SHIFT

MetamorphoSis of cultural Heritage
Into augmented hypermedia assets
For enhanced accessibiliTy
and inclusion

Funded by
the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

DOCUMENT INFO

Document ID:	**D5.1 - System architecture**
Version date:	03/01/2024
Total number of pages:	110
Abstract:	The document contains the project's technical requirements and proposes architecture for the End-To-End Platform Architecture. It is the result of task T5.1
Keywords	SHIFT, software architecture, data mesh, peer-to-peer, nodes, workspaces, artifacts, cultural heritage.

AUTHORS

Name	**Organization**	**Role**
Iacob Crucianu	SIMAVI	Document maintainer
Krishna G. Chandramouli	QMUL	contributor
Anika Spiesberger	UAU	contributor
Dionyssos Kounadis-Bastian	AUD	contributor
George Margetis **Katerina Valakou**	FORTH	contributor
Philippos Orphanoudakis	MDS	contributor

REVIEWERS

Name	**Organization**	**Role**
Felix Burkhardt	AUD	reviewer
Martin Zamorano	ERC	reviewer

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

VERSION HISTORY

Version	Description	Date
0.1	1st complete draft	23/05/2023
0.2	Place in a new template	24/05/2023
0.3	QMUL Contribution to Module Description	17/08/2023
0.4	AUD Contribution to Module Description	17/08/2023
0.5	UAU Contribution to Module Description	22/08/2023
0.6	FORTH Contribution to Module Description	24/08/2023
0.7	MDS Contribution to Module Description	28/09/2023
0.8	SIMAVI cross-check and contribution to general chapters	28/09/2023
1.0	SIMAVI Close and send the final version	30/09/2023
2.01	SIMAVI Treatment of the review remarks related to the modules and tools	27/12/2023
2.02	SIMAVI Treatment of the review remarks related to use cases and tools	29/12/2023

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

EXECUTIVE SUMMARY

We are defining the project's technical requirements and proposing architecture for the End-To-End Platform Architecture.

The work is based on the outputs of the other work packages, i.e. WP2, WP3, and WP4. It uses the output of the deliverables D1.1 (SHIFT-D1.1, 2023) - SHIFT requirements, user evaluation guidelines, and acceptance metrics and of the draft deliverables (SHIFT-D2.1, 2023), (SHIFT-D3.1, 2023), (SHIFT-D3.2, 2023), (SHIFT-D3.3, 2023), (SHIFT-D4.1, 2023).

The Concept defined for the End-To-End Platform Architecture is based on the Data Mesh concept as it is defined in Chapter 4.2.

Data referring to the artifacts governed by the platform is placed in workspaces. A workspace represents a subdomain of data and knowledge. The content of a workspace is built by directly placing inputs, or by calling the software tools available in the system. Each software tool is a collection of modules, working together for the same business objective.

The implementation of the concept is proposed to be a distributed system around a Peer-to-peer (P2P) network.

The main physical elements of the Platform backend are Nodes. A Node offers data storage, processing, retrieval, validation, collaboration, governance, and distribution functionality. Nodes can be added or removed at any moment from the system. In the case of such operations, the fully distributed system is synchronized automatically. Full Nodes operates all the software tools and can work in standalone mode also. This offers a high availability of the system. All the data or knowledge is exposed via APIs, presented at the level of each node. A simple node contains references to data and indexes using hashes to data. It needs access to full nodes to get all the data.

The document starts with the identification of the technical requirements, according to the DOA (SHIFT Consortium, 2022), and continues with the analysis of actual systems used in the field. After a presentation of the state of the art defining complex architectures, we are using a 4+1 methodology, combined with Data-Driven System Architecture (DDSA) pattern to define the proposed system architecture.

The 4+1 views (Logical, Process, Physical, Development, Usecase) are applied to data components to show different perspectives of a Data Product.

The architecture of the whole system is defined to allow the fulfillment of the objectives of the project and validate this on the four use cases defined.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

Table of Contents

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

List of Figures

List of Tables

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

Abbreviations and Acronyms

ACRONYM	DESCRIPTION
API	Application Programming Interface
CI	Continuous Integration
CPU	Central Processing Unit
CRF	Conditional Random Field.
D	Deliverable
GUI	Graphical User Interface
JEE	Java Enterprise Edition
JSON	JavaScript Object Notation
NoSQL	Not only Structured Query Language
OS	Operating System
OSD	Object Storage Device
OT	Operational Technology
RDBMS	Relational Database Management System
RAM	Random-Access Memory
RBN	Radial Basis Network
RNN	Recurrent Neural Networks
REST	Representational state transfer
SIEM	Security Information and Event Management
SSL	Secure Sockets Layer
T	Task
VM	Virtual Machine
VPN	Virtual Private Network
WP	Work Package

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

1. Introduction

1.1. Scope

The present document contains the End-to-end Platform Architecture, Specifications, and Development lifecycle with detailed component sub-architectures that fully cover all the functionality of the SHIFT platform. The document describes the proposed prototyping approach for designing the modular SHIFT Platform. It is based on the user requirements specified in the User Experience Stories and partner input. It will form the basis of the planning of the development phase and will be updated using the results from the various development iterations to come to a complete architectural design. It is intended for review by members of the project, notably the system architects of the development team. This design follows the Rapid Application Development (RAD) / rapid prototyping lifecycle.

The document explains how the proposed platform will deliver a set of technological tools, loosely coupled that offers cultural heritage institutions the necessary impetus to stimulate growth and embrace the latest innovations in artificial intelligence, machine learning, multi-modal data processing, digital content transformation methodologies, semantic representation, linguistic analysis of historical records, and the use of haptics interfaces to effectively and efficiently communicate new experiences to all citizens. (SHIFT Consortium, 2022)

1.2. Structure of the report

After this chapter (Introduction), the deliverable is organized as follows:

Chapter 2. Objectives. Presents how the objectives and the ambition of the project are reflected in the technical specifications and the system architecture.

Chapter 3. Methodology. Presents the methodologies used to define the system architecture. First, a study on possible choices is done. Next, based on the pros and cons, the methodologies are selected. The selected methodologies are a combination of 4+1 (Krunken) and Data-Driven System Architecture (DDSA).

Chapter 4. Approach. The main conceptual elements selected for the definition of the architecture are presented. They include a description of the 4+1 methodology, the concepts of Data Mesh, Knowledge Mesh, and Distributed Data and Knowledge Mesh (DDKM).

Chapter 5. End-to-End-Architecture definition. Presents the architecture of the platform, using the five Views: Logical, Process, Physical, Development, and Use Case. It is the most consistent part of the document as it reflects the platform as it is planned to be developed and implemented.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

2. Objectives

In the present chapter, we are presenting the objectives of the project from the technical architecture perspective. We are analyzing existing solutions that cover some of the elements defined for the SHIFT project, we are identifying the missing elements, and we are proposing the main requirements and also the main approach for the final product.

2.1. Ambition

Based on the DOA (SHIFT Consortium, 2022), the ambition of the project, is to advance beyond the state-of-the-art in ***three directions***:

- Improving accessibility.
- Improving social richness.
- Improving appeal.

2.2. Objectives

Based on the DOA (SHIFT Consortium, 2022) the objectives of the project are:

- Adoption of digital transformation strategy within cultural heritage institutions
- Tools and algorithms to revitalize historical and cultural high-value content
- Enriching user experiences for interacting with cultural assets
- Enhance the preservation of historical archives using contemporary language models
- Development of accessibility tools and methodologies in compliance with international standards
- Implementation of inclusion by design methodologies
- Contribution to international standards to exchange metadata models with cultural institutions and copyright protection of content ownership
- Dissemination and communication strategies for wider-scale adoption of SHIFT results across cultural and creative industries

2.2.1. SCIENTIFIC AND TECHNICAL OBJECTIVES

The general objectives of the project are translated into technical objectives, affecting the design of the system architecture, and referring to the inclusion in the platform of:

- Computer vision tools and algorithms to revitalize historical and cultural high-value content.
- Innovations in linguistic processing algorithms to enhance access and representation of cultural content.
- Haptic interfaces for enriched user experience.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

- Text-to-audio digital asset transformation methodologies for enriching the user experience.
- An accessible framework of inclusive museum exhibits for 3D digital asset perception.
- A semantic framework for long-term preservation of historical archives.
- Innovation in accessibility standards for interacting with cultural heritage.

All those tools will be accommodated in the platform and give value to the end users.

2.2.2.MAIN TECHNICAL REQUESTED CHARACTERISTICS

Derived from the objectives of the project mentioned in DOA (SHIFT Consortium, 2022), and considering the ambition of the project presented in the previous subchapter, we have derived several technical characteristics for the proposed product of this project.

1. **Data collection**: The system should be able to collect data using an adequate user interface, able to interact with users involved in the SHIFT project.
2. **Data storage and archiving**: The system should be able to manage different types of data, and models, collected and stored in a wide variety of formats.
3. **Data retrieval and discovery**: Teams should be able to search for and easily discover and access the data they need within the system.
4. **Data audit and tracking**: The system should provide tools for tracking the origin and history of data, as well as any transformations or processes it has undergone.
5. **Data presentation**: Adequate tools (Audio, Video, text) will be used for data presentation, including haptic tools.
6. **API access**: The system should provide APIs for accessing and integrating data from other sources and applications.
7. **Decentralization**: The Platform should allow the distribution in several peer-to-peer nodes, and the teams are responsible for managing their knowledge domains and are empowered to make decisions about how to manage and share their data.
8. **Domain ownership**: In the Platform, each data domain is owned by a specific team, which is responsible for managing the data within that domain.
9. **Standardization**: The system should be developed to implement the existing standards for data storage, retrieval, and distribution.
10. **Data governance**: The Platform should provide tools for managing data quality, privacy, and security.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

3. Methodology

In this chapter, we are presenting several methodologies available for defining the software architecture of a system, and based on the specificity of SHIFT, we are selecting the ones that best match our needs.

The methodology is based on the SIMAVI ISO 9001 certified methodology, which includes procedures and work instructions.

For the completeness of the document and better understanding, we are including here the methodological elements from SIMAVI documents (SIMAVI, 2021).

3.1. Defining software architecture methodology

In the last decades, several methodologies used to define the software architecture for large systems were defined and used. We are using two categories of such methodologies, defined on the object of definition.

The first category refers to Views and Perspectives. The second category refers to architectural patterns. The categories mentioned here overlap, as they offer different points of view on the same target: The software architecture.

3.2. View-based approach

The view-based approach to software architecture is a method for designing and documenting the architecture of a software system by focusing on views of the system and treating each view from different perspectives. It is a matrix treatment of the architecture where each view provides a different representation of the system as they are addressed by different stakeholders (users, developers, engineers, managers, administrators). For each view, different perspectives are addressed (for example security, availability, performance, etc.)

3.2.1. STATE OF THE ART

Several methodologies use a view-based approach to software architecture. Some of the most popular ones include:

Kruchten 4+1 View Model: The Methodology was developed by Philippe Kruchten (Kruchten, 1995). It uses five views as the 4+1:

"The 4+1 System Architecture Methodology is a software development approach that provides a comprehensive view of a software system by describing it from five different perspectives or views. The five views are:

1. **Logical View**: Describes the functionality of the system from a user's perspective. It can be represented in UML with use cases, class diagrams, and sequence diagrams.
2. **Process View**: Describes the concurrency and synchronization aspects of the system. It can be represented in UML with activity diagrams and state-transition diagrams.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

3. **Physical View**: Describes the physical deployment of the system, including hardware and network components. It can be represented in UML with deployment diagrams, component diagrams, or network diagrams.
4. **Development View**: Describes the organization of the software components and the development process. It can be represented in UML with package diagrams and component diagrams.
5. **Scenario View**: Illustrates the system's behavior in specific scenarios or situations. It can be represented in UML with use cases or activity diagrams.

" (Kruchten, 1995)

Rational Unified Process (RUP) (Per Kroll, 2003)**:** This methodology uses a set of 4 views to describe the architecture of a software system. Apart from the others, RUP is an iterative and incremental methodology that emphasizes the importance of architecture-centric development.

"The views defined by RUP are:

1. **Use Case View**: This view describes the functional requirements of the system from the perspective of its users. It can be represented in UML with use case diagrams and scenarios that illustrate how the system is used to accomplish specific tasks.
2. **Logical View:** This view describes the system's software architecture in terms of its components, classes, and interfaces. It can be represented in UML with class diagrams, object diagrams, and sequence diagrams that illustrate how the system's components interact with each other.
3. **Process View**: This view describes the system's dynamic behavior in terms of its processes and threads. It can be represented in UML with activity diagrams and state diagrams that illustrate how the system processes data and responds to events.
4. **Physical View**: This view describes the system's hardware and network architecture in terms of its nodes and connections. It can be represented in UML with deployment diagrams and component diagrams that illustrate how the system's components are deployed and connected.

In addition to these four views, RUP also defines a fifth view called the "implementation view" which describes how the system is implemented in terms of its code, data structures, and algorithms. This view is not considered a separate view in the 4+1 view model but is instead integrated into the other views.

" (Per Kroll, 2003)

Viewpoints and Perspectives (V&P) (Nick Rozanski, 2005): This methodology uses viewpoints to describe the architecture of a software system, where a viewpoint is a set of conventions for constructing and interpreting views. V&P emphasizes the importance of defining clear and concise viewpoints that are tailored to the needs of the stakeholders

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

"

The following 7 viewpoints are defined:

1. **Context viewpoint**: Describes the relationships, dependencies, and interactions between the system and its environment (the people, systems, and external entities that it interacts with
2. **Functional Viewpoint**: Defines the system's architecturally significant functional elements, the responsibilities of each, the interfaces they offer, and the dependencies between elements.
3. **Information Viewpoint**: Defines the structure of the system's stored and transient information (e.g. databases and message schemas) and how related aspects such as information ownership, flow, currency, latency, and retention will be addressed.
4. **Concurrency Viewpoint:** Defines the set of runtime system elements (such as operating system processes) into which the system's functional elements are packaged.
5. **Development Viewpoint**: Defines any constraints on the software development process that are required by the architecture. This includes the system's module organization, common processing that all modules must implement, any required standardization of design, coding, and testing, and the organization of the system's code line.
6. **Deployment Viewpoint**: Defines the important characteristics of the system's operational deployment environment. This view includes the details of the processing nodes that the system requires for its installation (i.e. its runtime platform), the software dependencies on each node (such as required libraries), and details of the underlying network that the system will require.
7. **Operational Viewpoint**: Defines how the system will be installed into its production environment, how data and users will be migrated to it, and how it will be configured, managed, monitored, controlled, and supported once this is achieved. The aim of the information in this view is to show how the operational environment is to be created and maintained, rather than to define detailed instructions or procedures.

The following perspectives are defined:

1. Performance and Scalability
2. Security
3. Availability and Resilience
4. Evolution
5. Accessibility
6. Internationalization
7. Regulation
8. Usability

" (Nick Rozanski, 2005)

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

3.3. Architectural patterns

Architectural patterns are another way of organizing software architectures. It focuses mainly on the components used by the architecture, viewed from an IT perspective.

3.3.1. STATE OF THE ART

There are several architectural patterns for software architecture that are widely used in the software industry (Richards, 2015). Some of the most popular ones include:

"

Object-Oriented Analysis and Design (OOAD): This methodology focuses on creating a software architecture by identifying the objects and their relationships in a system, and then designing a solution around these objects.

Service-Oriented Architecture (SOA): This methodology focuses on creating a system based on services, where each service is a self-contained unit of functionality that can be reused across multiple applications.

Model-Driven Architecture (MDA): This methodology uses models to create a software architecture, where the models are representations of the software system at different levels of abstraction.

Agile Architecture: This methodology emphasizes the importance of collaboration, feedback, and continuous improvement in software architecture design. It advocates for an iterative and incremental approach to architecture, where the architecture evolves over time based on the changing requirements of the system.

Domain-Driven Design (DDD): This methodology focuses on creating a stop architecture that is aligned with the business domain. It emphasizes the importance of understanding the domain and modeling it in the software architecture.

Event-Driven Architecture (EDA): This methodology focuses on creating a system where the components communicate with each other by producing and consuming events. It emphasizes the decoupling of components and the ability to handle large volumes of events.

Data-Driven Software Architecture (DDSA): This methodology uses data analysis and machine learning techniques to inform software architecture decisions. DDSA involves collecting and analyzing data from various sources, such as code repositories, issue trackers, and user feedback, to identify patterns and trends that can inform architecture decisions.

" (Richards, 2015)

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

3.4. Selected methodology

For SHIFT we have selected the View-based methodology **Kruchten 4+1 View Model,** and for architectural pattern the **Data-Driven Software Architecture (DDSA):**

Kruchten 4+1 View Model mainly focuses on the architecture and offers a clear view for all stakeholders.

DDSA aims to provide objective data-driven insights into the performance and quality of the software system, which can help architects make informed decisions about trade-offs, such as between performance and maintainability. It can also help identify areas of the system that may require refactoring or redesign.

DDSA is a relatively new approach and is still evolving. It requires specialized skills in data analysis and machine learning, and there are currently limited tools and frameworks available to support it. However, it has the potential to be a powerful methodology for software architecture, particularly for large, complex systems with significant amounts of data.

The main elements of DDSA include:

"

Data sources: DDSA relies on a variety of data sources to inform the architecture design process. These data sources can include user feedback, system logs, performance metrics, and other types of data that provide insights into how the system is being used and where improvements can be made.

Data analysis: Once the data has been collected, it needs to be analyzed to identify patterns, trends, and areas where improvements can be made. This analysis can be done using a variety of techniques, including data mining, machine learning, and statistical analysis.

Architecture design: Based on the insights gained from the data analysis, the software architecture can be designed or refined. The design process should take into account the specific needs and requirements of the system, as well as any constraints or limitations that may be present.

Implementation: Once the architecture design is complete, the system can be implemented using standard software development techniques. The implementation should be guided by the architecture design but can also be informed by ongoing data analysis and user feedback.

Monitoring and feedback: Finally, DDSA requires ongoing monitoring of the system and user feedback to ensure that it continues to meet the needs of its users. This feedback can be used to inform further data analysis and architecture refinement, creating a continuous feedback loop that improves the system over time.

Overall, the main elements of DDSA are focused on using data to inform every step of the software architecture design process, from initial analysis to ongoing

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

monitoring and feedback. By using data in this way, DDSA can help to create software systems that are more effective, efficient, and user-friendly.

" (Richards, 2015)

A detailed description and motivation of the two selected methodologies are contained in the next chapter.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

4. Approach

In this chapter, we explain the methodologies and the elements used for the definition of the system architecture.

First, an explanation and motivation for the methodologies used are presented. Then the main elements defining the final product are explained.

4.1. 4+1 Views and DDSA approach

For the definition of the architecture, we are using the 4+1 methodology applied to create a Data-Driven Software Architecture.

That means, that the same Data Driven architecture will be presented from 5 (4+1) stakeholders' perspective.

The central focus is on a data product as the main elements for this system refer to a large quantity of data, managed, processed, and delivered for final users.

In fact, it will be a data representation, explained from 5 points of view.

4.1.1. MOTIVATION

The selection of the 4+1 methodology is explained by some elements specific to this methodology, which better fits the objectives and the requirements of the project. They include:

Focus: The 4+1 view model is primarily focused on software architecture, while RUP is a more comprehensive methodology that covers all aspects of software development, including project management, requirements gathering, testing, and deployment.

Simplicity: The 4+1 view model defines five views (logical, process, physical, development, and use case), which are more concise and easier to understand.

Sequence: The 4+1 view model places more emphasis on the use case view and considers it to be the central view, while RUP places more emphasis on the logical view and considers it to be the foundation for the other views.

Formality: The 4+1 view model is a more formal methodology that uses UML diagrams to represent the views of the system. V&P is a less formal methodology that can use a variety of techniques to communicate the architecture, including diagrams, narratives, and prototypes.

Usage: The 4+1 view model is a very well-known methodology, with proven results in the last decades.

The selection of DDSA comes from the fact that our target is to design a data-intensive system (Kleppmann, 2017) where:

1. The target product is a Distributed Data Repository. Data is the central part, and it will drive the architecture.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

2. The objectives of the data repository are to have complex data structures, large amounts of data to process, and complex data flows manage complex data structures, large amounts of data to process, and complex data flows.
3. The value of the system comes from the value of data and knowledge exposed. Data is processed and finally, knowledge is available to the users.

4.1.2. USAGE

The 4+1 methodology is used to present the proposed architecture for the SHIFT platform.

