

 UNIVERSITÉ DE GENÈVE DIVISION DU SYSTÈME ET DES TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION				 Co-funded by the European Union	
Organisation:		Université de Genève (UNIGE)			
Project:		DNA MICROFACTORY FOR AUTONOMOUS ARCHIVING			
Grants:		EIC Pathfinder Challenges 2022 Project No.: 101115389			
Title:		2.1 Framework architecture specifications			
				<i>Rev: 1</i>	<i>Page: Page 1 of 18</i>

Deliverable Number 2.1

Framework architecture specifications Related WPs: D2.4 + CODEC (ICL) Related tasks: 2.1 OAIS framework
--

	Affiliation / Position	Name	
Author	UNIGE / Deputy CIO	Pierre-Yves Burgi	
Verifier	UNIGE / CTO	Hugues Cazeaux	
<i>Issued:</i>	2024.02.02	Update:2024.27.03 Version 1.0	P.-Y. Burgi

1. MANAGEMENT

Description:	Definition of the OAIS framework, which provides the archival dimension of the overall DNA architecture.
Products:	An OAIS-compliant framework with all the associated archiving processes described, taking into account the specificities of DNA data storage and the various interactions required to work with the micro-lean laboratory.
Reporting:	Monthly report (https://forms.office.com/e/vK6Wfd99yV) Gantt chart (Mindomo Cloud)
Accountable (name/e-mail)	Pierre-yves.burgi@unige.ch
Responsible (name/e-mail)	Hugues.cazeaux@ unige.ch

Accountable	<i>Person approving and validating actions. They are accountable for the scope entrusted to them. They are the real managers. They have authority over the Responsible.</i>
Responsible	<i>Person in charge of carrying out the task for which they are appointed. They report to an A. Usually more than one name</i>

KEYWORDS:

OAIS, DLCM,, OLOS, archiving process, access process, disaster recovery, disposal process

TABLE OF CONTENTS

1. MANAGEMENT	2
2. INTRODUCTION	6
3. THE OPEN ARCHIVAL INFORMATION SYSTEM (OAIS) REFERENCE MODEL	6
3.1 General concepts	6
3.2 OAIS information definitions	8
3.3 Archival Storage Functional Entity specifications.....	9
4. THE DLCM FRAMEWORK.....	10
5. SPECIFICATIONS OF THE ARCHIVAL STORAGE ENTITY APPLIED TO DNA.....	11
5.1 Structure of the AIP.....	11
5.2 Archiving Process	12
5.3 Accessing Process.....	12
5.4 Fixity Checking Process.....	13
5.5 Disaster Recovery.....	13
5.6 Replace process	14
5.7 Disposal process.....	14
6. IMPLEMENTATION OF A SIMPLIFIED SCENARIO	14
6.1 Writing an archive	15
6.2 Reading an archive	16
6.3 Disaster recovery	17
7. CONCLUSIONS.....	18
8. REFERENCES	18

TABLE OF FIGURES

FIGURE 1:	THE DNAMIC ARCHITECTURE
FIGURE 2:	OAIS FUNCTIONAL ENTITIES
FIGURE 3:	ENVIRONMENT MODEL OF AN OAIS
FIGURE 4:	FROM DATA OBJECT TO INFORMATION OBJECT
FIGURE 5:	FUNCTIONS OF THE ARCHIVAL STORAGE FUNCTIONAL ENTITY
FIGURE 6:	THE DLCM OAIS MODULAR ARCHITECTURE
FIGURE 7:	ARCHIVING PROCESS
FIGURE 8:	ACCESSING PROCESS
FIGURE 9:	RELOADING PROCESS (DISASTER RECOVERY)
FIGURE 10:	REPLACE PROCESS
FIGURE 11:	DISPOSAL PROCESS
FIGURE 12:	(DETAILED) WRITING PROCESS
FIGURE 13:	(DETAILED) READING PROCESS
FIGURE 14:	(DETAILED) DISASTER RECOVERY PROCESS

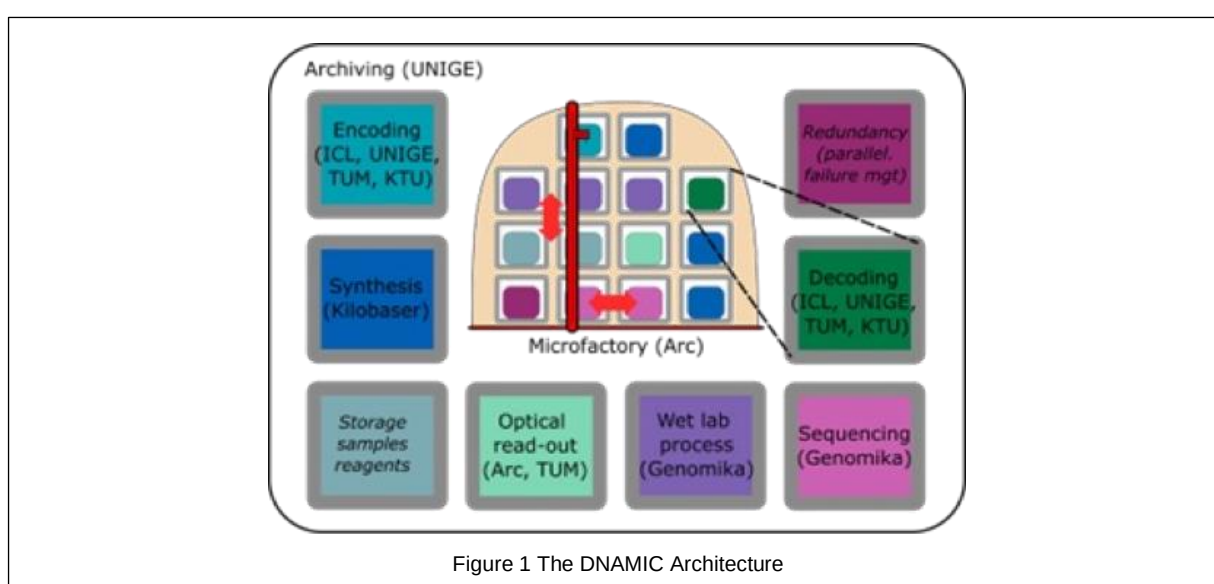
ABBREVIATIONS

AIP	ARCHIVAL INFORMATION PACKAGE
API	APPLICATION PROGRAMMING INTERFACE
CCSDS	CONSULTATIVE COMMITTEE FOR SPACE DATA SYSTEMS
CODEC	ENCODER AND DECODER
DIP	DISSEMINATION INFORMATION PACKAGE
DLCM	DATA LIFE-CYCLE MANAGEMENT
DNA	DEOXYRIBONUCLEIC ACID
DOI	DIGITAL OBJECT IDENTIFIER
MILL	MICROLEAN LAB
NFS	NETWORK FILE SYSTEM
OAIS	OPEN ARCHIVAL INFORMATION SYSTEM
OAI-PMH	OPEN ARCHIVES INITIATIVE PROTOCOL FOR METADATA HARVESTING
PDI	PRESERVATION DESCRIPTION INFORMATION
PUID	PRONOM PERSISTENT UNIQUE IDENTIFIER
REST	REPRESENTATIONAL STATE TRANSFER
SIP	SUBMISSION INFORMATION PACKAGE

2. INTRODUCTION

The general architecture of the DNAMIC project is represented in Figure 1. The main idea is to have a modular approach where each entity is autonomous and is part of the micro-factory that orchestrates the whole. On top of all this is an archival framework defined in accordance with ISO 14721, also known as the Open Archival Information System (OAIS) reference model (CCSDS, 2019).

