen, M., Raes, N., Islam, S. and Figueira, R. (2021). D1. 1 Report on life sciences use cases and user stories. https://doi.org/10.34960/xhxw-cb79

Glöckler F, Reis J.P, von Mering S, Petersen M, Weiland C, Dillen M, Leeflang S, Haston E, Addink W, Fichtmüller D (2022) 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

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

Hardisty AR, Ellwood ER, Nelson G, Zimkus B, Buschbom J, Addink W, Rabeler RK, Bates J, Bentley A, Fortes JAB, Hansen S, Macklin JA, Mast AR, Miller JT, Monfils AK, Paul DL, Wallis E, Webster M, et al. (2022). Digital Extended Specimens: Enabling an Extensible Network of Biodiversity Data Records as Integrated Digital Objects on the Internet. Bioscience 72 (10): 978-987. https://doi.org/10.1093/biosci/biac060

Hedrick BP, Heberling JM, Meineke EK, Turner KG, Grassa CJ, Park DS, Kennedy J, Clarke JA, Cook JA, Blackburn DC, Edwards SV (2020). Digitization and the future of natural history collections. BioScience 70 (3): 243-251. https://doi.org/10.1093/biosci/biz163

Islam S, Beach J, Ellwood ER, Fortes J, Lannom L, Nelson G, Plale B (2023). Assessing the FAIR Digital Object Framework for Global Biodiversity Research. Research Ideas and Outcomes 9: e108808. https://doi.org/10.3897/rio.9.e108808

Leeflang, S., Weiland, C., Grieb, J., Dillen, M., Islam, S., Fichtmueller, D., Addink, W., & Haston, E. (2022). DiSSCo Prepare D6.2 Implementation and construction plan of the DiSSCo core architecture. https://doi.org/10.5281/zenodo.6832200

Tilley LJ, Woodburn M, Vincent S, Casino A, Addink W, Berger F, Bogaerts A, De Smedt S, French L, Islam S, Mergen P, Nivart A, Papp B, Petersen M, Santos C, Schiller EK, Semal P, Smith VS, Wiltschke K (2024) Systematic Design of a Natural Sciences Collections Digitisation Dashboard. Research Ideas and Outcomes 10: e118244. https://doi.org/10.3897/rio.10.e118244