4.2. Data Mesh

The term DATA MESH was coined by Zhamak Dehghani in 2019 (Dehghani, 2022) (Data-Mesh-General, 2023) and is based on four fundamental principles that bundle well-known concepts:

"The **domain ownership** principle mandates the domain teams to take responsibility for their data. According to this principle, analytical data should be composed around domains, similar to the team boundaries aligning with the system's bounded context. Following the domain-driven distributed architecture, analytical and operational data ownership is moved to the domain teams, away from the central data team.

The **data as a product** principle projects a product thinking philosophy onto analytical data. This principle means that there are consumers for the data beyond the domain. The domain team is responsible for satisfying the needs of other domains by providing high-quality data. Basically, domain data should be treated as any other public API.

The idea behind the **self-serve data infrastructure platform** is to adopt platform thinking to data infrastructure. A dedicated data platform team provides domain-agnostic functionality, tools, and systems to build, execute, and maintain interoperable data products for all domains. With its platform, the data platform team enables domain teams to seamlessly consume and create data products.

The **federated governance** principle achieves interoperability of all data products through standardization, which is promoted through the whole data mesh by the governance group. The main goal of federated governance is to create a data ecosystem with adherence to the organizational rules and industry regulations."

The MESH emerges when teams use other domains' data products. Using data from upstream domains simplifies data references and lookups (such as getting a new image), while data from downstream domains enables analyzing effects, e.g. for A/B tests (such as changes in the formulas). Data from multiple other domains can be aggregated to build comprehensive reports and new data products.

Data mesh, at its core, is founded on *decentralization* and *distribution of responsibility* to people who are closest to the data to support continuous change

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

and scalability. The components here are made of *data*, its *metadata*, and the *computation* necessary to serve it.

" (Data-Mesh-General, 2023)

4.2.1.MOTIVATION

SHIFT requirements fit very well with Data Mesh as defined above.

First, it is a data-intensive platform. That involves "data as product". Next, there are subdomains very well defined, where the domain ownership is precisely known. As presented in the objectives and ambition, the platform should deliver information and knowledge for a wide area of users, which are collaborating in creating, and consuming data products. This involves a self-serve data infrastructure platform.

And finally, federated governance is necessary to have an ecosystem created in a standardized form.

4.2.2.USAGE

The **domain ownership** will be implemented by defining workspaces for subdomains of data in cultural heritage institutions. A workspace is a group of data, processes, and knowledge targeted to a specific scope.

The data as a product principle is reflected in choosing data nodes as the main building block of the system and placing them in a P2P network.

The **self-serve data infrastructure platform** is proposed to be a P2P network, with data and knowledge nodes, each one exposing APIs for access.

The **federated governance** principle is reflected in the definition of standard APIs to access data from different data nodes, all equal in a P2P network.

In the same P2P network used for Data Mesh, there will be nodes containing the data and data accessing graphs.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

5. End-to-End Architecture definition

5.1. Overview

We are presenting in this chapter the end-to-end architecture of the platform.

It is a distributed system that is fully horizontally scalable and has specific front-end components. Data can be stored on several nodes or all nodes.

It will use a mechanism to retrieve data on a node, from any node of the system, fully transparent to the user.

It will permit data synchronization between nodes but without a consensus. That means that it is possible to have, for a short period (a few minutes), nodes with fewer data than others, and provide information at the time of interrogation, and the synchronization be done after the interrogation. Also, there are no parallel processes that try to update nodes and proceed only if they gain consensus to access a resource.

Even more, not all data is necessary to be synchronized.

There are elements fully synchronized, (index and hash), and based on the synchronized elements, it will be possible the data retrieval from any nodes.

There is common information presented in each node. This is summary information of all nodes. Also, each node is implementing the visualization and presentation tools developed in the project.

The core of the system is a backend distributed environment that offers the functionality for accessing and presenting artifacts.

Access to data, and visualization, is possible by using the client (Art wallet) which can be a haptic tool.

First, a general description and a global presentation of the 4+1 view are included. Next, each of the views is applied and the details of the architecture are presented.

The conceptual building blocks as they are described in DOA, where the starting point of our logical architecture. They are presented in the next figure (Figure 1. SHIFT conceptual building blocks)

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

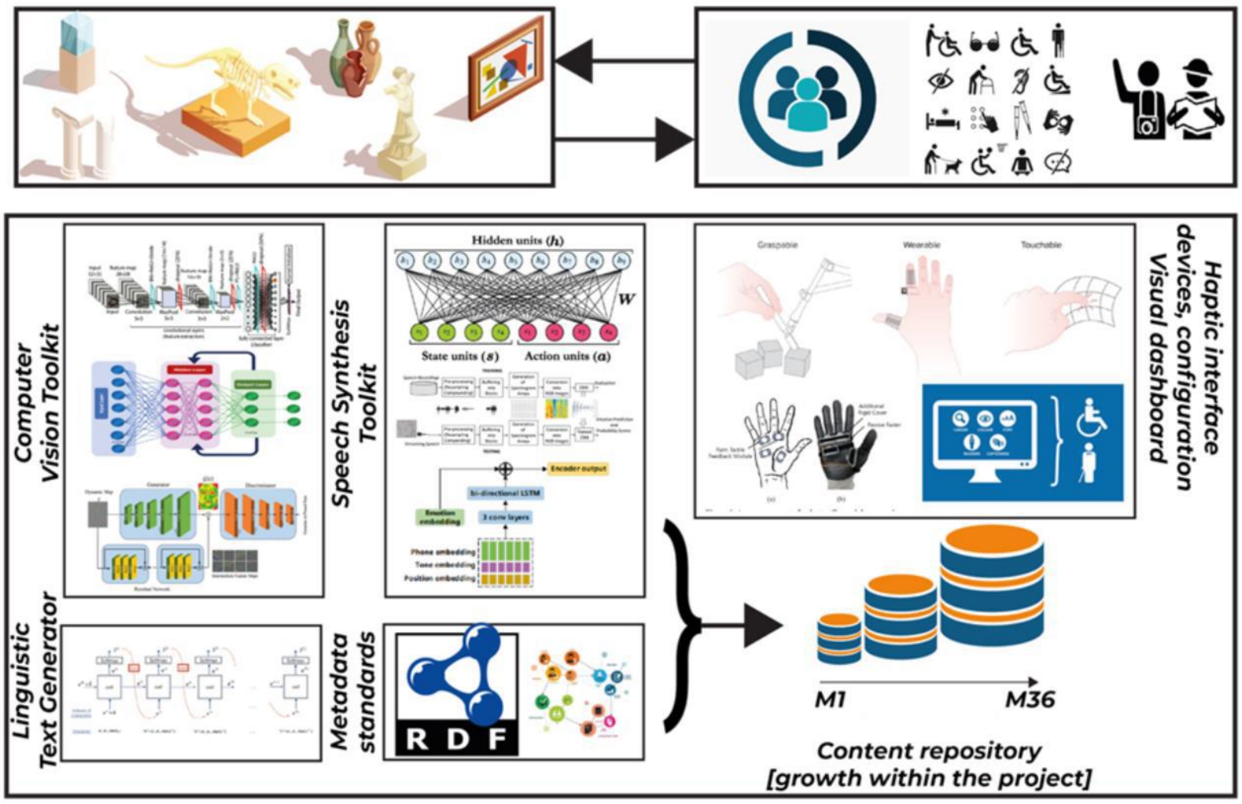


Figure 1. SHIFT conceptual building blocks

5.2. Synergies with other EU Horizon 2020 projects

The definition of the architecture for the present project was created in the context of developing R&D projects, under Horizon 2020, and having as one of the objectives the creation of a **repository** for some specific data.

The projects which were found to have more similarities were MES-CoBRaD[1] and MatCHMaker[2]

We presented in the next subchapters where the similarities of the projects generated synergies, and where each project followed a different branch.

[1] Multidisciplinary Expert System for the Assessment & Management of Complex Brain Disorders-GA 965422

2 HORIZON-CL4-2022-RESILIENCE-01-19: Advanced materials modelling and characterization-GA 101091687

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

5.2.1.SIMILARITIES IN USING THE METHODOLOGIES

The global methodology used to create the architecture is 4+1 for all the three mentioned projects. This methodology fits the needs of all projects, as described in Chapter 4.1.1: Motivation and expressed in terms of:

- Focus
- Simplicity
- Sequence
- Formality
- Usage

Considering that the same methodology is used, the 5 (4+1) views are present in each architecture, and this involved similar drawings of the global views and corresponding placement of specific objects in each architecture.

All the three mentioned projects have as an objective the treatment of a large and heterogenous amount of data, placed in many places. The specific treatment considered the usage of some specific concepts and architectural patterns.

1. In **MES-CoBRaD** (MES-CoBraD, 2021), a central Data Lake is used to store metadata and anonymized data, and distributed Edge components are used to store sensitive data. There are links between edge and central Data Lake, but no links between Edge components.

For this reason, a centralized data management mechanism was created, having at the center the Data Lake, and as dependent elements the Edge components.

2. In **MatCHMaker** (MatCHMaker, 2023) data is distributed in several places, and they are equal in what they allow to process. Data refers to models, metadata, algorithms, raw data. For this reason, the concept of Distributed Data and Knowledge Mesh (DDKM) is used. The DDKM concept is implemented in a distributed environment with equal nodes, each node containing models, implemented algorithms, metadata, or data. Each node can connect and exchange data with any other node. Nodes are almost all the time synchronized.
   Access to data is possible through a client, which can present data stored in any place of the distributed environment but indexed in the synchronized nodes. If necessary, calls to models or algorithms are possible.

3. In **SHIFT** data is distributed in several places. Each distributed node has its own processing system (which is different from MatCHMaker, where uniform processing is in place) Data refers to a different format of data describing artefacts. The data referring to artefacts is grouped in workspaces (specific to an exhibition). For this type of data, the concept of

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

Distributed Data Mesh (DDM) is used. The main difference in the concepts refers to the lack of K(Knowledge), as there are no models or algorithms placed in a node. The DDM concept is implemented in a distributed environment with equal nodes, each node containing data, metadata, or links to data. Each node can connect and exchange data with any other node. Nodes are almost all the time synchronized.
Access to data is possible through a client, which is able to present data stored in any place of the distributed environment but indexed in the synchronized nodes. In each node we can call specific tools to process data and create new data which is added to the workspace.

Based on the description of the data storage and processing requirements we found that:

A. The concepts used in MES-CoBRaD do not fit the requirements of SHIFT, so, they are not used.
B. A part of the concepts used in MatCHMaker are useful also for SHIFT, and the development in parallel enriches both projects.

The common elements identified and used refers to:

1. Usage of Data Mesh concept as the base of development. At the basis there is Data Mesh, but MatCHMaker will develop further the Knowledge Mesh, not present in SHIFT.
2. Both systems will develop and deploy a P2P (Peer to Peer) network. This involves the placement of equal nodes with links between all of them. This has a similar drawing representation, and similar functionalities, including:
    a. Seed node
    b. Discovery of nodes
    c. Synchronization of nodes
3. The functionalities of nodes are described in the architecture as Horizontal data exchange, are similar in the two projects and have similar drawing representation, similar sequence diagrams, and similar APIs use to command action in nodes.
4. The content of nodes has similar parts as different components. The similar parts refer to:
    a. Metadata
    b. Organization (Usage of Merkle Trees) to compact information
    c. Data used for the discovery and synchronization mechanism.

    The content which differs, and is specific only to Shift are:

    a. Storage of tools and modules
    b. The storage for workspaces
    c. Mechanism to use workspaces.

5. The way data in a node is placed and used is different.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

If in MatCHMaker the data is selected from an existing open repository, and links are created in the node to access the data, in SHIFT the process is more complex and involves:

- Management of workspaces (created to represent exhibitions).
- Add data (pictures, text, video, etc.) to workspaces.
- Call tools on a workspace to produce new data and place it in the workspace (for example call text to speech on a text, and place the resulted audio on the same workspace)
- Final user access to workspaces.

For this reason, the whole vertical data exchange is different in MatCHMaker and SHIFT and is developed on different branches.

6. The clients proposed are different. Both MatCHMaker and SHIFT are proposing WEB and text clients. But the content of what they are getting or displaying is completely different, since the vertical process is different. For example, MatCHMaker will be able to open a link to open repository where the user interacts, while in SHIFT the user can create workspaces, upload data, call modules, place the result of modules in workspaces.

5.2.2. SUMMARY OF SIMILARITIES AND DIFFERENCES

5.2.2.1. SIMILARITIES

- 4+1 methodology is used for all projects mentioned, and the views representation have similarities.
- Data Mesh concept is used in MatCHMaker and SHIFT
- P2P network is used in MatCHMaker and SHIFT
- Horizontal data exchange between nodes is similar in MatCHMaker and SHIFT (seed, discovery, synchronization)

5.2.2.2. DIFFERENCIES

- Knowledge Mesh concept is not used in SHIFT.
- SHIFT implements the concept of Workspaces.
- SHIFT implements functionalities to add data in Workspaces.
- SHIFT implements functionalities to call tools and create new data in workspaces.
- The whole vertical processing is completely different in SHIFT and MatCHMaker.
- Clients in SHIFT are specialized in displaying artefacts and are completely different from those implemented in MatCHMaker.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

5.3. Requirements

The definitions of the user requirements are extensively presented in D1.1 (SHIFT-D1.1, 2023). We are summarizing here the functional requirements extracted from the user requirements, and we are introducing the non-functional (technical requirements).

MoSCoW prioritization is specified for each requirement (functional or technical).

The requirements considered are an important driver for the definition of the system architecture.

5.3.1.FUNCTIONAL REQUIREMENTS

The functional requirements are based on the user requirements defined in D1.1 (SHIFT-D1.1, 2023).

The functional requirements are addressed to different end users.

The end users considered are grouped into 4 categories:

EU1-CH Professional

EU2-VI Visually impaired user

EU3-BL Blind user

EU4-HI Hearing impaired user

For each functional requirement, the MoSCoW prioritization is also included.

In summary, the functional requirements are mentioned in the next table:

Table 1 SHIFT Functional requirements

Code	Description	End Users	MoSCoW
FR1-VC	Visual contrast: For good visual perception, adjacent surfaces should differ not only in color but also in shade. For people with partial or total color blindness, this light/dark contrast is extremely important.	EU2-VI	M
FR2-VC	Visual contrast: Matching or similar colors, such as light blue and dark blue or light green and dark green, should therefore be avoided.	EU2-VI	S
FR3-VC	Visual contrast: The color combination red/green is completely unsuitable	EU2-VI	S

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

	(approx. 9% of the population suffer from red-green color blindness).		
FR4-PA	Picture to animation transformation	EU1-CH	M
FR5-TS	Text to Speech transformation.	EU1-CH, EU2-VI, EU3-BL	M
FR6-EC	Image and video processing by enhancing the contrast of the visual content.	EU1-CH, EU2-VI	M
FR7-UF	User-friendly access to various approaches and perspectives regarding CH artifacts, with the possibility of sorting, filtering, labeling, and classification is the most important benefit of a technology-assisted system for curating efficiency.	EU1-CH, EU2-VI, EU4-HI	M
FR8-FS	Facilitating access to various digitized CH resources (publications, studies, collections, catalogs, exhibitions, virtual tours, audio-video materials, etc.)	EU1-CH, EU2-VI, EU3-BL, EU4-HI	S
FR9-MC	Multimedia content: images, videos, podcasts, and other multimedia formats, which the IT system can sort, metadata, and classify according to the topics relevant to the end users' preferences.	EU1-CH, EU2-VI, EU3-BL, EU4-HI	M
FR10-VG	Virtual Guides: An artificial intelligence assistant could be programmed to provide information about exhibits and events, as well as answer user questions.	EU1-CH, EU2-VI, EU3-BL, EU4-HI	S
FR11-DR	Digital representation of objects to watch on visitor's devices (like tablets): to magnify images, highlight details, strengthen contrasts, to delete details; this could help partially sighted persons or persons with motoric problems.	EU1-CH, EU2-VI, EU4-HI	S
FR12-AG	An automatic guide system that composes special tours in the collection of objects related to children, gender	EU1-CH, EU2-VI,	C

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

	equality, ethnic aspects, disabilities, etc.	EU3-BL, EU4-HI	
FR13-AC	Accessibility: the possibility of access to cultural heritage information in a variety of formats, including text, images, and media, through an intuitive and personalized interface. AI-assisted computer systems could provide accessibility through advanced search tools and recommendation algorithms.	EU1-CH, EU2-VI, EU3-BL, EU4-HI	S
FR14-IA	Interface accessibility: the system can be used easily by all users as a priority, including disabled people	EU1-CH, EU2-VI, EU3-BL, EU4-HI	M
FR15-VI	Using a clear and intuitive navigation menu adapted to the visually impaired.	EU1-CH, EU2-VI	M
FR16-FO	Using an appropriate font to facilitate reading for people with low vision.	EU1-CH, EU2-VI	M
FR17-SC	Suitability of the CH content to the cultural diversity of the users	EU1-CH	S
FR18-DS	Information and components of the IT system should be delivered to users in ways that they can receive and understand correctly regardless of any disabilities or physical limitations they may encounter.	EU1-CH, EU2-VI, EU3-BL, EU4-HI	M
FR19-ST	Stories that present oral or traditional histories collected from local communities.	EU1-CH, EU4-HI	M
FR20-VT	Stories like virtual tours that provide information about the cultural heritage of an area, heritage buildings, museums, libraries, archives, etc.	EU1-CH, EU2-VI, EU3-BL, EU4-HI	M
FR21-AR	Stories representing descriptions of tangible (photographs, works of art, monuments, etc.) and intangible (landscapes, attributes /approaches/songs, etc.) CH items.	EU1-CH, EU2-VI, EU3-BL	S

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

FR22-EM	Stories that increase the emotional impact of CH digital content by integrating musical compositions or suggestive images.	EU1-CH	
FR23-HA	Translation of physical objects to digital objects and uses haptics to "feel" the objects. Implement haptic interaction with 3D digital tangible and intangible cultural heritage assets, augmenting the user experience (UX) with new interaction paradigms that can be used in situ or remotely	EU1-CH, EU2-VI, EU3-BL	M
FR24-AS	It should be possible to ask for assistance	EU2-VI, EU3-BL, EU4-HI	M
FR25-MM	Multimodality of engagement or alternative formats	EU2-VI, EU3-BL, EU4-HI	M

5.3.2. NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements refer to the general technical requirements proposed for all modules, to conduct a coherent system deployment, able to respond to the user needs presented in D1.1 (SHIFT-D1.1, 2023). Non-functional requirements are included in the next table:

Table 2 SHIFT Non-functional requirements

ID	**Category / Requirement**		MoSCoW
NFR	System Requirements		
NFR-001	Support high-availability requirements	SHIFT must support high-availability requirements, operating without outages within working hours.	M
NFR-002	Support backup and recovery operations	SHIFT must support copying and archiving data for restoring the original after a data loss event. Backups should be programmed outside working hours.	M
NFR-003	Support both horizontal and vertical scalability scenarios	SHIFT must support both horizontal and vertical scalability for capacity expansion scenarios. This	M

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

		should be realized using a distributed approach.	
NFR-004	Support agreed on workload and response time performance requirements	SHIFT must support agreed workload and response time performance requirements guaranteeing operation within considered time limits.	M
NFR-005	Allow working on different hardware platforms (x64 compatible)	SHIFT must be designed to allow running server-side services on Linux and Windows hardware platforms (x64 compatible).	M
NFR-006	Support virtualization options for hardware resources	SHIFT must support virtualization options for hardware resources to allow optimal use of existing capacity, within budget constraints.	M
NFR-007	Support cloud deployments	SHIFT must support deployments in the cloud.	
GUI	Graphical User Interface Requirements		
NFR-008	Provide a common look-and-feel to the graphical user interface	SHIFT must provide a common look-and-feel through a portal-like user interface to guide the user to the underlying functionality of the internal components. The look-and feel should be similar for different devices (Desktop, mobile, etc.).	M
NFR-009	Provide intuitive general navigation methods	SHIFT must provide intuitive general navigation methods. At least two of the following navigation methods should be provided: main menu, site map, and search engine. Breadcrumbs must be used to indicate the current feature and to provide easy hierarchical navigation. Ona scrolled page should be used for large content.	M
NFR-010	Use unambiguous text to describe features	Text used for the SHIFT's user interface should be unambiguous. Typefaces and fonts used should be easily readable and should support international accents. Where symbols are used consider associating descriptive text as well. Beware of cultural	M

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

		differences related to naming and symbols.	
NFR-011	Support localized texts for international end-users	SHIFT must support localized texts for international end-users, by using Unicode compliance for global text display, independent from a specific language/character set encoding, while also supporting right-to-left languages.	M
NFR-012	Use appropriate colors and contrast	SHIFT must use appropriate colors and contrast, avoiding conveying meaning by color, as there may be cultural differences in interpreting colors between users of the system while making sure that information is comprehensible, even if the colors are absent.	S
NFR-013	Provide consistent labels for buttons and fields	SHIFT must provide consistent labels for buttons and fields. An explicit label must be provided for each form field. Each label must be placed close to the field to which it is attached. Group together related fields. Indicate mandatory fields and provide help for entering data.	S
NFR-014	Indicate the size and format of documents available for download	SHIFT must indicate the size and format of each document that can be downloaded. For each link that points to a document that can be downloaded, the link text should include the document name, file format, and size.	S
NFR-015	Provide account authentication interface for user identification	SHIFT must provide a graphical interface for authenticating end-users. Reset password or request user account should be provided.	M
NFR-016	Provide adaptive user interface based on authorization access rights	SHIFT must provide an adaptive user interface based on the authorization access rights of the user role, such as displaying only links/menus to activities and resources to which the user has access.	M

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

NFR-017	Provide accessibility options (as mentioned in functional requirements)	SHIFT must provide user interfaces adapted to different users considered the four groups: **EU1-CH** professional **EU2-VI** Visually impaired user **EU3-BL** Blind user **EU4-HI** Hearing impaired	
INT	Interoperability Requirements		
NFR-018	Integrate components and external systems in a loosely coupled way	SHIFT must integrate internal components and external systems in a loosely coupled way; such that each of its components has, or makes use of, little or no knowledge of the definitions of other separate components.	M
NFR-019	System components expose and consume data in a standardized way	SHIFT components must expose and consume data in a standardized way. (MIME format, JSON packaged)	M
NFR-020	Publish and describe exposed interfaces towards other systems	SHIFT must publish and describe exposed interfaces towards other systems by managing a set of open APIs towards other systems, used both to develop the actual SHIFT system and to extend the system in the future if needed.	M
NFR-021	Describe the syntax and format used for data exchange messages	SHIFT must describe the syntax and format used for data exchange messages, while also specifying the semantics of data fields. Standardization of messages ensures that messages are robust, interoperable, and reusable.	M
NFR-022	Leverage open standards to communicate with external systems	SHIFT must leverage open standards, where available, to communicate with external systems, as opposed to implementing custom means of data exchange.	S

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

5.3.2.1. KEY METRICS

For all the non-functioning requirements we Are proposing a set of key metrics to express the “definition of done” for each one.