The aim of this document (deliverable 2.1) is to clarify the role of the OAIS and its implementation, OLOS, within the overall framework of the end-to-end DNA archiving solution. Therefore, in order to set the scene for a more in-depth examination of the development of a new DNA connector for document archiving via the OLOS user interface, we first present the OAIS concepts in Section 3, followed by the DLCM (OLOS) framework in Section 4. Section 5 then presents a macro-analysis of the processes required to implement the OAIS standard. Section 6 describes in greater detail the interactions between DLCM and the microfactory.



3. THE OPEN ARCHIVAL INFORMATION SYSTEM (OAIS) REFERENCE MODEL

3.1 General concepts

The OAIS model is a conceptual framework that provides a common understanding of the processes and functions involved in the long-term preservation of digital information. It was developed by the Consultative Committee for Space Data Systems (CCSDS) and is designed to address the challenges associated with the preservation of digital information over the long term, ensuring its accessibility, usability, and authenticity. It is intended to be technology-independent, allowing organizations to implement it using different technological solutions (and thus even DNA !). It provides a framework for understanding the roles and interactions of different stakeholders involved in the preservation process and helps establish best practices for long-term digital information preservation.

The OAIS is organized into six functional entities and related interfaces, as shown in Figure 2 where only the main information flows are represented. The key components include (CCSDS, 2019):

1. **Ingest:** This phase involves the intake of digital information into the archival system. This includes tasks such as receiving, verifying, and preparing the information for preservation, based on the Submission Information Package (SIP).

2. **Archival Storage:** This phase involves the storage and management of the digital information in a way that ensures long-term preservation. It includes tasks such as data management, migration, and replication. It involves the Archival Information Package (AIP).
3. **Data Management:** This component is responsible for the organization and maintenance of the digital information within the archive. It includes activities such as data formatting, metadata creation, and access control.
4. **Administration:** This phase involves the overall management and coordination of the archival system. It includes tasks related to system monitoring, auditing, and policy implementation.
5. **Access:** This component is responsible for providing access to the preserved digital information. It includes tasks such as retrieval, dissemination, and user support. Access to information is enabled through the Dissemination Information Package (DIP).
6. **Preservation Planning:** This aspect involves the development and implementation of strategies to ensure the long-term preservation of digital information. It includes activities such as risk assessment, technology monitoring, and migration planning.

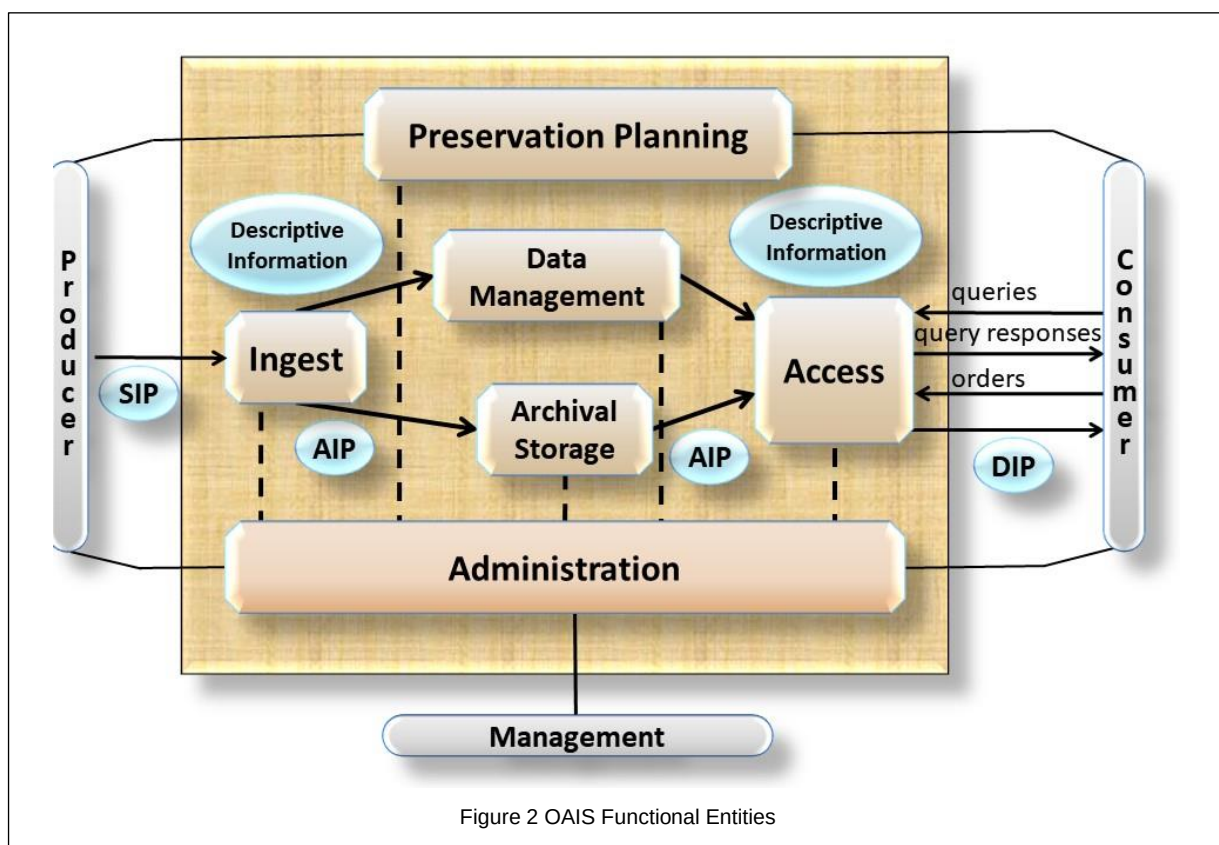
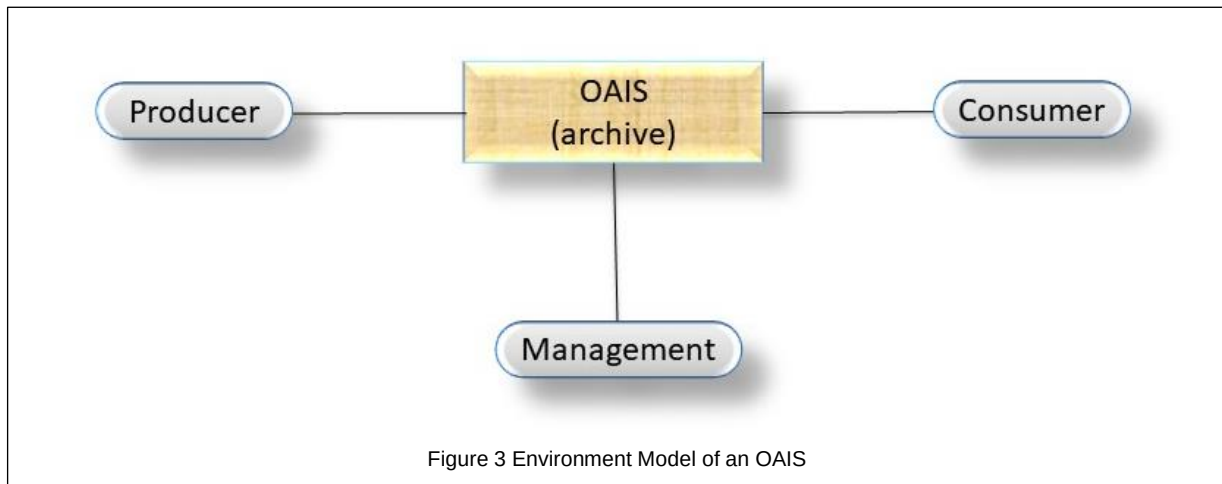


Figure 2 OAIS Functional Entities

Outside the OAIS are Producers, Consumers, and Management (Figure 3) :

- The producer is the role played by the people or client systems that supply the information to be preserved;
- Management is the role played by those who set overall OAIS policy. Management is not involved in day-to-day archive operations. The responsibility of managing the OAIS on a day-to-day basis is included within the OAIS in an Administrative Functional Entity (see text above and Figure 5);

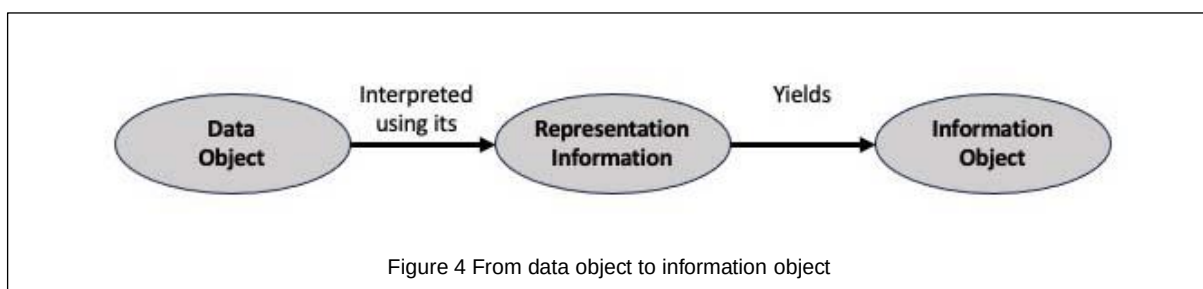
- The consumer is the role played by those persons, or client systems, that interact with OAIS services to find and acquire preserved information of interest. A special class of consumers



is the “Designated Community”. The Designated Community is the set of consumers who should be able to understand the preserved information. A given individual or system may act in the role of both a consumer and a producer.

3.2 OAIS information definitions

To qualify as an OAIS Information Object, the data items (Data Object) intended to be maintained must be associated with a "Representation Information", a piece of information documenting this data (how it was produced, what it is made of, what media and formats were used to contain it, etc.) so that it can be understood at any time (Figure 4). For this Information Object to be successfully preserved, it is essential that an OAIS clearly identifies and understands the data object and its associated representation information. For DNA information, this means that the OAIS must clearly identify the nucleotides and the representation information that applies to those four letters.



To complicate matters further, the recursive nature of representation information, which is usually composed of its own data and its own representation information, usually leads to a network of representation information objects. Since one of the main objectives of an OAIS-compliant system is to maintain information for a designated community, it must be able to interpret the knowledge base of its designated community in order to understand the minimum representation information that needs to be maintained. A choice has to be made between maintaining the minimum amount of representative information necessary for the designated community, and maintaining a greater amount of representative information enabling a wider community of consumers with a less specialized knowledge base to understand the product, which would mean broadening the definition of the designated community. Over time, the evolution of the designated community's knowledge base may require updates to the representation information to ensure continued understanding.

In the OAIS model, the Packaging Information consists of a conceptual container holding (1) Content Information, i.e. the object to be preserved, and its associated Representation Information described above, and (2) Preservation Description Information (PDI). PDI is needed to (1) ensure data is clearly described and (2) understand the environment in which it was created. PDI is divided into five types of preservation information called Provenance, Context, Reference, Fixity and Access Rights. In DLCM technology, the PDI is described using the DataCite and PREMIS metadata formats.

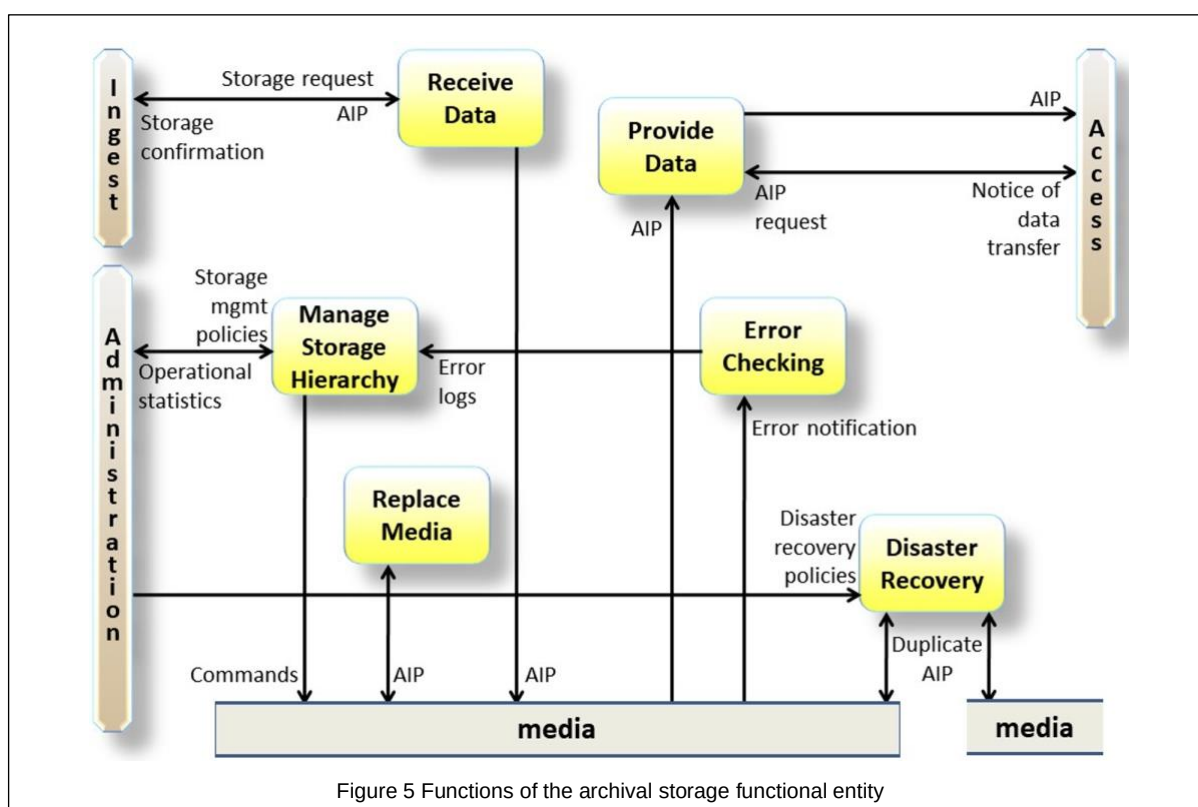
3.3 Archival Storage Functional Entity specifications

The main objective of our framework is to see how and to what extent it is possible to implement DNA as a means of long-term preservation of digital information in an OAIS-compliant system. It is essential to understand the different functions of archival storage in order to define how DNA can play a role in the AIP module. To do this, we need to examine the various components of the **Archival Storage** function, shown in Figure 2 and zoomed in on Figure 5 :

Receive Data: receives a storage request and an AIP from the ingestion entity and moves the AIP to permanent storage within the archive. Once the transfer is complete, this function sends a storage confirmation message to Ingest, including the AIP storage identification.

Manage Storage Hierarchy: this entity includes several functionalities, such as selection of the appropriate media on the basis of storage management policies, operational statistics, or ingestion instructions via the storage request. It will also comply with any special service levels required for AIP, or any special security measures required, and provide the appropriate level of protection for the AIP. These include on-line, off-line or near-line storage, required throughput rate, maximum bit error rate allowed, or special handling or backup procedures. It monitors error logs to ensure AIPs are not corrupted during transfers. This function also provides management with operational statistics summarizing available media inventory, available storage capacity in the various levels of the storage hierarchy, and utilization statistics.