This is presented in the next table:

Table 3 SHIFT Non-functional requirements key metrics

ID	**Category / Requirement**		Value
NFR	System Requirements	Key metrics	
NFR-001	Support high-availability requirements	Availability during working hours in a week measured as AVW=Time of availability/Total time working hours	AVW>99.5%
NFR-002	Support backup and recovery operations	Percent of data coverage by backups (COV) Tine for full recovery (TFR) of 100 Gb data	COV=100% TFR < 20 Min
NFR-003	Support both horizontal and vertical scalability scenarios	Vertical scalability (VSCA). = Percent of CPU and memory that can be added to improve performance Above this percent there will be no improvements. Tests using Jmeter. Horizontal scalability (HSCA) = Maximum number of nodes for which the system is manageable using the same distribution mechanism. Tests using Jmeter.	VSCA>100% HSCA>100.
NFR-004	Support agreed on workload and response time performance requirements	The volume of data on a node (VDN) is measured as available space on node storage. Answer time for simple page display measured in browser developer tools (Networking) Waiting for the Server to response (WSR) Page Content Download (PCD)	VDN>100G WSR<0.3s PCD<1s

Funded by the European Union

This project has received funding from the European Union’s Horizon Europe research and innovation programme under grant agreement no 101060660.

NFR-005	Allow working on different hardware platforms (x64 compatible)	Support Linux and Windows platforms measured as installation and working with success	>=1 Linux nodes >=1 Windows nodes
NFR-006	Support virtualization options for hardware resources	Support VMWare or Oracle VM VirtualBox. Tested by creating virtual machines on the two virtualization environments	>=1 VMWare nodes >=1 VirtualBox nodes
NFR-007	Support cloud deployments	Support cloud AWS, AZURE. Testing by deploying nodes in the two cloud environments	>=1 AWS nodes >=1 AZURE nodes
GUI	Graphical User Interface Requirements		
NFR-008	Provide a common look-and-feel to the graphical user interface	Usage of one CSS for all modules in web applications. Use the same theme for all the other applications (Compliant with ISO 9241)	1 CSS 1 Theme
NFR-009	Provide intuitive general navigation methods	Number of clicks to reach any business function NCLI. (Compliant with ISO 9241)	NCLI <=4
NFR-010	Use unambiguous text to describe features	Test how users understand the features without training (UXFNT) and after training (UXFAT) Measured the percentage of failures. (Compliant with ISO 9241)	UXFNT <10% UXFAT<2%
NFR-011	Support localized texts for international end-users	Languages supported: English Hungarian Romanian Serbian German	5 languages supported
NFR-012	Use appropriate colors and contrast	(Compliant with ISO 9241) Test how users understand the colors (including users with disabilities. Measure the failure color identification rate (FCR)	FCR<=10%

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

NFR-013	Provide consistent labels for buttons and fields	Test how users understand the features without training (UXFNT) and after training (UXFAT) Measured the percentage of failures. (Compliant with ISO 9241)	UXFNT <10% UXFAT<2%
NFR-014	Indicate the size and format of documents available for download	Presence of document size to be downloaded. (PDS)	PDS=100%
NFR-015	Provide account authentication interface for user identification	Possibility to access without authentication (FREEACC)	FREEACC=0
NFR-016	Provide adaptive user interface based on authorization access rights	Number of roles and different UIs. **EU1-CH** professional **EU2-VI** Visually impaired user **EU3-BL** Blind user **EU4-HI** Hearing impaired **ADMIN**	NROL=5
NFR-017	Provide accessibility options (as mentioned in functional requirements)	Accessibility for: **EU2-VI** Visually impaired user **EU3-BL** Blind user **EU4-HI** Hearing impaired	3 accessibility roles
INT	Interoperability		
NFR-018	Integrate components and external systems in a loosely coupled way	Loosely coupled indicator (LCI=Number of modules uncoupled/no total modules) * 100	LCI>70%
NFR-019	System components expose and consume data in a standardized way	Data Formats Used to exchange data (DFU) JSON, XML, CSV, Binary	DFU<=4
NFR-020	Publish and describe exposed interfaces towards other systems	Percentage of documented exposed interfaces over all interfaces (DEI)	DEI=100%
NFR-021	Describe the syntax and format used for data exchange messages	Percentage of documented data exchange messages from all messages (DEM)	DEM=100%

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

NFR-022	Leverage open standards to communicate with external systems	Percentage of open standards over all standards (POS)	POS=100%

5.4. Global architecture views

Considering the methodologies selected, the global view of the whole system is that of the main concept: Distributed Data and Knowledge Mesh, viewed (illuminated) by the 4+1 views: Logical, process, physical, and Development, plus the central one: use case.

We will present separate views of the main concept in the next subchapters.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

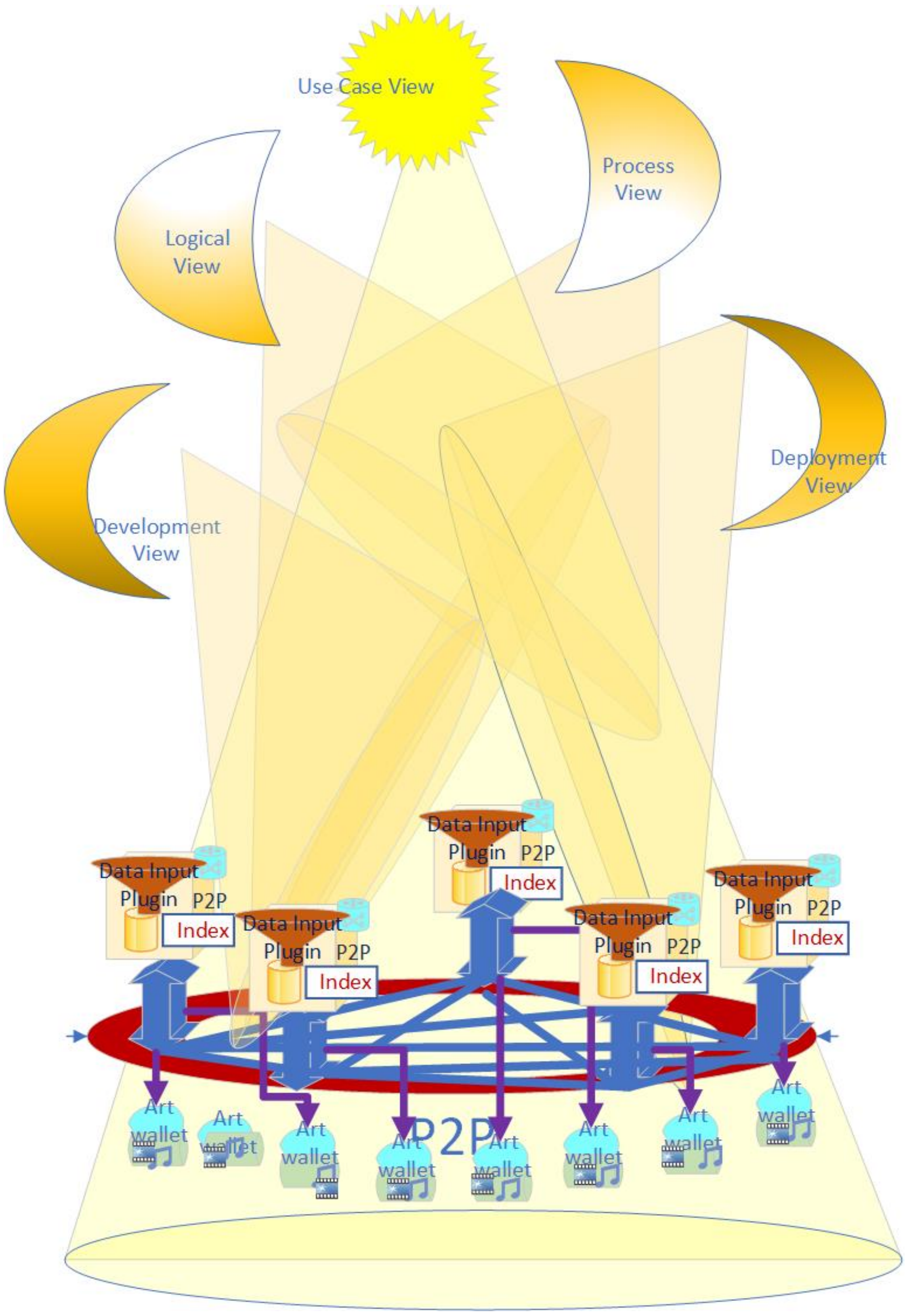


Figure 2. SHIFT architectural views

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

5.5. USE-CASE View

We are presenting in this chapter the user point of view on the system, expressed in the description of use cases and detailed in user requirements D1.1 (SHIFT-D1.1, 2023).

Four use cases are considered and are described in the following subchapters.

5.5.1. UC1- 19TH TO MODERN DAYS SERBIAN PAINTINGS AND MODERN ART

According to the DOA (SHIFT Consortium, 2022), this use case is described as:

"The exhibition aims to augment the experience of visitors with an exhibition focused on the most significant 19th-century and contemporary Serbian paintings using innovative tools. The beauty of still images (paintings or photos) is not easily appreciated by regular visitors and short video clips will improve the social richness of the cultural assets. On top of this, for people with visual impairments, we'll provide 'audio captioning' for the short videos.

Also, museum curators will have better support in organizing the exhibition layout/objects in a culturally significant order, with contemporary references.

The situation is similar in other member museums of The Balkan Museum Network, and from other countries, as well.

SHIFT will become an essential service and mechanism for The Balkan Museum Network Groups to increase access by "modernizing" and being more up to date with a new and intense solution that creates excitement among visitors."

Based on D1.1 (SHIFT-D1.1, 2023), the use case will prepare an exhibition "based on the selection of paintings, drawings, graphics, icons, posters, and photos from the artistic collection of the museum. The working title of the exhibition will be "Pictures speak". Using digital content and tools will provide the possibility to bring two-dimensional objects, such as paintings, to life. Each object will have an audio description explaining the art works, or recorded narration about the person or event/place presented, customs, and objects combined with digital tools and effects such as 3D animation, and AR/VR elements, providing multimodal and multisensory access to the collection, etc. The added value of the selected approach is that it provides access and inclusion to the collection, not only for one target group, but for all, and improves the quality of museum programs, services, and work, in general."

This will be achieved by revitalizing existing CH, through testing the following tools (as mentioned in D1.1 (SHIFT-D1.1, 2023)):

"

- Tool to enhance Photos / Paintings to Short Videos
- Audio tool – "Video to Speech" capable of interpreting visual stimuli (e.g., actions explained in visual sequences)

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

- Tool to "Text to Speech" that automatically can provide complementary information regarding the cultural heritage assets (books, paintings, photos)
- Tool that translates historical meaning into more contemporary language and for auto-tagging/ auto-categorization of cultural heritage resources.

- Events/presentations/workshops that will serve as pilot testing of proposed and possible outcomes.
-

"

5.5.2. UC2- EXPERIMENTING WITH THE TRANSFORMATION OF MEDICINE AND PHARMACY

According to the DOA (SHIFT Consortium, 2022), this use case is described as:

"This exhibition aims to emerge the visitors into the history of medicine and let them "feel" how different illnesses have been treated before modern times. This will be achieved using several tools within the project, haptics being among them.

To achieve this objective, this pilot exhibition will test the following tools:

- Tool to enhance Photos / Paintings to Short Videos
- Audio tool – "Video to Speech" capable of interpreting visual stimuli (e.g., actions explained in visual sequences)
- Tool that translates physical objects to digital objects and uses haptics to "feel" the objects. To implement haptic interaction with 3D digital tangible and intangible cultural heritage assets, augmenting the user experience (UX) with new interaction paradigms that can be used in situ or remotely
- Tool that translates historical meaning into more contemporary language and for auto-tagging/ auto-categorization of cultural heritage resources

"

5.5.3. UC3- ROMANIAN HISTORY AND CUSTOMS EXPLAINED TO DIGITAL NATIVES

According to the DOA (SHIFT Consortium, 2022), this use case is described as:

"This pilot aims to support and engage at least 10 member libraries to revitalize their book collections presentations and descriptions, to boost the interest also for the digital native generation of European citizens. Member libraries will also be engaged in a contest to raise interest and will encourage citizens, through social media, to share their collection of historical photos to create short motion videos.

To achieve this objective, within this pilot exhibition will be tested the following tools:

- Tool to enhance Photos / Paintings to Short Videos

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

- Tool to "Text to Speech" that automatically can provide complementary information regarding the cultural heritage assets (books, paintings, photos)
- Tool that translates historical meaning into a more contemporary language (e.g. better understanding old languages like Shakespeare) and for auto-tagging/ auto-categorization of cultural heritage resources
- Comprehensive intuitive and accessible tool for all (including individuals with disabilities) multimodal storytelling of cultural heritage assets. "

5.5.4. UC4- CH EXHIBITION AS VISITOR'S JOURNEY'S, WITH NO SENSING BOUNDARIES

According to the DOA (SHIFT Consortium, 2022), this use case is described as:

"The SMB Museum is well aligned to the modern digital tools but still lacks sufficient solutions for people with visual impairments, and a stream of newly transformed content. Therefore, we aim to test innovative solutions that will provide superior assistance for people with visual impairments while visiting the SMB museum.

A novel exhibition on "CH exhibition as visitor's journey, with no sensing boundaries" will be prepared and will encompass the following:

- Tool to "Text to Speech" tool for people with visual impairments – will use book resources, descriptions of photos/ paintings from curators.
- Tool to "Text to Speech" that automatically can provide complementary information regarding the cultural heritage assets (books, paintings, photos).
- Tool that translates physical objects to digital objects and uses haptics to "feel" the objects. To implement haptic interaction with 3D digital tangible and intangible cultural heritage assets, augmenting the user experience (UX) with new interaction paradigms that can be used in situ or remotely.
- Comprehensive intuitive and accessible tool for all multimodal storytelling of cultural heritage assets. "

5.5.5. SUMMARY OF TOOLS PROPOSED IN ALL USE CASES.

D1.1 (SHIFT-D1.1, 2023) defines the usage of tools in each use case. The technologies comprising the SHIFT tools aim at expanding the inclusivity of CH organizations, embracing in more efficient ways the individuals at risk of exclusion, such as people with sensory disabilities like visual impairments, and enticing new audiences such as the younger generation.

Considering the proposed tools for each use case, and summarizing them, it results in 7 main tools to be developed as data processing tools and be placed in each node of the Platform.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

Table 4 SHIFT Tools by Use Cases

Code	Description	Use cases
T1-IV Image to video	Tool to enhance Photos / Paintings to Short Videos	UC1, UC2, UC3
T2-VS Video to Speech	Audio tool capable of interpreting visual stimuli that is first converted to text and then into speech that is finally embedded to the video	UC1, UC2, UC4
T3-HA Haptic Interaction	A tool that translates physical objects to digital objects and uses haptics to “feel” the objects. To implement haptic interaction with 3D digital tangible and intangible cultural heritage assets, augmenting the user experience (UX) with new interaction paradigms that can be used in situ or remotely	UC2, UC4
T4-AN Audio Narrative	The tool that automatically can provide complementary information regarding the cultural heritage assets (books, paintings, photos)	UC1, UC2, UC3
T5-CT Contemporary Translation	A tool that translates historical meaning into more contemporary language and for auto-tagging/ auto-categorization of cultural heritage resources.	UC1, UC2, UC3
T6-AF Accessibility Framework	Comprehensive intuitive and accessible tool for all (including individuals with disabilities) multimodal storytelling of cultural heritage assets.	UC1, UC3, UC4
T7-AT Accessible Text-to-Speech	Tool to “Text to Speech” that automatically can provide complementary information regarding the cultural heritage assets, by generating and transforming image to text to speech	UC3, UC4

The usage of the tools is summarized in D1.1 (SHIFT-D1.1, 2023) and presented also in the next table:

Funded by the European Union

This project has received funding from the European Union’s Horizon Europe research and innovation programme under grant agreement no 101060660.