Replace Media: ensures the system's competency to replicate the AIPs in the long term. Content Information and Preservation Description Information (PDI) must not be altered. However, the data



constituting the Packaging Information may be changed as long as it continues to perform the same function and there is a straightforward implementation that does not cause information loss.

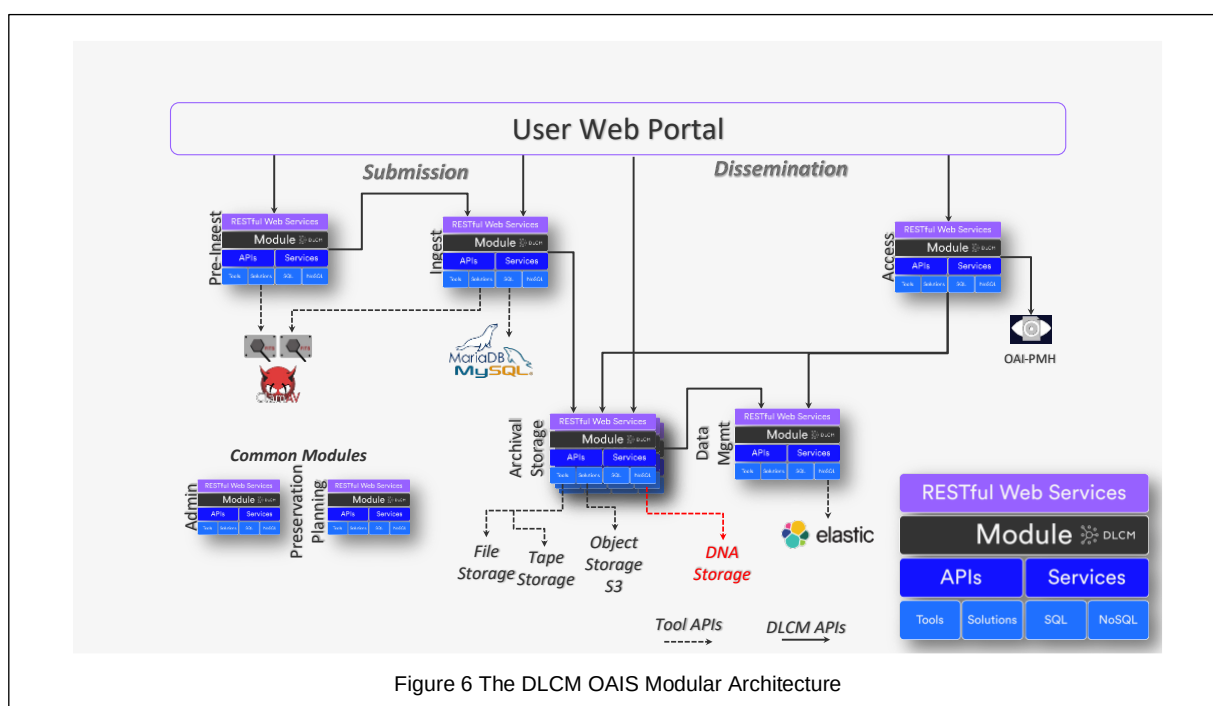
Error Checking: provides statistically acceptable assurance that no AIP elements are corrupted in the archive or during internal archive data transfer. This function requires that all archive hardware and software within the archive provide notification of potential errors, and that these errors are routed to standard error logs which are checked by archive storage staff. PDI fixity information provides some assurance that content information has not been altered during movement and access to the PDI. Similar information is needed to protect the PDI itself.

Disaster Recovery: provides a mechanism for duplicating the digital content of the archive collection and, for example, storing the duplicate in a physically separate facility. While duplication (replication) is one aspect of disaster recovery, the OAIS model also considers other crucial elements to ensure the preservation and accessibility of digital information in the face of unforeseen events or disasters. For instance, documentation and metadata, which are essential for understanding the context, structure, and characteristics of the digital objects, facilitate their recovery and reuse.

Provide Data: delivers copies of stored AIPs on access. This function receives an AIP request which identifies the requested AIP(s) and provides them on the requested media type or transfers them to a temporary storage area. This function also sends a data transfer notification to the access entity on completion of an order.

4. THE DLCM FRAMEWORK

Within the framework of the Swiss national DLCM project (2015-2021), two implementations of long-term preservation systems, yareta.unige.ch and olos.swiss, have been realized in a perspective of safeguarding research and patrimonial information (Burgi et al. 2017). The main competitive advantage of the DLCM technology comes from its modular and distributed architecture (Figure 6), as well as its strict compliance with the OAIS reference model (Burgi et al. 2022). Standard metadata ensure proper description of the data, i.e DataCite for descriptive metadata and PREMIS for long-term preservation. Persistent identifiers (based on the DOI scheme) and the OAI-PMH protocol for indexing and harvesting guarantee system interoperability. Various connectors to NFS, S3, and tape file systems enable the system to be easily integrated into standard storage infrastructures.



Within the DLCM technology, with the exception of the Preservation Planning part, all OAIS components are implemented (see Figure 6). In addition, to the three main OAIS entities (SIP, AIP, DIP), we added a pre-ingest module. This module allows for greater flexibility in data management by providing users with the ability to manipulate datasets prior to their final submission. Pre-ingestion takes place after the active data work phase, but before the archiving phase, which prevents any further modification.

In the context of the DNAMIC project, we intend to add a new connector to store the information in DNA oligonucleotides, accordingly to the general architecture represented in Figure 1. This connector is indicated by a dashed red line in Figure 6.

Because of the low throughput of DNA's synthesis and sequencing, we cannot envisage the use of DNA with as much flexibility as with current technologies (such as NFS for instance). We therefore propose DNA as a backup for patrimonial documents, and as a support system for disaster recovery. This would be an offline medium stored in a secure location, safe from hacking or technical failure.

5. SPECIFICATIONS OF THE ARCHIVAL STORAGE ENTITY APPLIED TO DNA

In this section, and referring to Figure 5, we present the various processes involved in archiving, accessing, updating, reloading (disaster recovery), checking fixity, and disposing of data using DNA as a carrier. Before describing these processes, it is important to understand the structure of the DLCM AIP.

5.1 Structure of the AIP

In DLCM technology, an AIP consists in a zip file containing:

- The files (i.e., content information)
- The associated descriptive metadata (i.e., Preservation Description Information)
- Some administrative metadata (i.e., Representation Information)

The preservation system maintains an associated database containing all the AIP information required for disaster recovery, the contents of which should ideally be fully recoverable from metadata.