Table 5 SHIFT Tools usage

Code	Use cases	Devices	User roles
T1-IV Image to video	UC1, UC2, UC3	Device-independent (desktop, laptop, kiosk, tablet, mobile)	heritage users heritage professionals
T2-VS Video to Speech	UC1, UC2, UC4	Device-independent	heritage users
T3-HA Haptic Interaction	UC2, UC4	Haptic devices	heritage users
T4-AN Audio Narrative	UC1, UC2, UC3	Device-independent	heritage users
T5-CT Contemporary Translation	UC1, UC2, UC3	Device-independent	heritage professionals
T6-AF Accessibility Framework	UC1, UC3, UC4	Extended Reality (XR) devices	heritage users
T7-AT Accessible Text-to-Speech	UC3, UC4	Device-independent	heritage users

5.5.6. DATA INPUTS AND OUTPUTS

The tools presented in previous chapters have the following input and output format:

Table 6 SHIFT Tools inputs/outputs

Code	Use cases	Input	Output
T1-IV Image to video	UC1, UC2, UC3	Images (JPG, PDF, PNG)	Videos (mp4, avi, srt)
T2-VS Video to Speech	UC1, UC2, UC4	Videos (mp4, avi, srt)	Audio (wav, avi)
T3-HA Haptic Interaction	UC2, UC4	Images (JPG, PDF, PNG)	Haptic output

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

T4-AN Audio Narrative	UC1, UC2, UC3	Images (JPG, PDF, PNG) Text (TXT, DOC, PDF)	Audio (wav, avi)
T5-CT Contemporary Translation	UC1, UC2, UC3	Text (TXT, DOC, PDF)	Text (TXT, DOC, PDF)
T6-AF Accessibility Framework	UC1, UC3, UC4	Images (JPG, PDF, PNG) Text (TXT, DOC, PDF)	Extended Reality (XR) output
T7-AT Accessible Text-to-Speech	UC3, UC4	Text (TXT, DOC, PDF)	Audio (wav, avi)

5.5.7. MAPPING OF FUNCTIONAL REQUIREMENTS ON THE TOOLS

The functional requirements defined in chapter 5.3.1 are mapped on the tools T1-T7 as specified in the next table:

Table 7 SHIFT Functional requirements - Tools mapping

Functional requirement	Description	Tools
FR1-VC	Visual contrast: For good visual perception, adjacent surfaces should differ not only in color but also in shade. For people with partial or total color blindness, this light/dark contrast is extremely important.	T1-IV
FR2-VC	Visual contrast: Matching or similar colors, such as light blue and dark blue or light green and dark green, should therefore be avoided.	T1-IV
FR3-VC	Visual contrast: The color combination red/green is completely unsuitable (approx. 9% of the population suffer from red-green color blindness).	T1-IV
FR4-PA	Picture to animation transformation	T1-IV
FR5-TS	Text to Speech transformation.	T2-VS, T7-AT

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

FR6-EC	Image and video processing by enhancing the contrast of the visual content.	T2-VS, T7-AT
FR7-UF	User-friendly access to various approaches and perspectives regarding CH artifacts, with the possibility of sorting, filtering, labeling, and classification is the most important benefit of a technology-assisted system for curating efficiency.	T1-IV, T2-VS, T4-AN, T5-CT, T7-AT
FR8-FS	Facilitating access to various digitized CH resources (publications, studies, collections, catalogs, exhibitions, virtual tours, audio-video materials, etc.)	T1-IV, T2-VS, T4-AN, T5-CT, T7-AT
FR9-MC	Multimedia content: images, videos, podcasts, and other multimedia formats, which the IT system can sort, metadata, and classify according to the topics relevant to the end users' preferences.	T1-IV, T2-VS, T4-AN, T5-CT, T7-AT
FR10-VG	Virtual Guides: An artificial intelligence assistant could be programmed to provide information about exhibits and events, as well as answer user questions.	All tools
FR11-DR	Digital representation of objects to watch on visitor's devices (like tablets): to magnify images, highlight details, strengthen contrasts, to delete details; this could help partially sighted persons or persons with motoric problems.	T1-IV, T6-AF
FR12-AG	An automatic guide system that composes special tours in the collection of objects related to children, gender equality, ethnic aspects, disabilities, etc.	T1-IV, T2-VS, T4-AN, T5-CT, T7-AT
FR13-AC	Accessibility: the possibility of access to cultural heritage information in a variety of formats, including text, images, and media, through an intuitive and personalized interface. AI-assisted computer systems could provide accessibility through advanced search tools and recommendation algorithms.	T1-IV, T2-VS, T4-AN, T5-CT, T6-AF, T7-AT

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

FR14-IA	Interface accessibility: the system can be used easily by all users as a priority, including disabled people	All tools
FR15-VI	Using a clear and intuitive navigation menu adapted to the visually impaired.	T1-IV, T2-VS, T6-AF, T7-AT
FR16-FO	Using an appropriate font to facilitate reading for people with low vision.	T1-IV, T2-VS, T6-AF, T7-AT
FR17-SC	Suitability of the CH content to the cultural diversity of the users	T1-IV, T2-VS, T6-AF, T7-AT
FR18-DS	Information and components of the IT system should be delivered to users in ways that they can receive and understand correctly regardless of any disabilities or physical limitations they may encounter.	All tools
FR19-ST	Stories that present oral or traditional histories collected from local communities.	T2-VS, T4-AN, T7-AT
FR20-VT	Stories like virtual tours that provide information about the cultural heritage of an area, heritage buildings, museums, libraries, archives, etc.	T2-VS, T4-AN, T7-AT
FR21-AR	Stories representing descriptions of tangible (photographs, works of art, monuments, etc.) and intangible (landscapes, attributes /approaches/songs, etc.) CH items.	T2-VS, T4-AN, T7-AT
FR22-EM	Stories that increase the emotional impact of CH digital content by integrating musical compositions or suggestive images.	T2-VS, T4-AN, T7-AT
FR23-HA	Usage of haptic tools	T3-HA
FR24-AS	It should be possible to ask for assistance	All tools
FR25-MM	Multimodality of engagement or alternative formats	All tools

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

5.6. Logical (Conceptual view)

We are starting the description of the logical view by presenting our understanding of the current situation:

IT systems are placed in different locations and deliver services to users who are connected to them.

A user should connect to the right place, to obtain the right result.

There are no links between IT systems.

One user can be connected to one or more systems.

The actual environment is presented in the next figure (Figure 3. Actual IT environments).

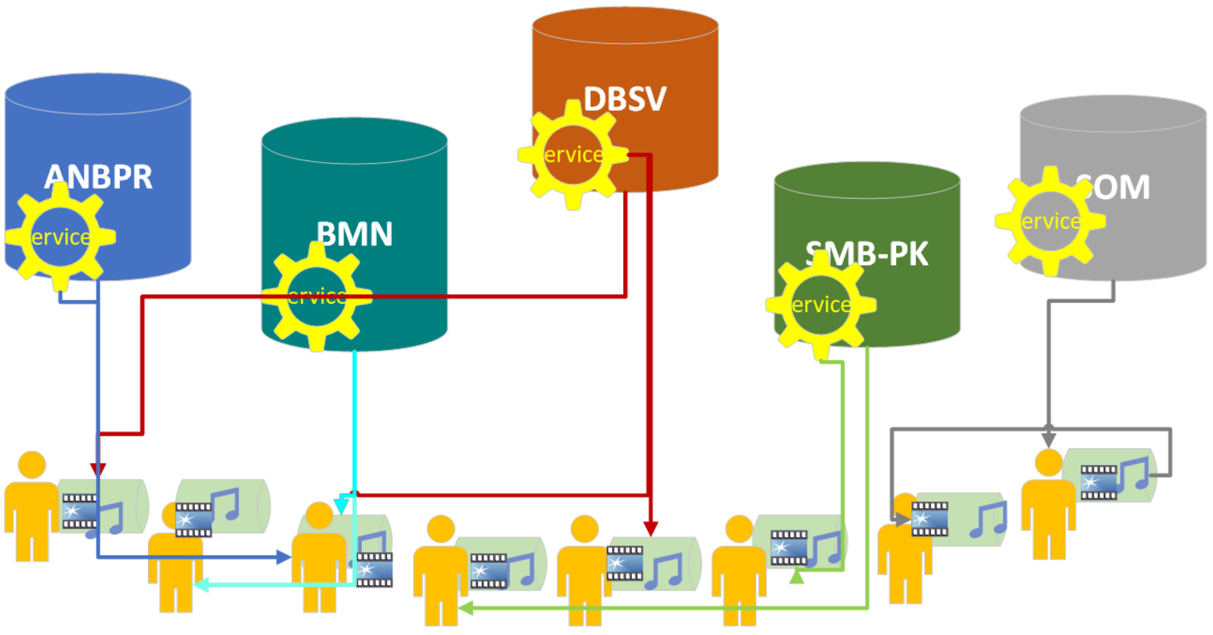


Figure 3. Actual IT environments

Starting with the actual context, and considering the ambition and the objectives of the system, we are proposing an environment, as in the next figure (Figure 4. Proposed Logical Architecture)

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

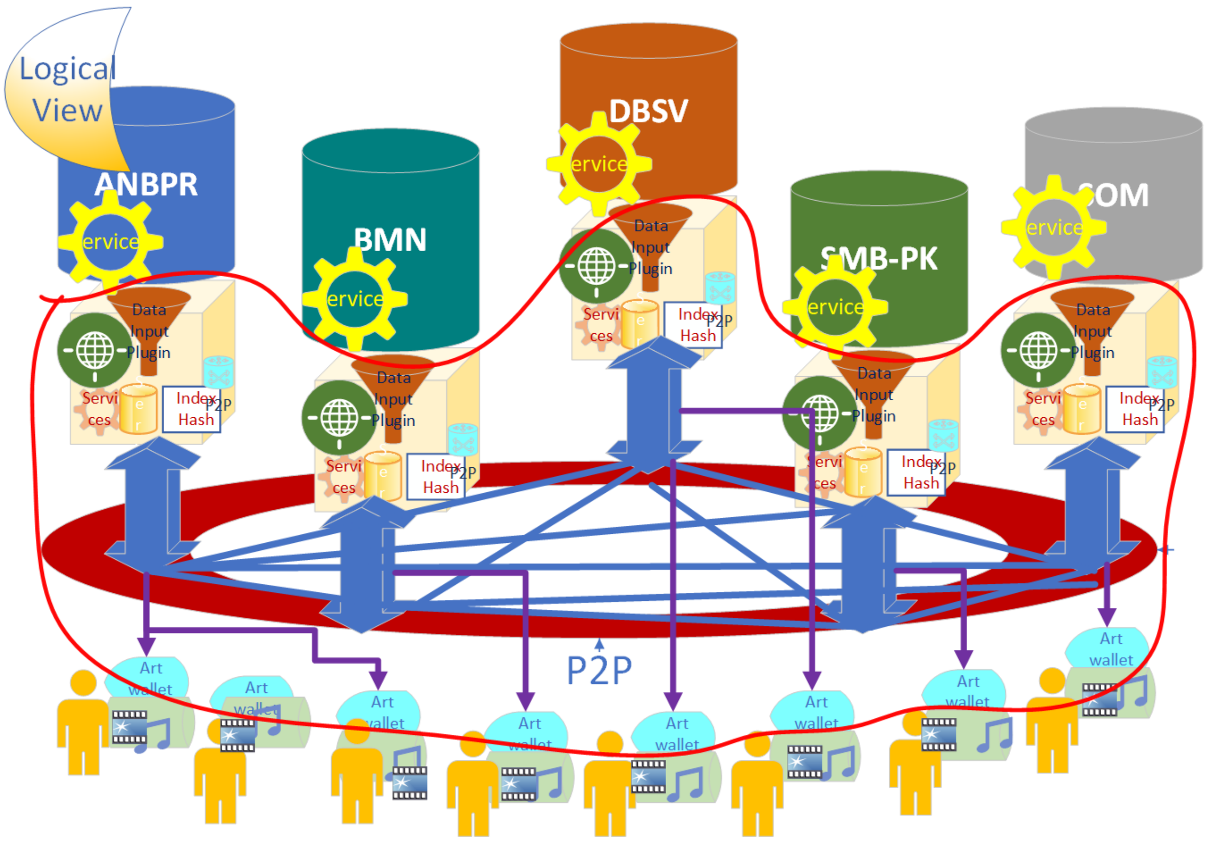


Figure 4. Proposed Logical Architecture

In the proposed architecture, we can see:

1. The actual IT systems remain in their place to serve the new components.
2. In each place of interest is placed a **backend node** (represented by a yellow cube in the figure), created as part of this project, which is responsible for:
    a. Access the data from the initial system. This is possible using specific plugins, specially developed for each site. The output is normalized and is in a similar format for all sites.
    b. Process the input data, using the algorithms described in WP2, WP3, and WP4 and associated with T1-T7 Tools (Table 4 SHIFT Tools by Use Cases)
    c. Index the data accessible and store it as a persistence mechanism.
    d. Synchronize indexed data from multiple sites.
    e. Expose the indexed data.
    f. Manage the list of all existing nodes.
    g. Employ a mechanism to implement a P2P protocol used to discover other nodes, update the list of nodes, and synchronize nodes (without consensus).
    h. Expose functionality via API.
    i. Expose indexed data.

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

   j. Exposed local (site data), when requested.
   k. Exposed local processed data (Exposed site data after being processed by one of the and associated with T1-T7 Tools (Table 4 SHIFT Tools by Use Cases).
3. **A P2P network** (represented in the figure by the sum of blue arrows), where all the nodes are connected, and where synchronization of indexed data is possible as also uniform access to all data, by just accessing one node.
4. A client, located on a user device (computer, mobile), acts as a wallet (named **Art Wallet** in the figure). The client will be able to present the processed data, received via a node.

- The logical architecture of a **backend node** is presented in the next figure (Figure 5. Node components)

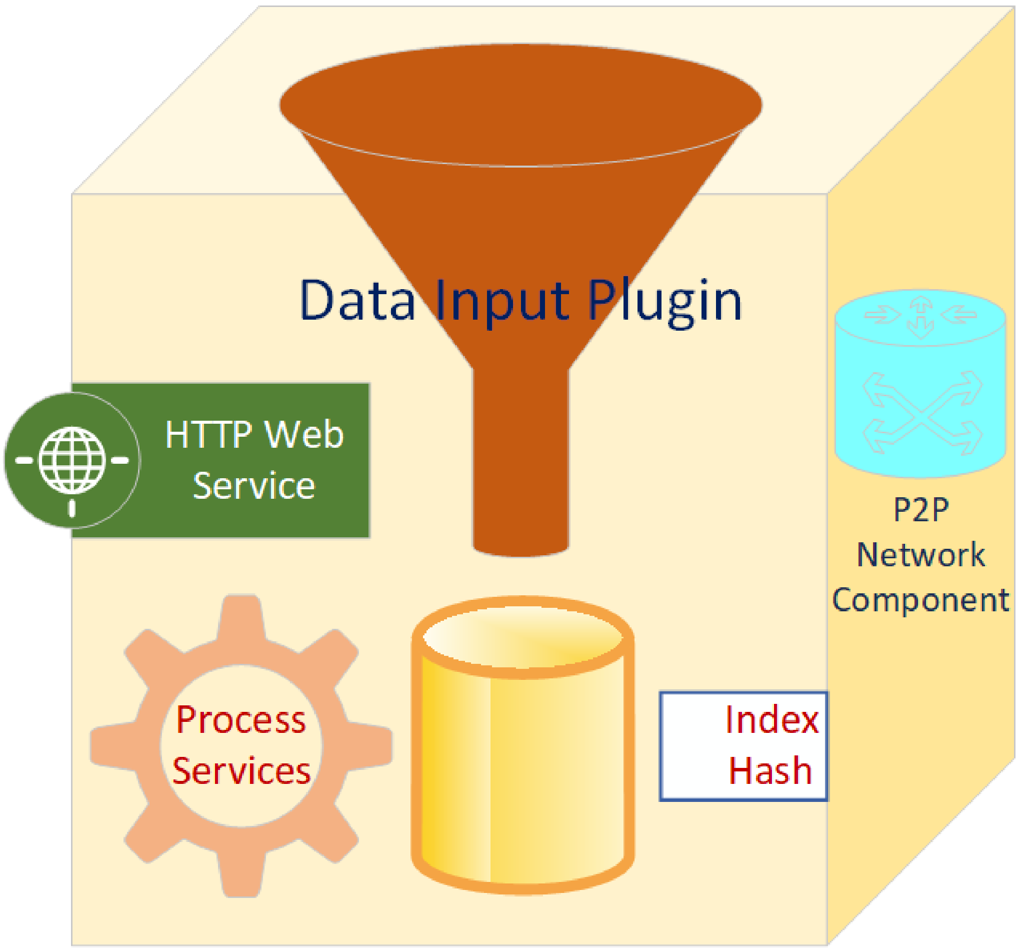


Figure 5. Node components

In this figure, we have:

1. Data input plugin. It is responsible for accessing data on a site. Is specific to each site. If a new site is added to the Platform, a new plugin should be created. Is similar to the database drivers used to access different

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

databases. The input is the data offered by the site system. The output is normalized data, able to be processed by any of the modules M1-M12. This data is placed in JSON format. Binary data can be placed in BASE64 format. Specific video and audio data will be placed in commonly used format mp4, wav, and avi.

2. Persistence system. It is the place where indexed and hash data are stored. It is used to quickly access the site data and also to have cross-access from site to site. This persistence content is synchronized all the time, via the P2P network and allows one node to give access to clients to data from the current site and from all the other sites. The implementation will be a Secured Data Store, exposing APIs, and persisting data in one of the File Systems, RDBMS, NOSQL.
    a. Process services. Is the place where are located all T1-T7 Tools (Table 4 SHIFT Tools by Use Cases)
3. P2P Network component. It is used to:
    a. Locate the list and address all nodes.
    b. Implement a discovery mechanism.
    c. Implement a synchronization mechanism.
    d. Implement seed mechanism.
4. APIs (HTTP(S) Web Services), which are used to expose all the functionalities to clients.

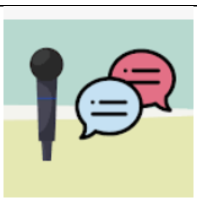
In detail, the node representation is shown in the next figure (Figure 6. Detailed Node components)

The symbols used by tools are:

T1-IV Image to video	
T2-VS Video&Text to Speech	
T3-HA Haptic Interaction	

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

T4-AN Audio Narrative	
T5-CT Contemporary Translation	
T6-AF Accessibility Framework	
T7-AT Accessible Text-to-Speech	

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

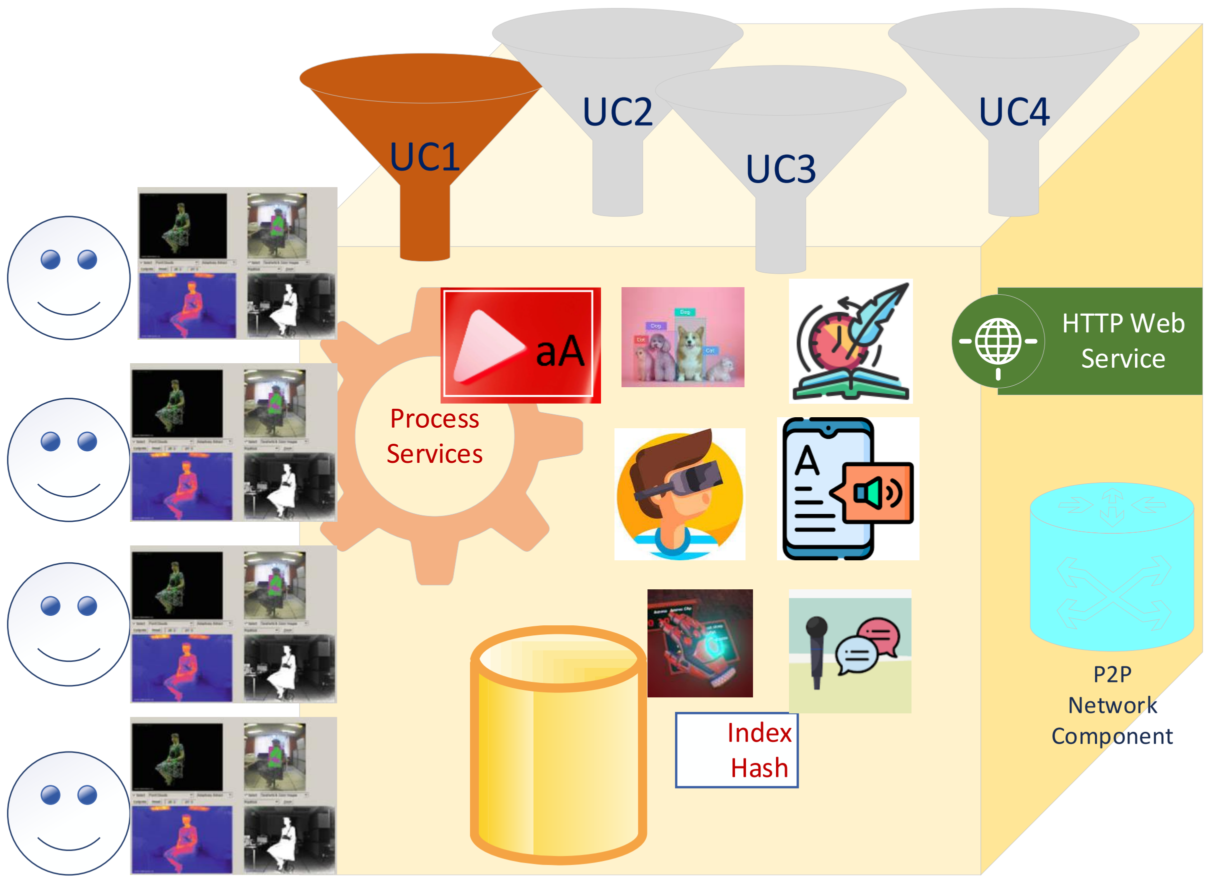


Figure 6. Detailed Node components

In this presentation, we can distinguish the 6 tools used to implement the processes in one node. The plugins are specific to use cases and are used to capture input data. It is considered that the plugins are placed in the nodes deployed on a specific site. That means that a node implemented in one site usually implements just the plugin for that site. A plugin should treat all types of data. It is a multimodal plugin, accommodating images, video, text, etc.

The data is presented to the user by using the clients. The clients are extensions to actual viewers accessible for use cases, on their specific site.

There will be multimodal viewers, able to present the information as described in the requirements (D1.1 (SHIFT-D1.1, 2023)).

The client (art Wallet) offers functionality to:

Connect to a backend node

Send requests to the backend node

Get data (processed) from the backend node

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

Expose data, in a user interface to final clients. Data exposure will consider the requirements of this project related to accessibility,

There will be two flavors of clients:

- A text-based used just for downloading content
- A full web-based UI, accessible locally or on a cloud. It will implement the mechanism to present data on all employed devices.

A representation of the logical view of the client is in the next figure (Figure 8. Client component).