Using DNA as a storage medium requires a specific set of additional parameters, including (but not limited to) :

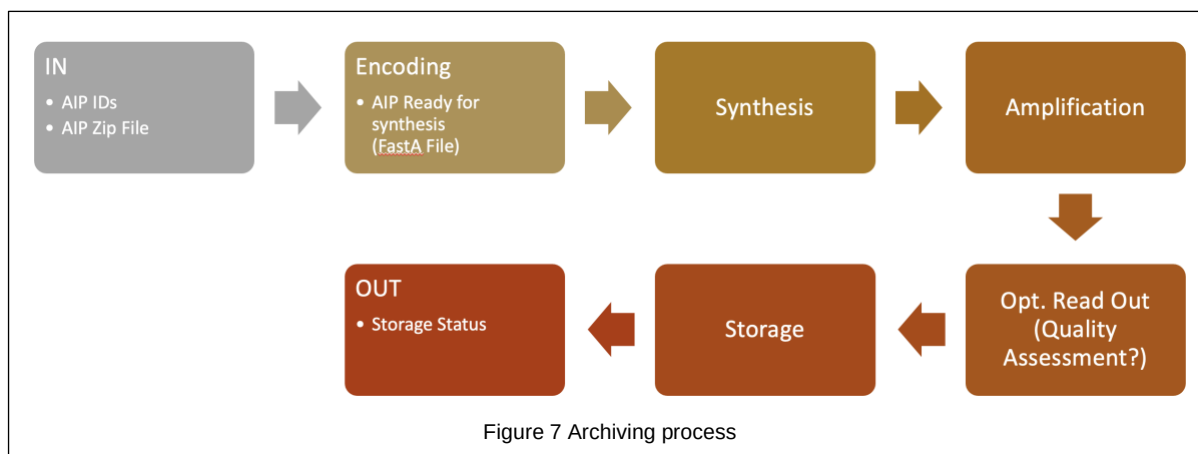
- Codec type, which comprises
 - The binary-to-ACGT conversion algorithm
 - The error correction algorithms
 - Strand length (in number of nucleotides)
 - Strand structure, which typically consists of:
 - Forward and backward primers
 - An index
 - Error correction information
 - The payload
 - Optionally a tag, which could indicate whether the strand consists of data or metadata
- The primers referring to a specific AIP
 - Suggestion: The use of checksums as a unique identifier for AIPs could serve as primers, identifying the strands pertaining to them. This would have the advantage of preserving the checksums within the primers. However, we need to choose hash codes that produce checksums that are not too large.

5.2 Archiving Process

The archiving workflow is represented in Figure 7 and is initiated on the basis of a *Storage Request* (Figure 5). From the OAIS point of view, the main steps consist in (1) extracting the identity of the Archival Information Package (AIP), which will be used as a primer to select related DNA strands, and (2) transferring the AIP to MILL in zip format. At this stage, however, it's important to decide whether the zip file should be encoded into DNA as such, or whether the metadata should be dissociated from the data content and stored separately in different flasks. One of the advantages of unbundling data and metadata is the ability to extract information without having to read the entire zip file. Also, certain parts of the metadata could even be encoded in a particular DNA structure (to be defined), enabling the information to be read without having to sequence the DNA.

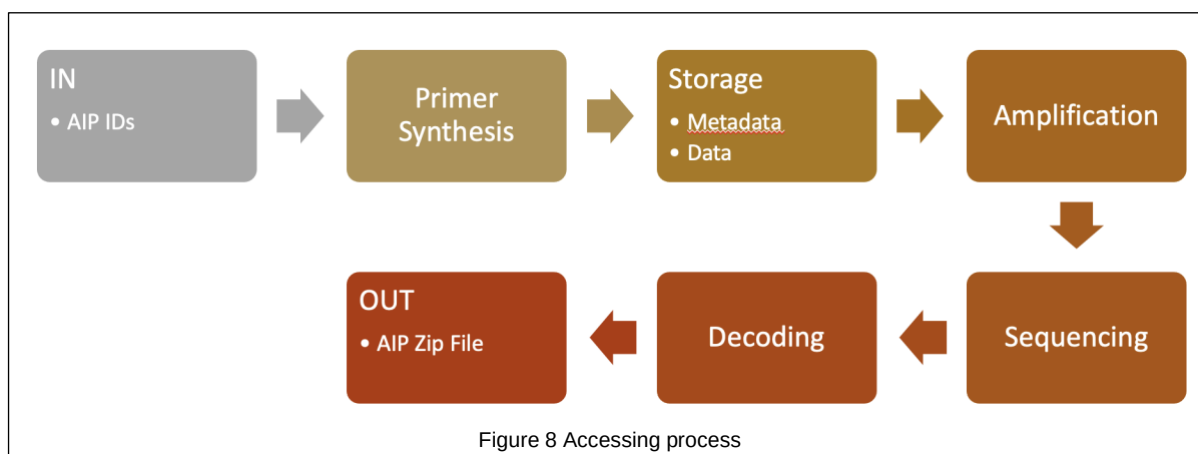
Another step, possibly optional, is the one preceding "Storage", which would consist in assessing the quality of the data after "Amplification". However, the mechanism for this quality control has yet to be determined.

Finally, the output of the archiving stage is a *"Storage Status"* transmitted to the OAIS log.



5.3 Accessing Process

The accessing workflow is shown in Figure 8. The starting point is an *"AIP request"* (Figure 5). Based on the identity of the AIP, logical mapping is performed to a specific primer. This primer either pre-exists in a DNA library or has to be synthesized. Note that in cases where metadata and data have been separated, they must be reassembled before amplification, followed by sequencing and decoding. The last step is to reconstitute the zip file, the condition for success being that no bit errors have occurred during decoding. Then, the *"Provide Data"* entity (Figure 5) delivers the AIP.



Since decoding the entire zip file structure can be very costly, as it requires the sequencing of a very large amount of DNA molecules, it could be interesting to be able to decode only the metadata in order to be informed about the contents of the zip file. Another option would be to have access to a specific file without having to read the whole zip file. This option requires the AIP zip structure to be suitably modified.

5.4 Fixity Checking Process

The error checking functionality is linked to the “*Manage Storage Hierarchy*” entity (Figure 5), whose function is to guarantee a certain level of service. However, in the case of DNA storage, to assess fixity, checksums need to be calculated, which would require repeating the entire “*Accessing Process*” illustrated in Figure 8. As the cost of these operations is prohibitive, we assume that the redundancy of the information encoded in the DNA and the number of copies of the DNA strands would guarantee that the information can be recovered without the need for checksum verification.

Another way of expressing this trade-off between cost and fixity check is to determine the period for which AIPs can remain uncontrolled. Since DNA is present in a certain number of copies, the system can tolerate a relatively long period between checks. When the threshold of the minimum number of readable copies is reached, determined by the DNA half-life, the data is replicated in a new batch of DNA. It is worth noting that when accessing an AIP (figure 8), the ability to create the zip file requires that no bits are false, which is equivalent to verifying a checksum at bit level.

5.5 Disaster Recovery

To comply with the OAIS standard, we must address the issue of disaster recovery (see Figure 5). To be able to do that, we must at a minimum know the codec type. But to read the codec type we must know how to decode DNA. Two possible solutions to this recursive problem:

- (1) The codec type is encoded in a DNA strand based on a very simple DNA encoding. This solution is referred to as “Sector 0”. This solution needs to manage the associated primer suitable for amplifying the DNA strand containing the codec type.
- (2) The codec type is encoded in **structural DNA**, for example in specific DNA configurations (origami or other kind of structures). However, in this case the structure must be simple enough to be easily understood and readable.

In both cases, the codec type could be a number referring to a format described in a PRONOM-type database (with a specific PUID fmt). It can be argued that maintaining such a database over the very long term is illusory. Yet, if we lose such a database, even the files contained in the AIP would no longer be understandable.

The process corresponding to “Disaster Recovery” is illustrated in Figure 9. In this case there no entry. Once the codec type has been identified, one option would be to have a primer library, which would retrieve either the list of AIPs with their associated metadata, or all AIPs. Amplification of all AIPs should be simple: one could probably ligate primers on all sequences and amplify.