The haptic and 3D representations have their representation offered by specific modules.

The UI user interface for Admin role will be like in the next figure (Figure 7. WEB Client for admin user).

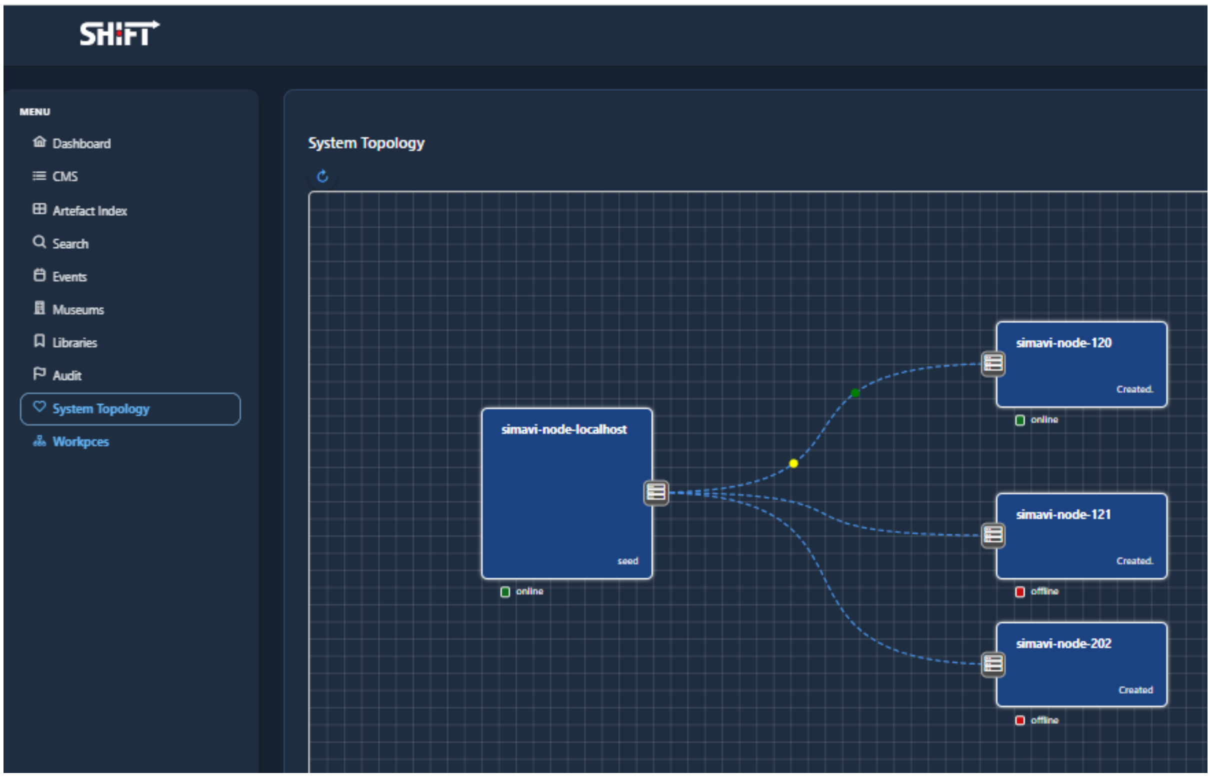


Figure 7. WEB Client for admin user

The logical diagram for the client component is presented in the next figure (Figure 8. Client component).

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

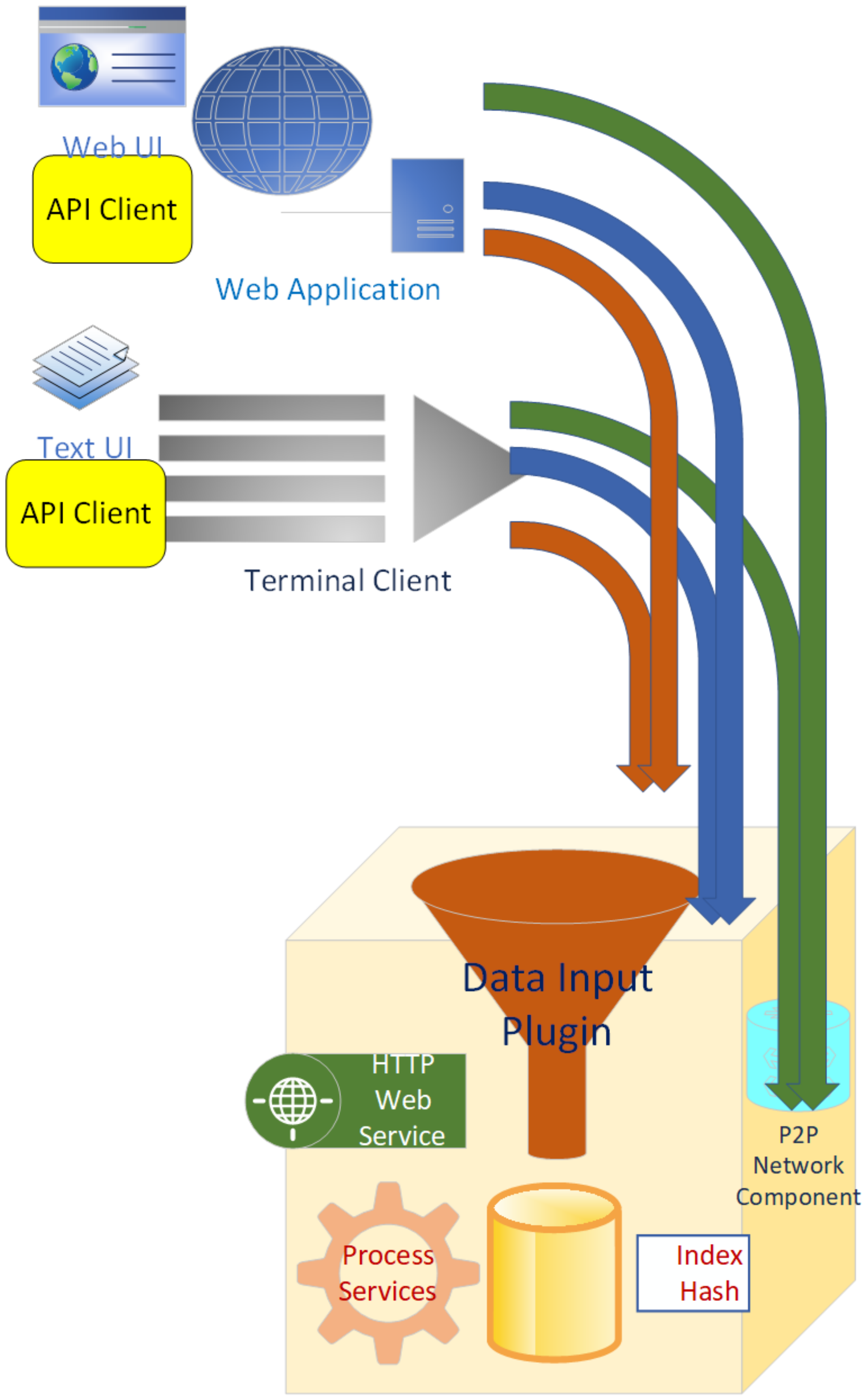


Figure 8. Client component

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

5.7. Process View

This view presents the actions and the workflow of actions possible with the SHIFT Platform. The description is from the Platform point of view.

We are defining three groups of processes:

1. Node management. It is a group of horizontal processes, used to initiate, place in the network and use the nodes. In the next figure (Figure 9. Process View) they are marked in yellow color. The main processes are:
    a. Seed a node. That means taking the initial node from the repository, installing it on a server, and creating a link to the first node of the P2P network. The seed does not send a request to a node. If it is identified, the node answers with an acknowledgment. Next, the seed node requests the list of all nodes. The node that sent the acknowledgment will send also the list of all nodes.
    b. Synchronize a node. This is necessary for a Seed node, but also for all the nodes after an update in the P2P network. A node (say Node A) having the list of all nodes will send a request for synchronization to a node from the list (say Node B). If node B answers that it is already synchronized, then the node who made the request (A) will ask for a chain of nodes available. Node B will deliver the chain of nodes registered on its side. If the requested node(B) does not respond that it is synchronized, node A makes the same request to another node, say Node C., and tries to synchronize. When the synchronization is finished, Node A is marked as synchronized.
    c. Use a node. This is valid for all nodes. The usage of a node means allocating data storage and placing data on the data storage.
2. Data and knowledge management. It is a group of vertical processes, used for placing and processing the data. In the next figure (Figure 9. Process View) they are marked in dark blue color. The main vertical processes are:
    a. Storage, index, and hashing. This process is used first to place data in the data store, and then construct the summary for the stored data.
    b. Normalization, Process AI, process Models. This kind of process is used to apply algorithms to data and to produce knowledge. Again, a chain of a block is created for the knowledge derived.
    c. Filter, Retrieval. This kind of process first applies to the chain of blocs. If the information is found there, the next search is performed in full blocks, where full data or knowledge exists.
3. Helper processes. We are including here processes not directly involved in the data and knowledge delivery but assisting these processes. We are mentioning here: all specific services associated with Tools T1-T7.
    a. User Access and Authorization
    b. Auditing
    c. Statistics
    d. Link with external systems

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

e. Cashing mechanism
f. Version management

5.7.1. NODE MANAGEMENT

The processes presented for the management of nodes (horizontal processes) are represented in the sequence diagram below (Figure 9. Process View)

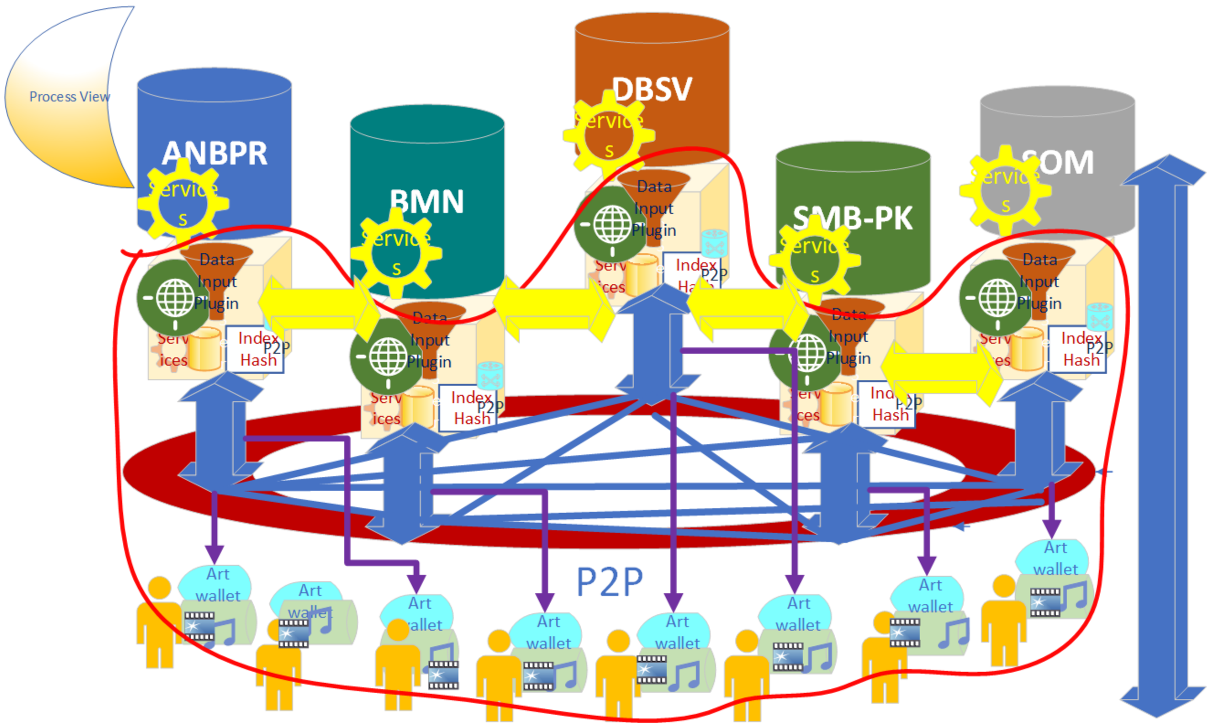


Figure 9. Process View

The sequence diagram for the initial creation of a node is presented in the next figure (Figure 10. Node sequence diagram).

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

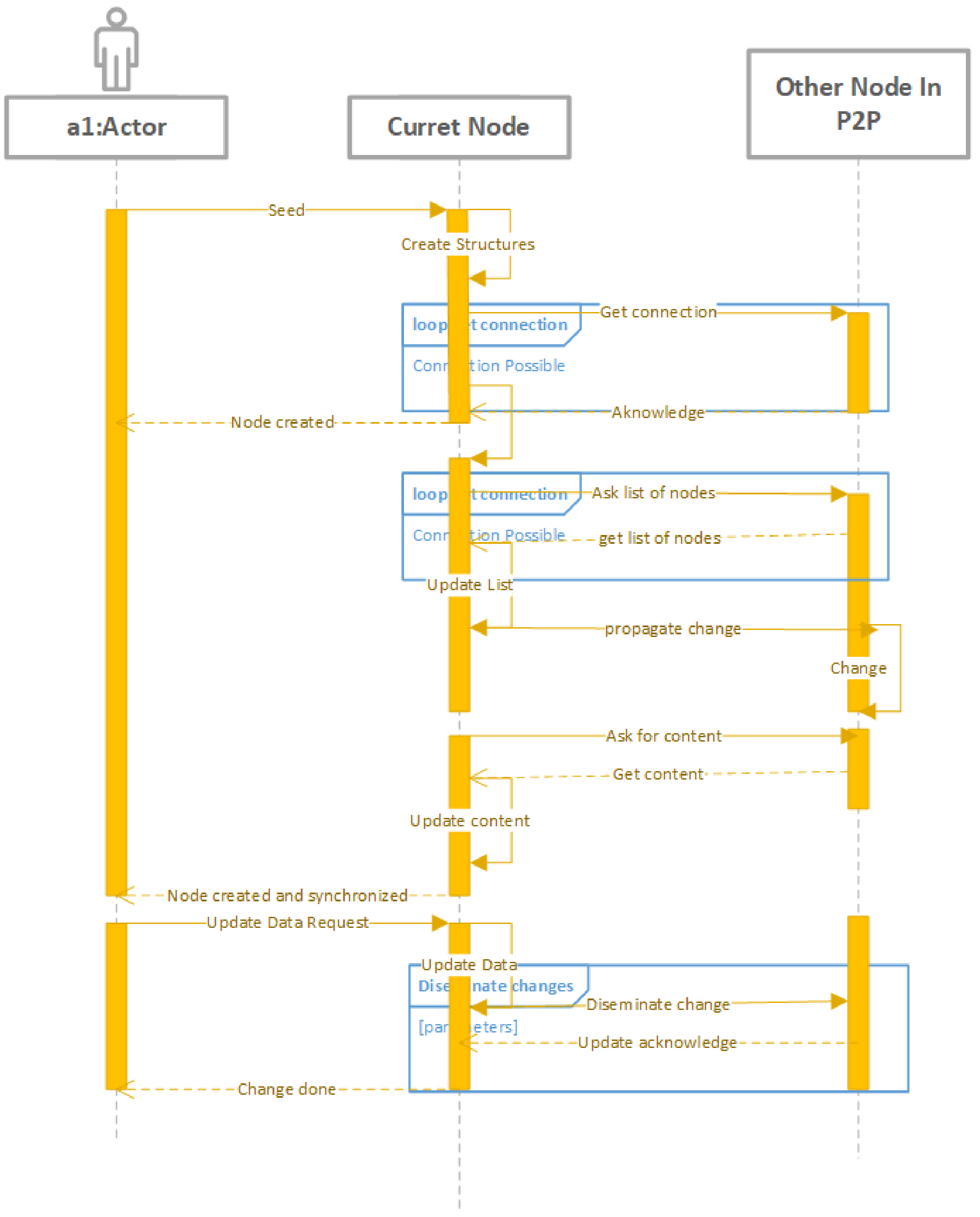


Figure 10. Node sequence diagram

The sequence diagram for disseminating the information between nodes (synchronization) is presented in the figure below (Figure 11. Node synchronization process sequence diagram)

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

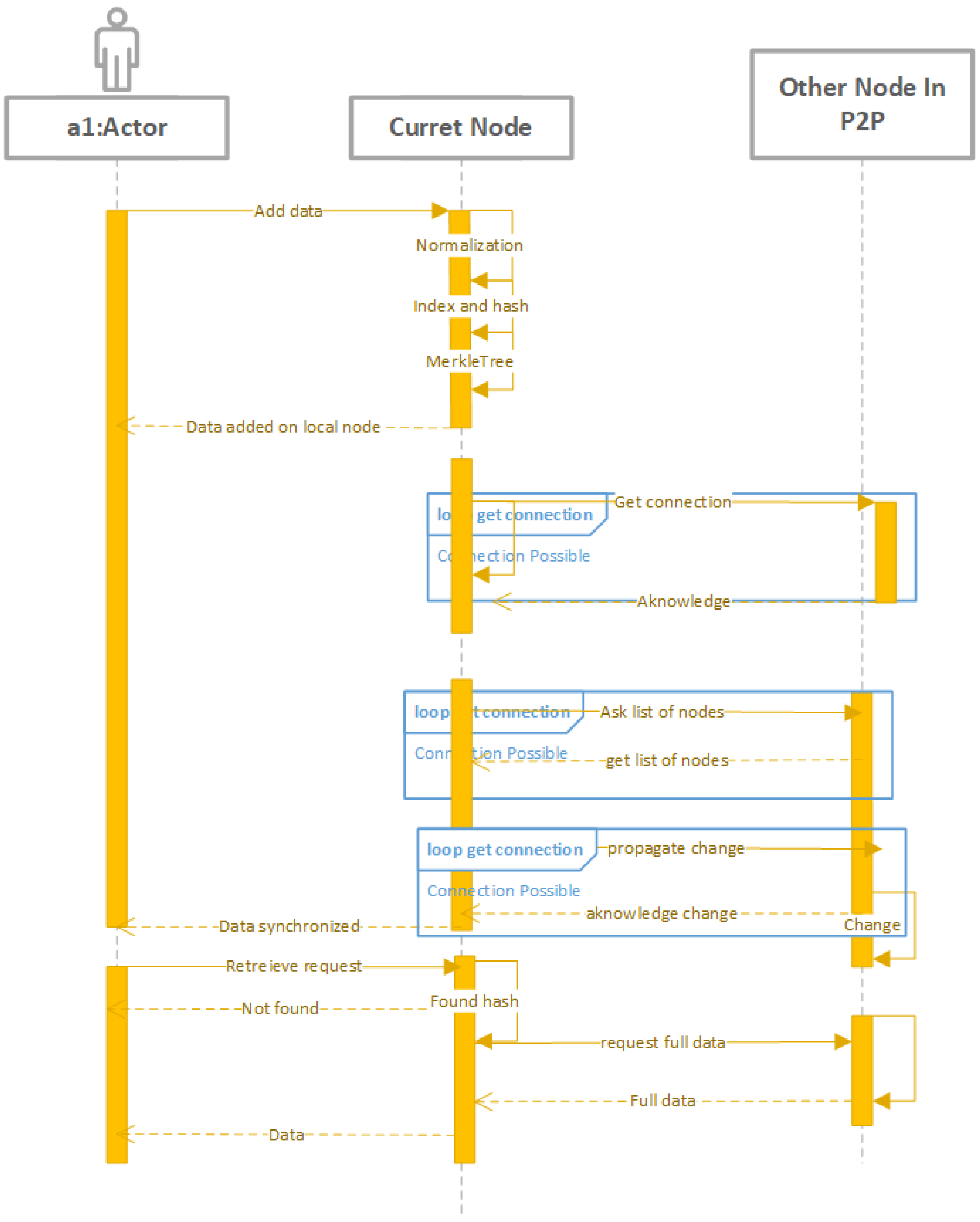


Figure 11. Node synchronization process sequence diagram

5.7.2. DATA AND KNOWLEDGE MANAGEMENT

When processing data and knowledge, the main element used is called "Workspace".

In a workspace, there are placed all the elements necessary for an exhibition.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

The creation of a workspace is done by a curator. For this reason, the next subchapters will include the processes and the sequence diagrams as they are planned for the curator role.

For the end user (visitor) role, things are much simpler. They just access an exhibition, and request information available. No interaction with any tools, as this interaction is done by the curator, and all the information, in any format, is prepared. The visitors will just use the client (art wallet) to access data.

For some outputs, the services are called, but this is transparent for the final user.

This is why, we are not including here processes or sequence diagrams from the final user point of view.