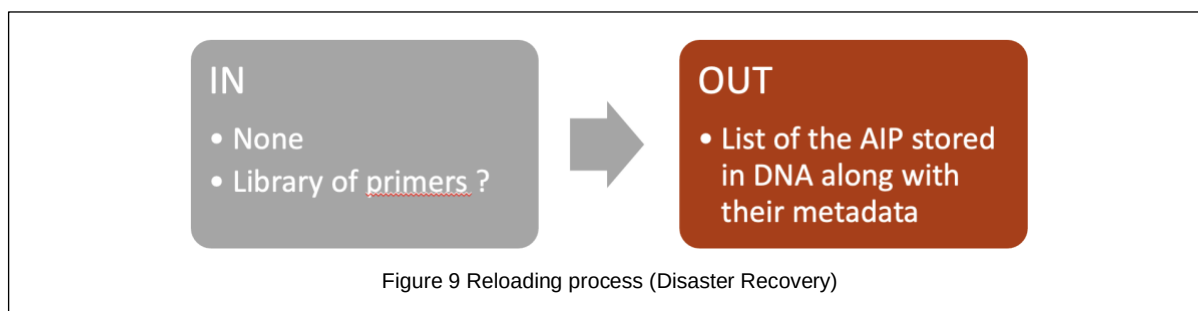
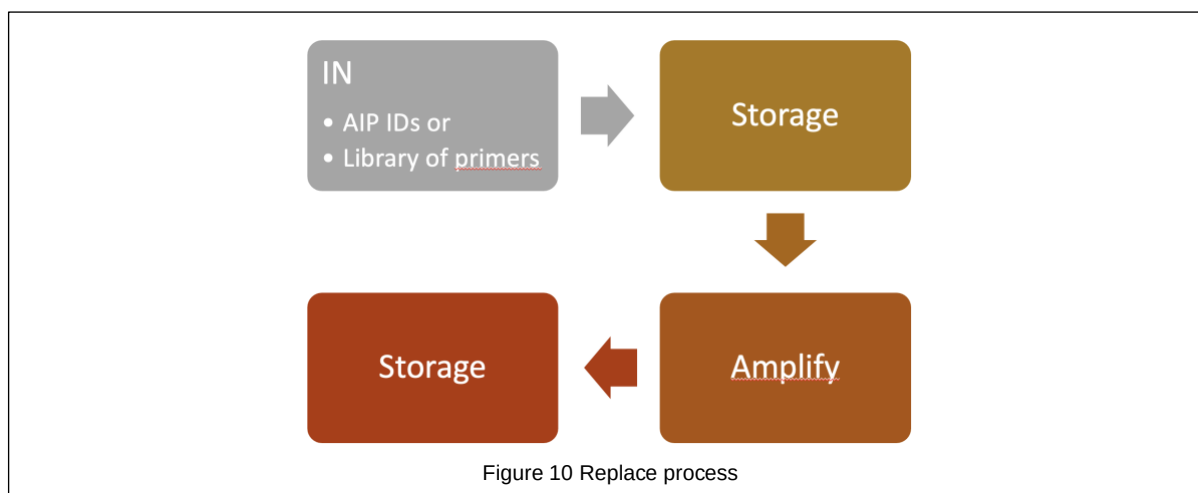


Figure 9 Reloading process (Disaster Recovery)

5.6 Replace process

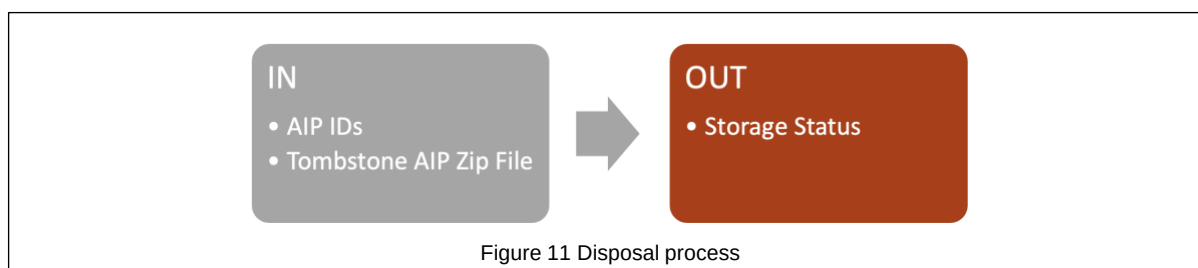
At a certain stage, particularly when the “Accessing Process” has been used a certain number of times (this number being determined on the basis of DNA concentration), it would be necessary to regenerate the DNA batch in order to guarantee a sufficient number of DNA strands for future accesses. As input, either the list of AIP’s is provided, or primer library is available for AIP amplification (Figure 10). However, we need to be aware that PCR amplification introduces bias, so we need to be careful about the number of times we amplify the strands, as at some point certain strands will be under-represented.



5.7 Disposal process

This process represents the ultimate stage in the data lifecycle, since it consists of eliminating the data. As the OAIS model does not, in principle, authorize the destruction of an archive, data disposal means retaining the metadata and deleting the data. This is the equivalent to placing a tombstone in the archive, as illustrated in Figure 11.

From the DNA’s point of view, physical disposal of the data amounts to eliminating all DNA strands relating to the removed AIP. If the AIP is isolated in a flask, this operation is straightforward. Otherwise, the removal of specific strands will require specific biological processes, not yet necessarily identified, to ensure that no trace remains (especially in the case of confidential information).



6. IMPLEMENTATION OF A SIMPLIFIED SCENARIO

The previous section presented general concepts without going into the details of implementation. In this section, we present in more detail the workflows between DLCM technology (OLOS) and the microfactory in the case where the following simplifications are applied:

- No nanostructures are implemented for storing metadata
- The AIPs are maintained as such and are not divided into their components
- The AIP format is a binary zip file that will be encoded in DNA