Then in the workspace are placed all the artifacts available, then specific services are called on artifacts, which will result in the creation of other artifacts, placed again in the workspace.

Finally, a workspace created for an exhibition will group all the inputs and all the created elements as a result of working with the tools.

We are presenting now, the sequence diagram for the usage of each tool.

In each sequence diagram, there are two common elements:

The actor a1: Curator. Is the person who initiates a workspace, and is working with the tool.

The Workspace. It is the place where all the data and knowledge is stored. Also is the software element used by the curator when interacting with the system.

Important to notice that the workspace can contain full data (for example images), or just link to the input images. In the case of a link, the platform can present the final user with the correct output, by using the link.

Also, some processed information is in the form of output data (image, video, text), or is a link to an API that will deliver the correct data for the final user.

The sequence diagrams refer to the module developed in this project. The module will be described in the next chapter view 5.8.

The sequence diagrams consider that there is no workspace created yet when a tool is used.

As a workspace can store the work of several tools, and each tool can be used many times on the same workspace, in the sequence diagrams presented in this chapter, the first part (Request and create a workspace) is no longer needed, and instead, open an existing workspace is necessary. All the other actions are the same.

We are just enumerating them now, and refer to the full description in the next chapter.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

Table 8 SHIFT List of Modules

Code	Module
M1-FOS	Foreground/background object segmentation
M2-MOS	Physics-informed machine learning algorithms and video generation
M3-ASR	Action sequence recognition within the CH video repository
M4-NLP	Comprehensive textual representation of assets based on NLP approaches
M5-TEL	Modeling Temporal Evolution of Language for Cultural Asset Curation
M6-TVS	Video/Text to Speech production tool
M7-HAP	Haptic Techniques for 3D Digital Asset Perception
M8-3DP	An accessible framework of inclusive museum exhibits for 3D digital asset perception
M9-FEX	Cultural Asset Pre-processing and Feature Extraction for Media Curation
M10-ASO	Multimedia Cultural Asset Curation Based on Association by Design
M11-DRM	Digital Rights Management
M12-PBK	Main Platform, backend
M13-PFB	Main platform client (Wallet)
M14-COM	Communication and Integration Platform

5.7.2.1. T1-IV IMAGE TO VIDEO

Tool to enhance Photos / Paintings to Short Videos.

For this tool, the curator places several images on the workspace and then calls the modules of the tool to process input images and give the required output.

First images are preprocessed and used to create videos. Next action sequences are detected. Using the results obtained from several processings, associations between elements are searched and presented to the system.

Finally, the Data Rights Management tool is used to inform the user about the rights to use the content of the workspace.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

The sequence diagram representing the processing flow inside this tool is presented in the next figure. (Figure 12. T1-IV sequence diagram).

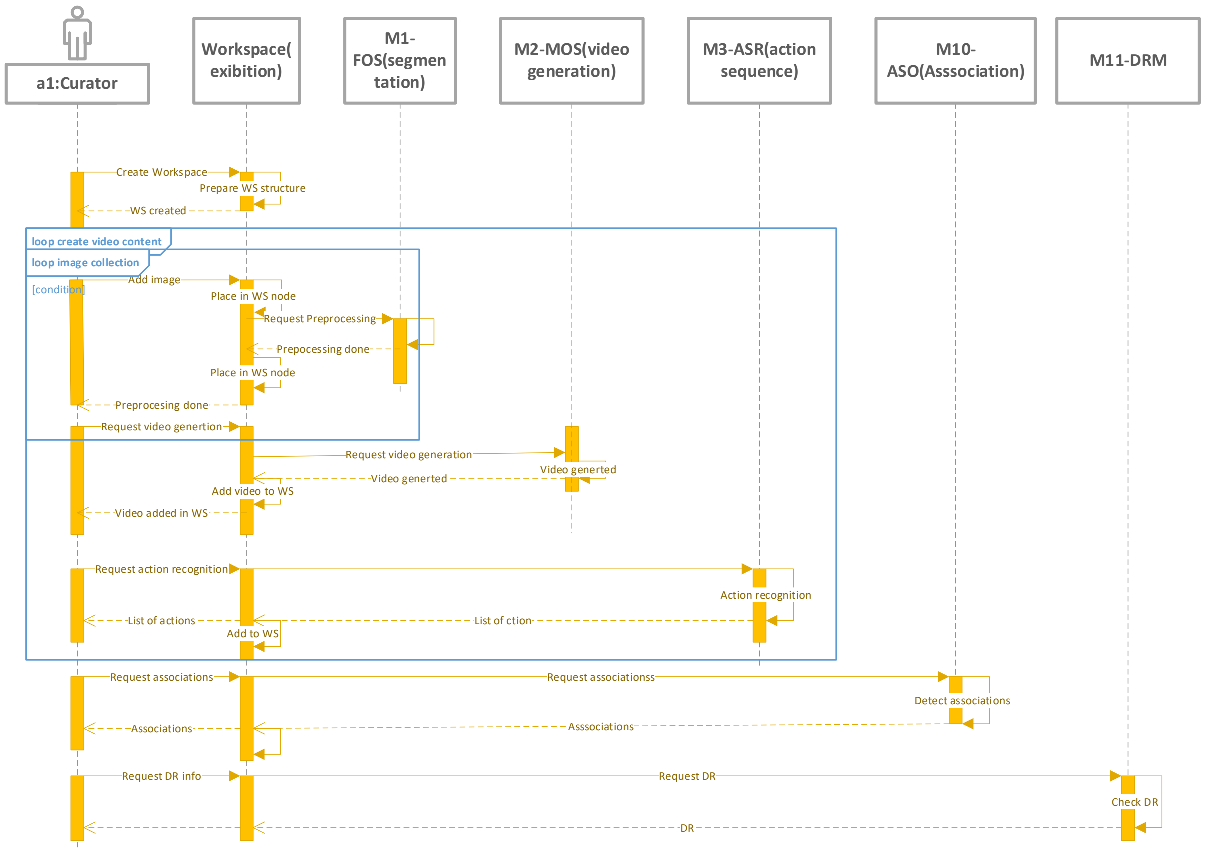


Figure 12. T1-IV sequence diagram

5.7.2.2. T2-VS VIDEO TO SPEECH

Audio tool capable of interpreting visual stimuli (e.g., actions explained in visual sequences)

The input of this tool can come from an existing video, or from a video obtained by using the tool T1-IV.

The video frames, the sequence of actions, are then processed by text and video to speech, and audio output is created.

The audio output can be further processed for feature extraction.

Finally, the Data Rights Management tool is used to inform the user about the rights to use the content of the workspace.

The sequence diagram representing the processing flow inside this tool is presented in the next figure. (Figure 13. T2-VS sequence diagram).

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

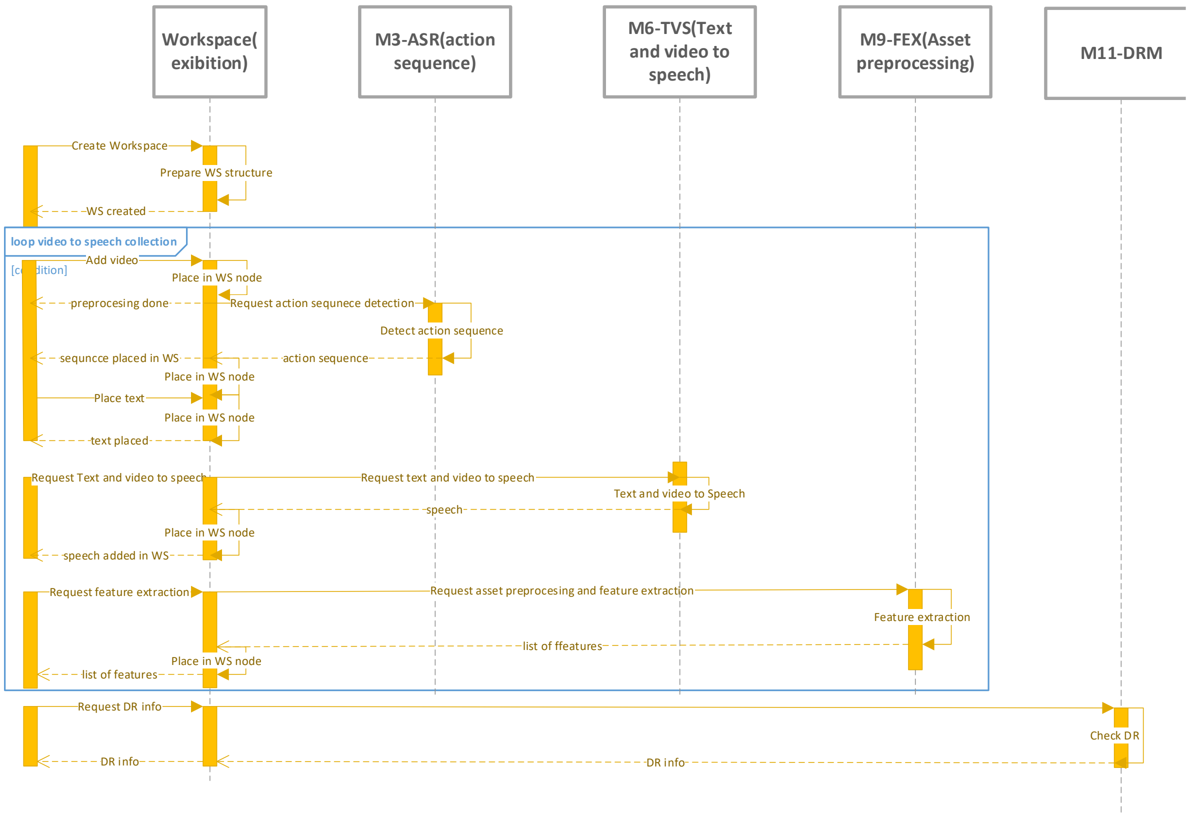


Figure 13. T2-VS sequence diagram

5.7.2.3. T3-HA HAPTIC INTERACTION

A tool that translates physical objects to digital objects and uses haptics to "feel" the objects. To implement haptic interaction with 3D digital tangible and intangible cultural heritage assets, augmenting the user experience (UX) with new interaction paradigms that can be used in situ or remotely.

The input of this tool can come from existing physical objects, or digital twins of physical objects. For physical objects first digital twins (3D representations are created).

Next 3D digital twins are passed to the Haptic tool which allows the user interaction with them.

In this case, the workspace is used to store the digital twins and the haptic representation of them.

Finally, the Data Rights Management tool is used to inform the user about the rights to use the content of the workspace.

The sequence diagram representing the processing flow inside this tool is presented in the next figure. (Figure 14. T3-HA sequence diagram).

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

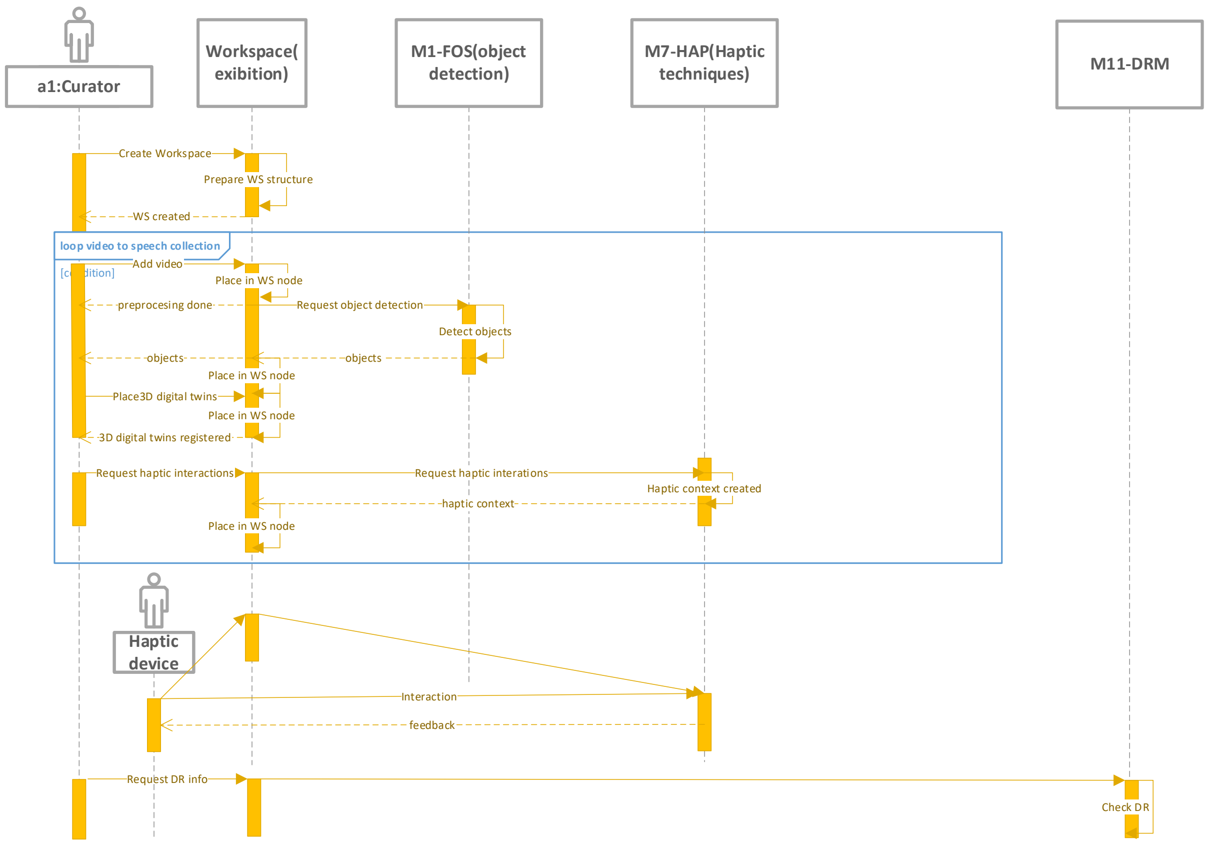


Figure 14. T3-HA sequence diagram

5.7.2.4. T4-AN AUDIO NARRATIVE

The tool automatically can provide complementary information regarding the cultural heritage assets (books, paintings, photos)

The input of this tool can come from an existing video, or from a video obtained by using the tool T1-IV. Also, text (in contemporary or old format) can be used as input.

The video inputs are used to create NLP textual representations

For their textual representations obtained from M4-NLP or directly registered by users, linguistic models are created to make it possible to reformulate the content.

The (reformulated) text content is then processed by text to text-to-speech module, and the audio narrative is obtained.

Finally, the Data Rights Management tool is used to inform the user about the rights to use the content of the workspace.

The sequence diagram representing the processing flow inside this tool is presented in the next figure. (Figure 15. T4-AN sequence diagram).

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

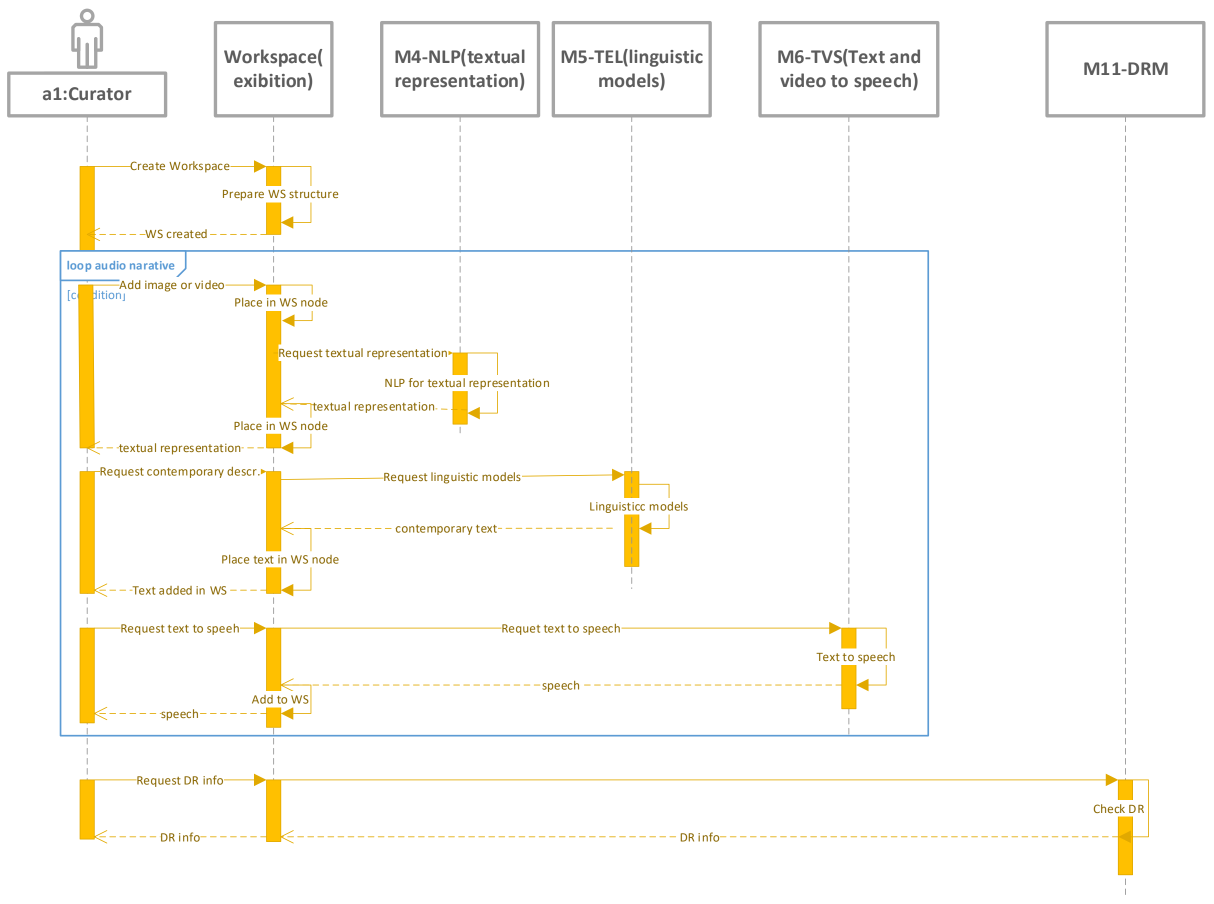


Figure 15. T4-AN sequence diagram

5.7.2.5. T5-CT CONTEMPORARY TRANSLATION

A tool that translates historical meaning into more contemporary language and for auto-tagging/ auto-categorization of cultural heritage resources.

The input of this tool can come from an existing video, or from a video obtained by using the tool T1-IV. Also, text (in contemporary or old format) can be used as input.

The video inputs are used to create NLP textual representations

For their textual representations obtained from M4-NLP or directly registered by users, linguistic models are created to make it possible to reformulate the content.

The (reformulated) text content is then sent to users.

Feature extraction or association detections can be also requested via this tool.

Finally, the Data Rights Management tool is used to inform the user about the rights to use the content of the workspace.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

The sequence diagram representing the processing flow inside this tool is presented in the next figure. (Figure 16. T5-CT sequence diagram).

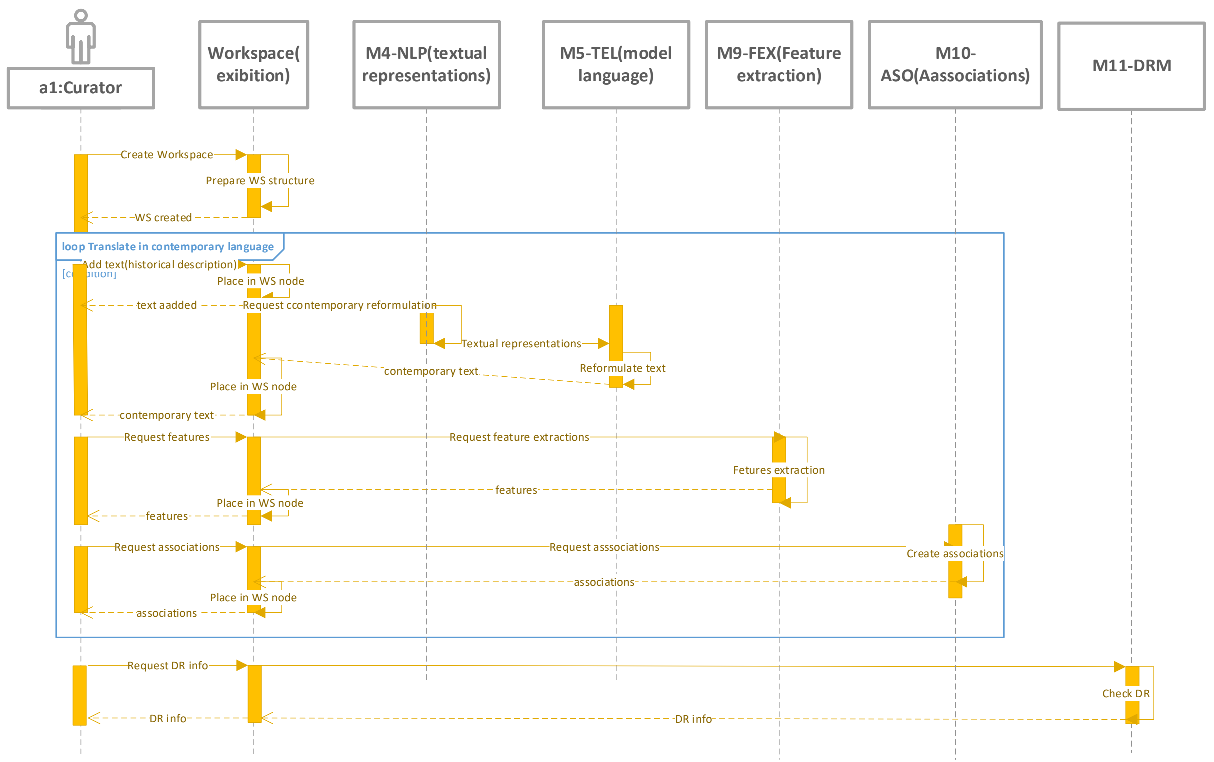


Figure 16. T5-CT sequence diagram

5.7.2.6. T6-AF ACCESSIBILITY FRAMEWORK

Comprehensive intuitive and accessible tool for all (including individuals with disabilities) multimodal storytelling of cultural heritage assets.

The input of this tool can come from an existing video, or from a video obtained by using the tool T1-IV. Also, text (in contemporary or old format) can be used as input.

The video inputs are used to create NLP textual representations

For their textual representations obtained from M4-NLP or directly registered by users, linguistic models are created to make it possible to reformulate the content.

The (reformulated) text content is then sent to users.

The text can be next converted into speech.

Multimodal representations are another possibility offered by the system.

Feature extraction or association detections can be also requested via this tool.

Finally, the Data Rights Management tool is used to inform the user about the rights to use the content of the workspace.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

The sequence diagram representing the processing flow inside this tool is presented in the next figure. (Figure 17. T6-AF sequence diagram).

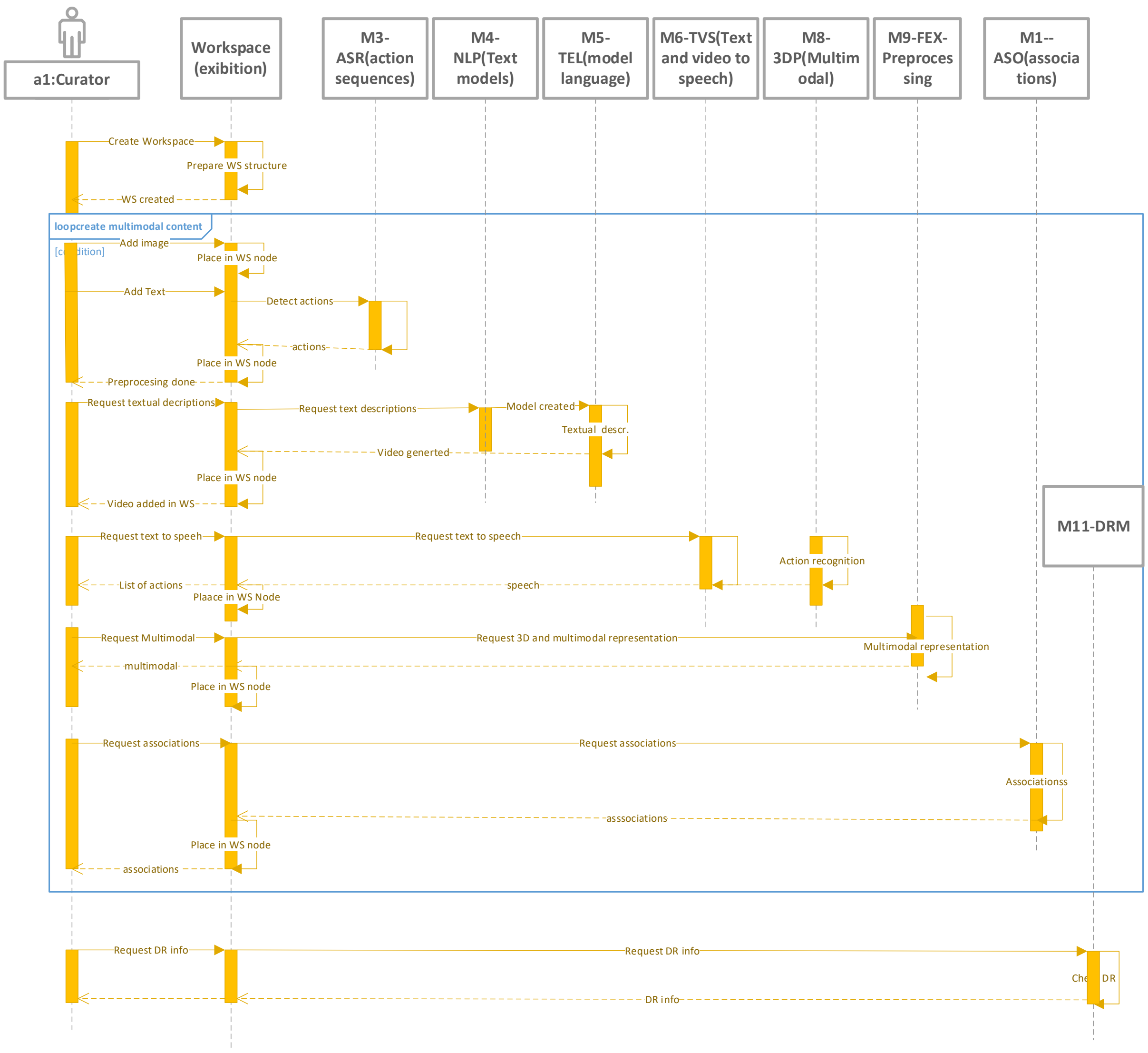


Figure 17. T6-AF sequence diagram

5.7.2.7. T7-AT ACCESSIBLE TEXT-TO-SPEECH

Comprehensive intuitive and accessible tool for all (including individuals with disabilities) multimodal storytelling of cultural heritage assets.

The input of this tool can come from an existing video, or from a video obtained by using the tool T1-IV. Also, text (in contemporary or old format) can be used as input.

The video inputs are used to create NLP textual representations

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

For their textual representations obtained from M4-NLP or directly registered by users, linguistic models are created to make it possible to reformulate the content.

The (reformulated) text content is then processed by a text-to-speech module, and multimodal storytelling is obtained.

Finally, the Data Rights Management tool is used to inform the user about the rights to use the content of the workspace.

The sequence diagram representing the processing flow inside this tool is presented in the next figure. (Figure 18. T7-AT sequence diagram).

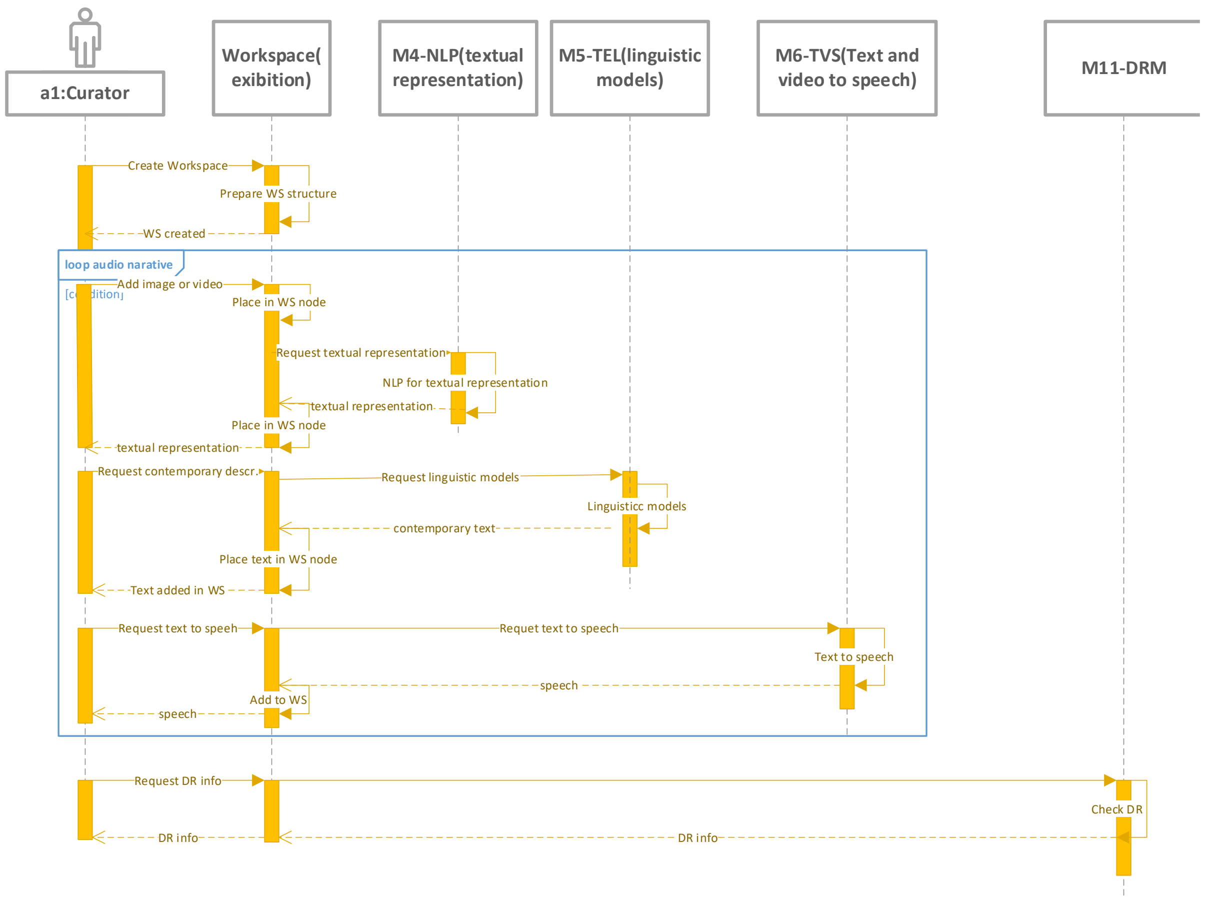


Figure 18. T7-AT sequence diagram

5.7.2.8. PROCESSES SPECIFIC FOR EACH USE CASE

We have presented the processes and the sequence diagrams for each tool proposed, based on the business needs and considering modules designed for each tool.

Now we are summarizing the usage of processes based on the four use cases. The point of view is that of a curator. As already mentioned, the final users (visitors) just access the exhibition (workspace) and search for the desired output.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

In each use case, the first action of a user is to create a workspace. Next, depending on the use case, different tools are used, and the workspace is updated.

When presenting the processes for each use case, we consider that the workspace is already created.

The usage of tools does not impose a specific sequence. Also, a tool can be called several times.

All the information referred to here is stored in a node of the distributed Data and Knowledge Mesh. The distribution and synchronization of the nodes are subject to the processes presented in the previous chapter (5.7.1)

UC1- 19TH TO MODERN DAYS SERBIAN PAINTINGS AND MODERN ART

The process (tools) involved are:

- T1-IV Image to video
- T2-VS Video to Speech
- T4-AN Audio Narrative
- T5-CT Contemporary Translation
- T6-AF Accessibility Framework

UC2- EXPERIMENTING WITH THE TRANSFORMATION OF MEDICINE AND PHARMACY

- T1-IV Image to video
- T2-VS Video to Speech
- T3-HA Haptic Interaction
- T4-AN Audio Narrative
- T5-CT Contemporary Translation

UC3- ROMANIAN HISTORY AND CUSTOMS EXPLAINED TO DIGITAL NATIVES

- T1-IV Image to video
- T4-AN Audio Narrative
- T5-CT Contemporary Translation
- T6-AF Accessibility Framework
- T7-AT Accessible Text-to-Speech

UC4- CH EXHIBITION AS VISITOR'S JOURNEY, WITH NO SENSING BOUNDARIES

- T2-VS Video to Speech
- T3-HA Haptic Interaction
- T6-AF Accessibility Framework
- T7-AT Accessible Text-to-Speech

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

5.8. Physical View

The physical view presents the software elements used to create the tools described in the Use Case View and details the implementation means of all the software components. Each tool is created based on several software modules, with functionality covering the tool's needs.

The description starts with the presentation of individual modules and goes further to describe how the modules are placed in tools and finally in the distributed environment based on nodes.

5.8.1. LIST OF ALL MODULES

We are presenting here the list of all modules of the system. The provider of the tool is indicated also. See Subchapters 5.7.3-5.7.16 for details. The table below indicates all functionalities planned to be developed for each module in the Platform.

Table 9 SHIFT List of Modules by Owner.

Code	Module	Owner
M1-FOS	Foreground/background object segmentation	MDS
M2-MOS	Physics informed machine learning algorithms and video generation	QMUL
M3-ASR	Action sequence recognition within the CH video repository	MDS
M4-NLP	Comprehensive textual representation of assets based on NLP approaches	UAU
M5-TEL	Modeling Temporal Evolution of Language for Cultural Asset Curation	UAU
M6-TVS	Text and video-to-speech production tool	AUD
M7-HAP	Haptic Techniques for 3D Digital Asset Perception	FORTH
M8-3DP	An accessible framework of inclusive museum exhibits for 3D digital asset perception	FORTH
M9-FEX	Cultural Asset Pre-processing and Feature Extraction for Media Curation	UAU
M10-ASO	Multimedia Cultural Asset Curation Based on Association by Design	UAU
M11-DRM	Digital Rights Management	QMUL
M12-PBK	Main Platform, backend	SIM

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

M13-PFB	Main platform client (Wallet)	SIM
M14-COM	Communication and Integration Platform	SIM

5.8.2.TEMPLATE USED TO DESCRIBE THE MODULES

The modules will be described using the template given in this subchapter.

There are specific paragraphs to be included.

Name: <Module Name>

Provider: <Company abbreviation>

Work Package<WP and Tasks where the module was developed>

Part of tools <Tools where the module is used**>**

Short description of the final module:

<text description>

Full description is/will be part of <deliverable>

Technologies used

<OS, Platform, Libraries, etc., text description>

Is containerized

<Container type, How to use it>

Installation and configuration

<text description>

Produced data (Outputs)

<List of data produced, (format, dimension, availability):>

Consumed data. (Inputs)

<List of topics. (format, dimension, availability):>

Asynchronous data exchange

<Explain if necessary>

APIs

<describe APIs I any**)**

Third-party libraries used

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

<enumerate the libraries>

User interface

<If available, describe if it is web, text-based, Windows-based, Java-based, etc.).

How to test the system

<text description>

5.8.3. M1-FOS FOREGROUND/BACKGROUND OBJECT SEGMENTATION

Provider: MDS

Work Package: WP2/T2.1: Deep-learning architecture design and implementation for foreground/background object detection.

Part of tools: T1-IV Image to video, T3-HA Haptic Interaction.

Short description of the final module:

Our pipeline shall execute preprocessing operations. It shall accept an image as input and generate multiple images containing potentially altered details that have evolved. Each resultant image shall be paired with its respective description. Furthermore, an option will be available for constructing three-dimensional models of the objects depicted in the images.

In order to achieve the results, the following steps are considered:

- **Super Resolution (SR)**: As we mentioned before, we utilize a deep learning-based model that significantly improves the resolution of low-quality images. This model employs advanced techniques such as CNNs and GANs.
- **Segmentation:** For segmentation, our objective is to leverage Segment Anything Model (SAM) tailored for cultural heritage data, especially for our coin dataset. The retraining session involves fine-tuning the model with a carefully curated dataset. In our approach, we use the World Coins Dataset as a base, from which we select 300 images for mask generation.
- For **3D modelling**, following the successful generation of high-quality segmentation masks, we implement the One-2-3-45 architecture. This innovative architecture is designed to create multi-view 3D meshes of the objects. It works by processing the segmented 2D images and reconstructing them into 3D models.

Full description is part of D2.1 Chapter 2.2 (SHIFT-D2.1, 2023)

Technologies used:

OS: Windows

Platform: Python 3

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

Libraries: Anaconda (all additional packages will be provided through a .yml file)

Is containerized

no

Installation and configuration:

Anaconda Environment

Produced data (Outputs):

1. Images (.png/.jpg)
2. Text (.txt)

Consumed data. (Inputs):

Images (.png/.jpg)

Asynchronous data exchange:

The task operates asynchronously, wherein we will receive images and subsequently provide the generated files.

APIs

N/A

Third-party libraries used

https://github.com/facebookresearch/segment-anything

User interface

CLI (python script)

How to test the system

- Currently, the end user can assess the quality of the pipeline's output. We are in communication with our partners and have received feedback on the system's performance.
- The primary evaluation criterion is the visual result, given the absence of a ground truth for comparison.
- In the next phase, we will explore an independent evaluation system, allowing users to autonomously utilize the pipeline.

5.8.4. M2-MOS PHYSICAL INFORMED MACHINE LEARNING ALGORITHMS AND VIDEO GENERATION

Provider: QMUL

Work Package: WP2/T2.2: Physics informed deep-learning network architecture.

Part of tools: T1-IV Image to video.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

Short description of the final module:

The module provides the following 2 functionalities:

1. For a given input image, the outcome from the task will result in the generation of a video sequence.
2. The video sequence generation will insert the artistic elements to make the videos more accessible and improve the appeal of the content.

Following the extensive review, during the first year of SHIFT project implementation, the input gathered from WP1 on use-cases consisting of an overview of cultural heritage assets have been analysed. Subsequently, the architecture of U-Net deep-learning model has been chosen as a basis for the development of stable diffusion architecture for the implementation of SHIFT computer vision toolkit to generate motion sequences.

Full description is part of D2.1(SEN[3]) Chapter 3 (SHIFT-D2.1, 2023)

Technologies used

OS: Linux PC, (GPU is required)

Platform: Python3

Languages: Python

Libraries: FFMPEG, OpenCV, PyTorch, Conda (for Windows)

Is containerized:

N/A. Native support is offered to access GPU.

Installation and configuration

Module installation requires the execution of Python dependencies and associated libraries as virtual environment. For the windows, Conda library support will be extended.

Produced data (Outputs)

1. Video (.mp4/.avi)

Consumed data (Inputs)

1. Images (.png/.jpg)

Asynchronous data exchange

- N/A

[3] SEN-Sensitive deliverable

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

APIs

- N/A

Third-party libraries used

- OpenCV libraries
- U-NET libraries
- Diffusion model libraries

User interface

- CLI (python script)
- Automated execution using Restful APIs

How to test the system

- The generation of the video sequences will be used to verify the module execution.
- The evaluation of the generated video sequences will be assessed based on the subjective and qualitative metrics that are gathered from the end-users.
- As reported in D2.1, the adoption of stable diffusion models will be subjected to community-based evaluation, that leads to the overall acceptance of the video sequences by the end-users.
- Formal evaluation methods and acceptance of end-users will be reported in D2.2 (due by M30).

5.8.5. M3-ASR ACTION SEQUENCE RECOGNITION WITHIN CH VIDEO REPOSITORY

Name: Action sequence recognition within the CH video repository

Provider: MDS

Work Package: WP2/T2.4: Action sequence recognition within the CH video repository.

Part of tools: T1-IV Image to video, T2-VS Video to Speech

Short description of the final module:

The model will receive a video input and subsequently generate a class label, representing an action selected from the provided vocabulary.

To explore the possibility of expanding the repertoire of actions, we employed a transfer learning methodology. We selected various architectures from the GluonCV library, which offers implementations of cutting-edge deep learning models in the domain of computer vision.

Our approach involved the removal of the final layers in these networks, followed by the implementation of new layers designed to produce the specific output set of interest. Subsequently, we conducted training using our newly created dataset. To be more precise, our focus was on transferring the models from the Kinetics 400 dataset to UCF101.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

Full description is part of D2.1(SEN) Chapter 4 (SHIFT-D2.1, 2023)

Technologies used:

OS: Windows

Platform: Python 3

Libraries: PyTorch, GlyonCV, Anaconda (all additional packages will be provided through a .yml file),

Is containerized: no

Installation and configuration:

Anaconda Environment

Produced data (Outputs):

Text (Label or Action .txt)

Consumed data. (Inputs)

Video (.mp4)

Asynchronous data exchange:

Yes, for uploading videos

APIs

N/A

Third-party libraries used

https://github.com/dmlc/gluon-cv

User interface

CLI (python script)

How to test the system

- Currently, the end user can assess the quality of the pipeline's output. We are in communication with our partners and have received feedback on the system's performance.
- There is no definitive reference to assess the rate of identifying actions, so the evaluation necessitates human intervention. We are contemplating the creation of a test set to verify the system's reliability.
- In the next phase, we will explore an independent evaluation system, allowing users to autonomously utilize the pipeline.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

5.8.6. M4-NLP COMPREHENSIVE TEXTUAL REPRESENTATION OF ASSETS BASED ON NLP APPROACHES

Provider: UAU

Work Package: WP3/T3.1: Comprehensive textual representation of assets based on NLP approaches.