- As an option, certain metadata can be encoded and physically stored separately

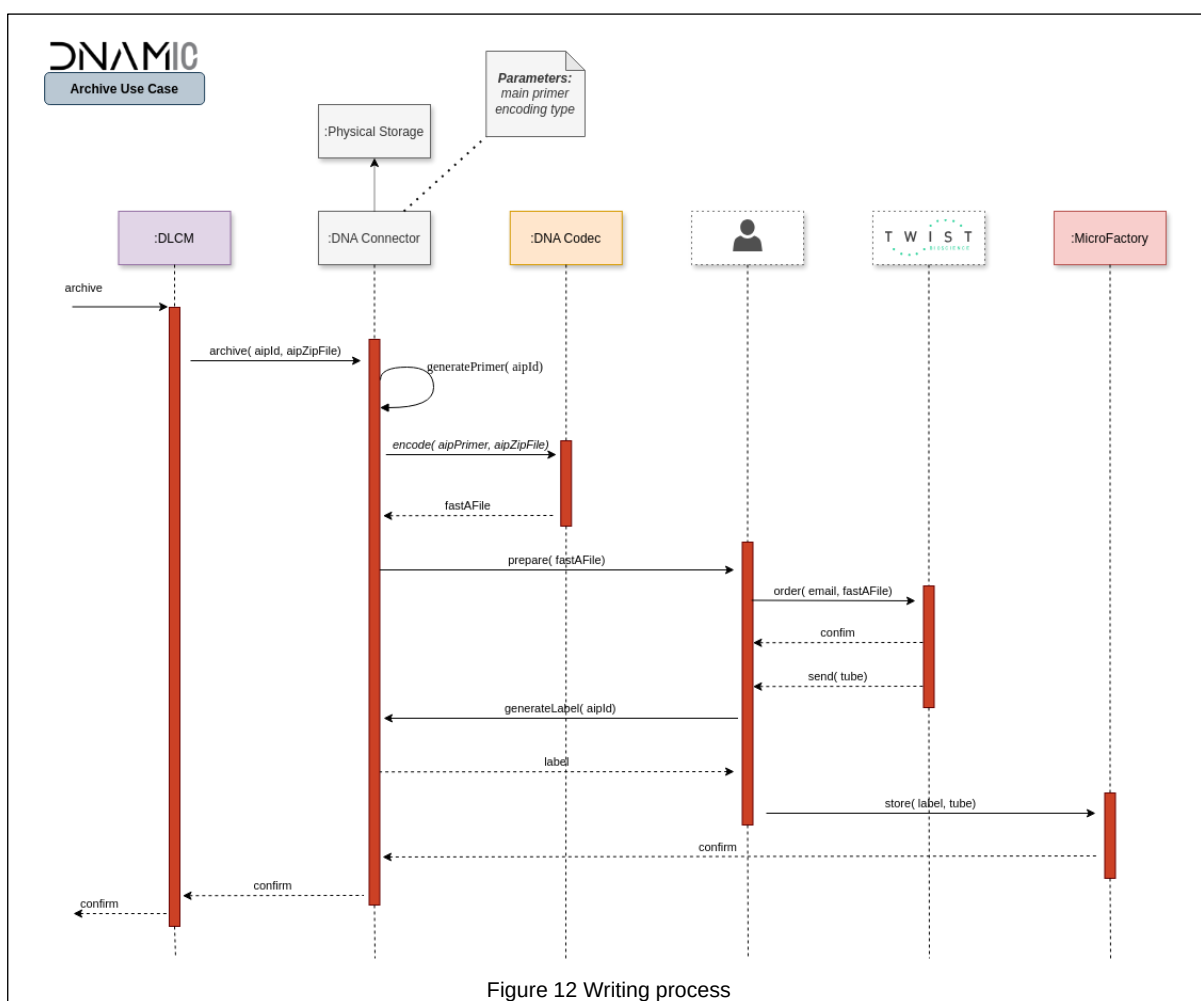
Based on these simplifications (which correspond to version 1 of the end-to-end solution), we now consider 3 processes: (1) writing an archive; (2) reading an archive; (3) disaster recovery.

6.1 Writing an archive

The writing process consists of receiving data (Figure 5) and sending it to the DNA medium. This operation is performed by the DNA connector, represented in Figure 12. Note that the DNA connector and codec are part of the software layers installed on a server which is not necessarily located close to the microfactory. Communications between this server and the microfactory are established via the TCP/IP protocol.

As it can be seen in this figure, the orchestration of all the process is done by the connector. The different steps are the following:

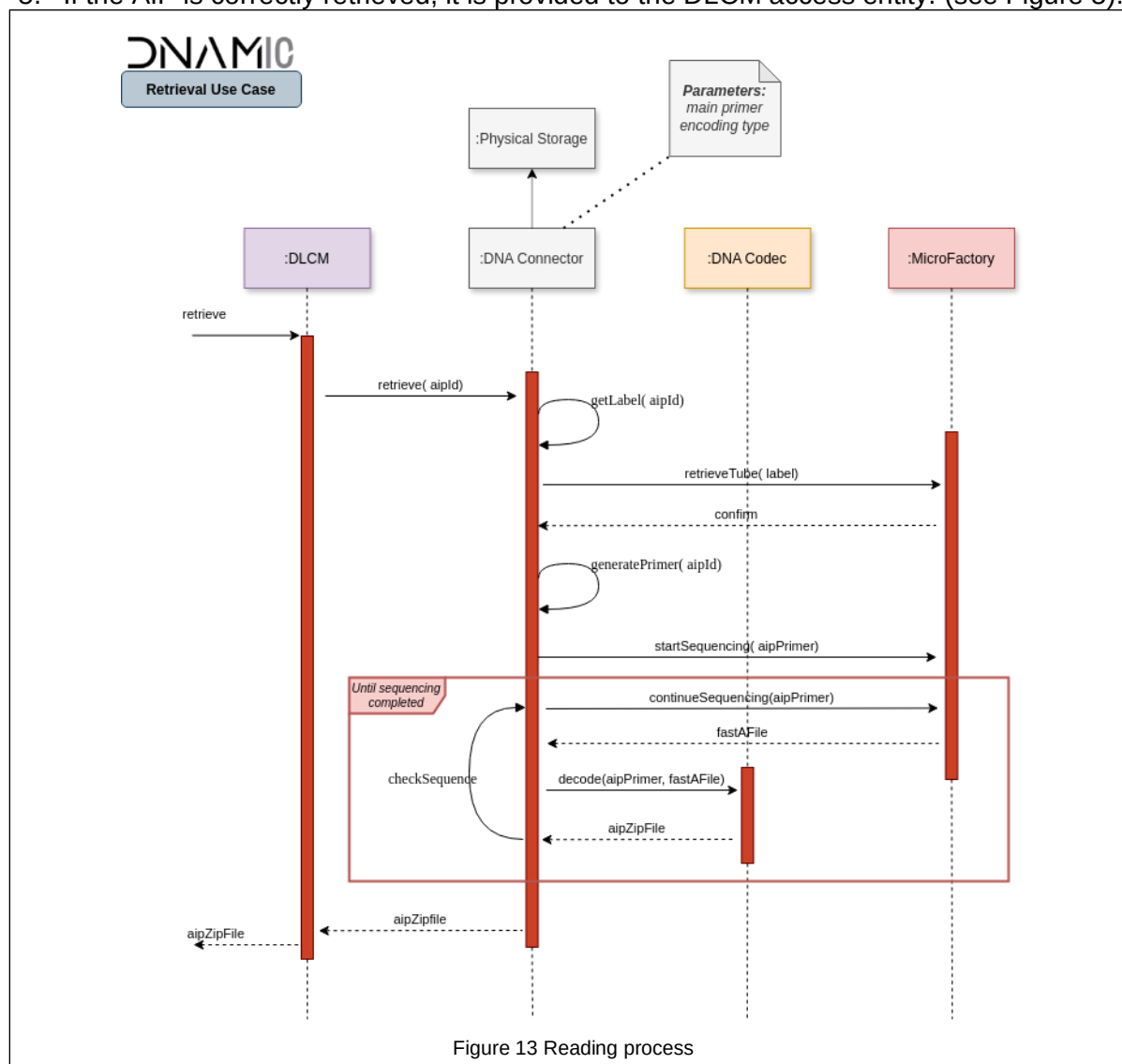
1. A primer matching only the AIP is generated. The hash of the AIP file could be used as the primer.
2. The AIP and the primer are sent to the DNA codec to be transformed into DNA strands. These strands are written to a file in FastA. This file is sent back to the connector.
3. The FastA file is prepared and sent to Twist Bioscience by email. This operation requires human intervention at different stages. Firstly, for biosafety reasons, Twist may request additional information about the object of the order. Secondly, the flask containing the DNA strands must be received in St-Imier. Thirdly, a label (QR-code) generated by the connector is printed and affixed to the flask. Fourthly, the flask is placed in the microfactory to be processed.
4. Finally, the connector receives confirmation of the storage status, which is in turn sent to the OAIS administrative entity (see Figure 5).



6.2 Reading an archive

In the simplified version (without nanostructures), the reading process consists in retrieving a set of strands corresponding to a specific AIP, identified by a primer. Again, the whole process is orchestrated by the DNA connector. The different steps, illustrated in Figure 13, are:

1. Based on the DLCM retrieval command, the DNA connector fetches the corresponding label to identify the flask. The label is sent to the microfactory, and a confirmation message is received in return.
2. The primer corresponding to the AIP identity is retrieved. This primer is sent to the microfactory to initiate DNA sequencing, a process that needs intermediary steps at the level of the microfactory (i.e., pipetting and PCR).
3. The output of the sequencer is a FastA file containing DNA strands. This file is sent to the codec corresponding to the encoding type.
4. At this point, the codec should attempt to recover the AIP based on the first FastA file received. However, there is no guarantee that all DNA strands are represented in the file. Consequently, the codec may fail in this task. The reason may be twofold: (a) there are unrecoverable errors, and (b) there are missing strands because the sequencer stopped sequencing too early. In both cases, the connector must send a command to the sequencer to resume sequencing. The resulting new FastA file is then received and added to the previous one. The completed file is sent back to the codec for another attempt. If the codec fails to recover the AIP, the process can be repeated. The question is: what would be the criteria for stopping the feedback loop?
5. If the AIP is correctly retrieved, it is provided to the DLCM access entity. (see Figure 5).



Regarding point 4 above, and to remain pragmatic, we propose that the sequencing could run until the end (i.e., 72 hours to be exact with the current NGS technology) and then decoding would start. And in a future iteration of the framework, we will be adding real-time data processing – a solution that will require further discussions with our partners and the codec designers.

6.3 Disaster recovery

This scenario consists in rebuilding all AIPs in the OLOS repository. The different steps are as follows:

1. All labels printed on the flasks are collected. One hypothesis is that the label contains (a) an indication on the codec and (2) specifies a general primer that allows amplification of all the DNA strands contained in the flask. Another alternative would be to use a sector 0 to retrieve this information, particularly if even the labels become illegible.
2. All strands corresponding to the primer are sequenced. Similar to the reading process, we have the option of either sequencing to the end or, in a more sophisticated version, adding real-time data processing (checkSequence in the diagram). All recovered AIPs (zip files) are sent to OLOS.

It should be noted that the disaster recovery process involves different constraints with regard to self-description. A specific study is underway on this subject, and the first results are expected in September 2024, and will be described in version 2 of this document.

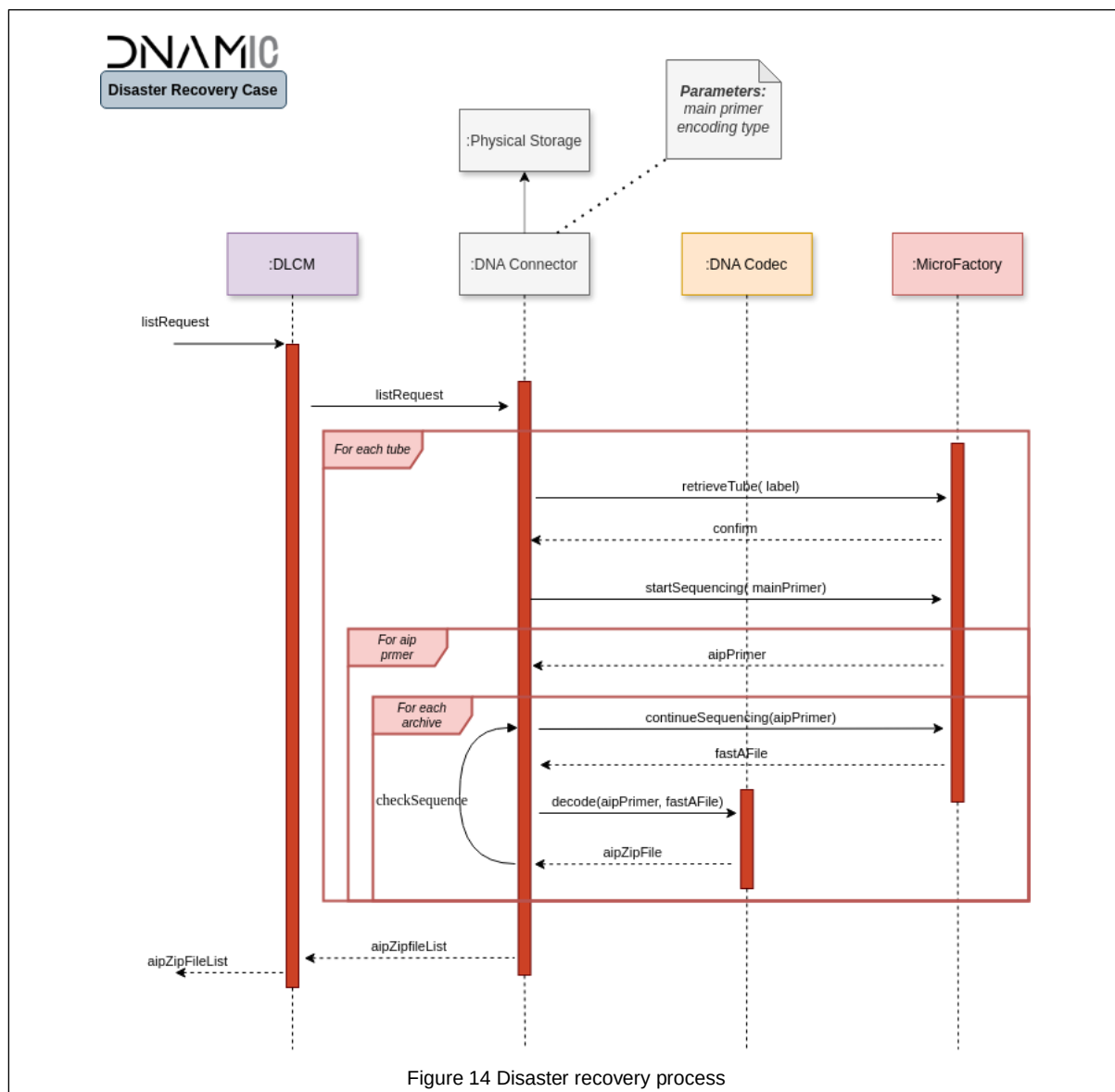


Figure 14 Disaster recovery process

7. CONCLUSIONS

This first version (v1.0) of Deliverable 2.1 aims to define the OAIS context of the DNAMIC project. After an introduction to OAIS concepts and a macro-specification of the processes involved, we proposed a more detailed implementation in section 6. The diagrams in Figures 12 to 14 presented in Section 6 should help clarify the MILL interface matrix between DLCM and the microfactory. Consequently, this version attempts to describe the processes involved at a level adequate for the microfactory to function properly. There are still some unresolved issues, such as the feedback loop between codec and sequencer, or the self-description mechanisms. Consequently, the next steps will require a deeper understanding of the genomic constraints on DNA reading and recovering. In addition, when we're ready to move from the simplified version to the more complete one that includes nanostructures, Figures 12 to 14 will need to be updated. Using the information represented in nanostructures could also help solve the unresolved issues mentioned above.

8. REFERENCES

Burgi, P.-Y., Blumer, E., and Makhlouf-Shabou, B. (2017) Research data management in Switzerland: National efforts to guarantee the sustainability of research outputs. *IFLA Journal*, **43(1)**, pp. 5–21. doi: 10.1177/0340035216678238.

Burgi P.-Y., Krause, J., Meiser, L., Andriamahady, D., Cazeaux H., Friha, L., Makhlouf Shabou, B. (2022) OAIS-Compliant digital archiving of research and patrimonial data in DNA. Proceedings of the 18th International Conference on Digital Preservation (iPres), pp. 220-224, September 12 - 16, 2022, Glasgow, Scotland.

CCSDS (The Consultative Committee for Space Data Systems), 2019. [Reference Model for an Open Archival Information System \(OAIS\)](#) – Pink (pre-magenta) Book [online]. 2019. [Accessed 18 January 2024].