Part of tools: T4-AN Audio Narrative, T5-CT Contemporary Translation, T6-AF Accessibility Framework, T7-AT Accessible.

Short description of the final module:

This component will leverage the knowledge and semantic features towards creating comprehensive textual representations of the CH assets. The devised mechanism will employ state-of-the-art NLP approaches to learn temporal embeddings and apply regularization terms to smooth embedding changes across time.

Full description is part of D3.1(SEN) (SHIFT-D3.1, 2023)

Technologies used

OS: Ubuntu, NixOS

Platform: Python 3, Poetry

Libraries: <not known yet>

Is containerized: no

Installation and configuration

e.g., pip requirements.txt

Produced data (Outputs)

Text

Consumed data. (Inputs)

Image, Text, Table

Asynchronous data exchange

- N/A

APIs

- N/A

Third-party libraries used

e.g., Hugging Face

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

User interface

Terminal

How to test the system

User Preference Testing: Users interact with textual representations generated by the module to rate their comprehensiveness and relevance to CH assets. Feedback reflects the positive effectiveness.

LLM Quality Check: An additional language model evaluates the generated textual representations for accuracy and semantic richness. This reflects assessing how well the temporal embeddings contributes to capturing the evolution and context of CH asset.

5.8.7. M5-TEL MODELLING TEMPORAL EVOLUTION OF LANGUAGE FOR CULTURAL ASSET CURATION

Provider: UAU

Work Package: WP3/T3.2: Modelling Temporal Evolution of Language for Cultural Asset Curation.

Part of tools: T4-AN Audio Narrative, T5-CT Contemporary Translation, T6-AF Accessibility Framework, T7-AT Accessible.

Short description of the final module:

This module will create linguistic models using deep learning algorithms to model associations of temporal evolution that enable time-independent curation and map contemporary asset descriptions with archive representation. The deep learning algorithms will be used to extract and train on inherent linguistic patterns used for the description of cultural assets.

Moreover, the tool aspires to establish linguistic models through the integration of state-of-the-art learning algorithms, designed to understand and represent temporal evolution associations. Harnessing meticulous representations acquired in T3.1, these models facilitate seamless and enhanced execution of time-independent curation tasks, utilizing subtly engineered prompts within an integrated ecosystem to narrow down the temporal focus. A property intrinsically ingrained in novel methods such as LangChain. With the objective of aligning contemporary asset description with archival representations, the tool ensures a comprehensive understanding of artifacts' multifaced contexts, originating from either image-processed captions or linguistic nuances. This synergistic method not only offers refined summarizations and translations, but also further fine-tuning guarantees contextually synchronized and coherent interpretations of CH artifacts.

Full description is part of D3.1(SEN) (SHIFT-D3.1, 2023)

Technologies used

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

OS: Ubuntu, NixOS

Platform: Python 3, Poetry

Libraries: <not known yet>

Is containerized: no

Installation and configuration

e.g., pip requirements.txt

Produced data (Outputs)

Text

Consumed data. (Inputs)

Image, Text, Table

Asynchronous data exchange

- N/A

APIs

- N/A

Third-party libraries used

e.g., Hugging Face

User interface

Terminal

How to test the system

A/B Testing and User Satisfaction: Users presented with linguistic models' outputs, like summarization and style transfer of CH assets of different language models. User preferences and feedback reflect the effectiveness.

LLM Comparative Assessment: An additional language model evaluates outputs for their ability to accurately represent the effect of transforming the text in different styles.

5.8.8. M6-TVS TEXT AND VIDEO-TO-SPEECH PRODUCTION TOOL

Provider: AUD

Work Package: WP3/T3.3: Text and video-to-speech production tool for the affective narration of cultural heritage assets.

Part of tools: T2-VS Video to Speech, T4-AN Audio Narrative, T5-CT Contemporary Translation, T6-AF Accessibility Framework, T7-AT Accessible.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

Short description of the final module:

The module provides the following 2 functionalities:

1. Given (plain) text, the module generates a (.wav) audio with Synthesized Speech.
2. Given Video (ffmpeg available formats) & and time-stamped Subtitles as **extra file** (.srt), the module replaces the audio in the input video with affective synthesized Speech. The synthesized Speech will pronounce the text from the **.srt** file.

Using a low-resource open-source TTS System along with the Speech Emotion Recognition module developed by AUD the tool is able to clone the emotion of an native/natural voice into synthesized speech thus transforming a text into affective synthesized speech.

Through the affective synthesis of speech, the aim is to transform CH items into vibrant and engaging narratives that stimulate the imagination and create strong emotional connections.

Full description is part of D3.2 (SHIFT-D3.2, 2023)

Technologies used

OS: Linux PC, (GPU is optional)

Platform: Python3

Languages: Python

Libraries: onnxruntime, ffmpeg, PyTorch, sox, transformers

Is containerized: No

Installation and configuration

No installation needed / repo of Python scripts. Requires virtualenv and installation of dependencies via pip (will be described on GitHub page of the Module).

Produced data (Outputs)

1. Audio (.wav)
2. Video (.mp4)

Consumed data. (Inputs)

1. Text (.txt) - For now only English
2. time-stamped English subtitles (.srt) & and video (.avi / .mp4)

Asynchronous data exchange

N/A

APIs

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

Ongoing Implementation: https://github.com/audeering/shift

Third-party libraries used

Open-Source TTS Library: https://github.com/MycroftAI/mimic3

User interface

CLI (python script)

How to test the system

Install/Run Python demo from https://github.com/audeering/shift

on (.txt) file to produce speech (.wav) and/or video (.mp4):

- Apt to pronounce English (.txt) and for the future also foreign-language(s) (.txt) with various perceivable emotional intonations/voices.
- Attain a high user satisfaction score for appealing TTS voice(s) along with clear speech pronunciation: Mean Opinion Score (MOS) >=3.

5.8.9. M7-HAP HAPTIC TECHNIQUES FOR 3D DIGITAL ASSET PERCEPTION

Provider: FORTH

Work Package: WP3, T3.4: Haptic techniques for 3D digital asset perception.

Part of tools: T3-HA Haptic Interaction.

Short description of the final module:

This library focuses on providing haptic-based interaction in Virtual Reality (VR) environments and supporting physical interaction via haptic gloves. Through the provided functionalities of the library, users can feel the emulated physical attributes of Cultural Heritage (CH) assets in the VR environment, providing them with haptic feedback. To that force and temperature input can be experienced by users when "touching" virtual 3D digital twins of the CH assets existing in a VR application. Furthermore, the users can also manipulate these CH digital twins with proper gestures (e.g., grab, move, rotate, etc.).

A dedicated VR scene was developed to focus specifically on testing the haptic feedback provided by the selected device (WEART TouchDIVER) on sample CH assets. Multiple 3D CH assets were integrated into the scene with different sizes, textures, and materials, as shown in Figure 8. From left to right, the assets are crafted from gold, soft limestone, stone, glass, and wood, showcasing a diverse range of materials regarding physical attributes. The WEART SDK was integrated into the scene to add haptic effects to the 3D objects. A range of haptic attributes is offered through the "*WeArtTouchableObject*" component, including:

• **Temperature:** The temperature value implemented on the target thimble or thimbles, ranging from 0.0 to 1.0. A value of 0.5 represents the environmental

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

temperature, while lower values indicate colder sensations and higher values convey hotter sensations.

• **Force:** The force value applied on the target thimble(s), ranging from 0 to 1. A value of 0.0 indicates no force, while a value of 1.0 represents maximum force.

• **Texture:** The type of texture rendered on the thimble or target thimbles, represented by an index ranging from 0 to N. A selection of textures that can be applied to the haptic feedback is provided, such as crushed rock, plastic, and more, as shown in Figure 9.

• **Volume Texture:** This attribute allows developers to configure the intensity of the texture rendering, adjusting the strength of the tactile feedback provided by the texture.

• **Graspable:** Enabling this option grants users the ability to grasp and lift the virtual object with virtual hands.

Full description is part of D3.3 (SHIFT-D3.3, 2023)

Technologies used:

OS: Windows

Platform: Unity 3D

Languages: C#

Libraries: Libraries in Weart SDK package for Unity: https://weart.it/developer-guide/ (WeArt.Core, WeArt.TouchEffect, etc.).

Middleware: Weart Middleware https://www.weart.it/docs/middleware/

Is containerized:

No, the module is provided as a Unity3D asset.

Installation and configuration

- Download and install the Weart middleware
- Download the unity.package
- Import the unity. package to an empty Unity3D project
- Add to the scene the WeArtController, the WEARTLeftHand, the WEARTRightHand, and the TouchableCube prefabs
- Configure the TouchableCube through the WeArtTouchableObject script with the haptic effects of your choice
- Connect the haptic gloves doggle to your computer
- Open the Weart middleware application and connect to the haptic gloves
- Connect the Oculus headset to your computer
- Wear the haptic gloves and the Oculus headset
- Run the Unity3D application as a Windows application

Produced data (Outputs)

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

Haptic effects (thermal and force queues)

Consumed data. (Inputs)

Hand tracking data from the haptic glove.

Asynchronous data exchange

N/A

APIs

N/A

Third-party libraries used

Weart SDK for Unity

User interface

It is embedded in a Unity application

How to test the system

- Install the Middleware to the PC
- Wear the TouchDIVER device
- Wear the Meta Quest Pro headset

Run the provided Unity application and check for:

- Enable the conveyance of essential information (e.g., size and material) about digital assets through haptic feedback, aiming for a minimum perceivability rate of 70%
- Attain a user satisfaction rate of at least 75% regarding the effectiveness of haptic feedback in perceiving the virtual artifacts
- Achieve a positive overall haptic experience of at least 70%

5.8.10. M8-3DP ACCESSIBLE FRAMEWORK OF INCLUSIVE MUSEUM EXHIBITS FOR 3D DIGITAL ASSET PERCEPTION

Provider: FORTH

Work Package: WP3, T3.5: Accessible framework of inclusive museum exhibits for 3D digital asset perception.

Part of tools: T6-AF Accessibility Framework.

Short description of the final module:

An accessible framework for Unity 3D application, providing accessibility features to VR museum 3D digital CH exhibits. It aims to support a personalized accessible environment for people with visual impairments, through multimodal interaction

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

and multifaceted information, considering their accessibility needs and individual characteristics. To achieve this the framework provides numerous accessibility features (e.g., magnifying glass, color change, etc.) and combines customized audio descriptions of the CH assets, tailored to the user's needs and preferences, 3D sounds, and haptic feedback by incorporating the output of the rest modules of WP3.

Full description will part of D3.4 (SHIFT-D3.4, 2024)

Technologies used:

OS: Windows

Platform: Unity 3D

Languages: C#

Libraries: Libraries in Weart SDK package for Unity: https://weart.it/developer-guide/ (WeArt.Core, WeArt.TouchEffect, etc.).

Middleware: Weart Middleware https://www.weart.it/docs/middleware/, oculus application https://www.meta.com/quest/setup/?utm_source=www.meta.com&utm_medium=dollyredirect

Is containerized:

No, it is a Unity3D application.

Installation and configuration

- Download and install the Weart middleware
- Download and install the Oculus Application
- Connect the haptic gloves doggle to your computer
- Open the Weart middleware application and connect to the haptic gloves
- Connect the Oculus headset to your computer
- Wear the haptic gloves and the Oculus headset
- Run the provided application (.exe)

Produced data (Outputs)

- Haptic effects (thermal and force queues)
- Text descriptions
- Audio descriptions

Consumed data. (Inputs)

Tracking data from the device

Asynchronous data exchange

N/A

APIs

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

N/A

Third-party libraries used

- Libraries in Weart SDK package for Unity
- XR Interaction Toolkit https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@2.4/manual/index.html
- UAP https://assetstore.unity.com/packages/tools/gui/ui-accessibility-plugin-uap-87935

User interface

Virtual Reality application

How to test the system

Run the provided Unity application and interact with the virtual environment and check for:

- Success rate of at least 85% for navigating the virtual environment and interacting with virtual artifacts for users with visual impairments
- Attain a high user satisfaction score (SUS), M>=75%
- Achieve a positive overall user experience (UEQ), M>=80%

5.8.11. M9-FEX CULTURAL ASSET PRE-PROCESSING AND FEATURE EXTRACTION FOR MEDIA CURATION

Provider: UAU

Work Package: WP4/T4.1: Cultural Asset Pre-processing and Feature Extraction for Media Curation.

Part of tools: T2-VS Video to Speech, T5-CT Contemporary Translation, T6-AF Accessibility Framework, T7-AT Accessible.

Short description of the final module:

A wide range of features exists, and the first step taken was to collect, group, and decide which features might be most interesting for further investigation. However, possible features differ depending on the modality of the asset (textual, visual, acoustic). We categorized the features into three fundamental types: factual, contextual, and subjective information.

For this first deliverable, we focused on two specific features: smell extraction and evoked emotion classification. We chose these, as they are so far less explored than other features and at the same time have the ability to add great value to the understanding of CH assets especially for visually impaired users.

Full description is part of D4.1(SEN) Chapter 2 (SHIFT-D4.1, 2023)

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

Technologies used

OS: Ubuntu, NixOS

Platform: Python 3, Poetry

Libraries: <not known yet>

Is containerized: no

Installation and configuration

e.g., pip requirements.txt

Produced data (Outputs)

Text

Consumed data. (Inputs)

Image, Text, Table

Asynchronous data exchange

N/A

APIs

<not specified yet>

Third-party libraries used

e.g., Hugging Face

User interface

Terminal

How to test the system

- Indirectly when testing other modules based on the models created.
- Acceptable UAR scores (defined depending on specific feature)
- Acceptable user satisfaction for each feature

5.8.12. M10-ASO MULTIMEDIA CULTURAL ASSET CURATION BASED ON ASSOCIATION BY DESIGN

Provider: UAU

Work Package: WP4/T4.2: Multimedia Cultural Asset Curation Based on Association by Design.

Part of tools: T1-IV Image to video, T6-AF Accessibility Framework, T7-AT Accessible.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

Short description of the final module:

This module will analyze, the features extracted in T4.1 to implement the methodology of association by design for establishing the correlation between cultural assets. More specifically, interlinking the cultural assets among themselves, but also with external information. To that end, a harvesting mechanism will be devised interoperating with a diversity of available APIs provided by open sources using the extracted features of T4.1 as arguments, to create multimedia cultural assets that are interlinked.

During the initial project phase, emphasis was placed on specifying ontological aspects that facilitate more effective interaction with users.

For the first period, we focused on tangible CH. We crafted a first ontology based on the data provided by the project partners. The created ontology is thought to enable users to specify a painting based on various features, including Title, Artist, Type, Style, Material, and Keywords. This initial version of the SHIFT ontology is designed to accommodate future expansions by seamlessly integrating new entities as they become available from the AI components of the platform.

Full description is part of D4.1(SEN) Chapter 3 (SHIFT-D4.1, 2023)

Technologies used

OS: Ubuntu, NixOS

Platform: Python 3, Poetry

Libraries: <not known yet>

Is containerized: no

Installation and configuration

e.g., pip requirements.txt

Produced data (Outputs)

Text

Consumed data. (Inputs)

Image, Text, Table

Asynchronous data exchange

N/A

APIs

<not known yet>

Third-party libraries used

e.g., Hugging Face

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

User interface

Terminal

How to test the system

- Indirectly when testing other modules.
- Knowledge graph fully defined
- Knowledge graph contains data from different partners
- Knowledge graph contains extracted features

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

5.8.13. M11-DRM DIGITAL RIGHT MANAGEMENT

Provider: QMUL

Work Package: WP4, T4.4: Digital Rights Management (DRM).

Short description of the final module:

The aim of the module is to enable traceability of the cultural assets for the protection of digital rights on newly generated digital content by CHI using SHIFT technologies. The implementation of the module will leverage on the existing international standards such as media value chain ontology and other well-established processes for implementing the traceability of the digital rights of content that are shared and exchanged among the stakeholders.

A full description will be part of D4.2 (SHIFT-D4.2, 2024)

Technologies used:

OS: Linux, Windows

Platform: Java, Python3

Languages: Java, Python, OWL, SWRL

Source code available: No

Is containerized

N/A.

Installation and configuration

Module installation requires the execution of Java/Python components and the associated traceability identifier of cultural assets.

Produced data (Outputs)

- For a given input cultural asset identifier, a list of users will be presented.

Data format: JSON.

Availability: when it is exposed by dependent modules.

Consumed data. (Inputs)

All the inputs which are considered by other modules.

Is asynchronous data exchange necessary:

Message queues will be established to trace the use of digital content.

APIs

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

N/A

Third-party libraries used

- Ontology schema repository
- Graph databases

User interface

- CLI (Java/Python script)
- Automated execution using Restful APIs

How to test the system

- The module evaluation will be carried out in compliance with the international standards procedures and protocols as recommended by ISO and other governing bodies.
- The usability aspects of the system will be validated by CHI representatives within the consortium.

5.8.14. M12-PBK MAIN PLATFORM, BACKEND

Provider: SIMAVI

Work Package: WP5, T5.1: End-to-end Platform Architecture, Specifications and Development lifecycle.

Part of tools: It is the backend hosting all the other tools.

Short description of the final module:

This module contains the main background elements used to offer basic functionalities for data storage, data processing, and data exchange between modules.

- Container management: Docker V19;
- Storage: Postgresql, MongoDB;
- Authentication: Oauth2 based on Keycloack;
- Asynchronous message processing: Kafka;
- Run time: Java Run Time 17
- API: REST API
- API documentation (OpenAPI) by Swagger

A full description will be part of D5.2 (SHIFT-D5.2, 2024)

Technologies used:

OS: Linux, Windows

Platform: JDK 18, Python 5

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

Languages: Java, Python

Source code available: Yes. Java, Python

Is containerized

Docker container.

Docker file available

100 MB image size based on Alpine.

Installation and configuration

Copy image file

Docker load <image file> tag <image name>

Docker run -d <image name>

Produced data (Outputs)

- List and description of foreground objects.
- List and description of background objects

Data format: JSON.

Availability: when it is exposed by dependent modules.

Consumed data. (Inputs)

All the inputs which are considered by other modules.

Is asynchronous data exchange necessary:

Yes, for uploading images.

APIs

GET fglist

GET bklist

GET object/{id}

GET objecttype/{id}

Third-party libraries used

Libraries: Spring, Kafka,

User interface

CLI is text-based, with commands calling the APIs.

Web-based (HTML, JS, Angular).

How to test the system

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

Use Postman to send API requests.

The tests conducted will be based on test scenarios and test cases defined in task T5.3.

The tests will be considered as passed if all calls to APIs will return code 200 as result code, and expected results defined in tests cases will be obtained.

5.8.15. M13-PFB MAIN PLATFORM CLIENT (WALLET)

Provider: SIMAVI

Work Package: WP5, T5.1: End-to-end Platform Architecture, Specifications and Development lifecycle.

Part of tools: T1-IV Image to video, T2-VS Video to Speech.

Short description of the final module:

This module contains the main foreground functionalities able to present the final users where the cultural assets are located, to display partial or complete representation of the assets, and to guide the usage of special devices (haptic for example) when necessary.

The module will be a web based UI, targeted to desktop/laptop or mobile platforms.

The UI will be targeted to 5 user roles.(**EU1-CH** professional , **EU2-VI** Visually impaired user, **EU3-BL** Blind user, **EU4-HI** Hearing impaired, **EU5-Admin** Administrator)

A full description will be part of D5.2 (SHIFT-D5.2, 2024)

Technologies used:

OS: Linux, Windows, IOS, Android

Platform: JDK 18, Python 5

Languages: Java, Python, HTML5, JavaScript

Source code available: Yes. Java, Python

Is containerized: No

Installation and configuration

Installation from App Store, Google Store.

Produced data (Outputs)

- Representations on the display, or other output device

Availability: when it is exposed by dependent modules.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

Consumed data. (Inputs)

All the inputs are considered by other modules.

Is asynchronous data exchange necessary:

Yes, for uploading images.

APIs

GET fglist

GET bklist

GET object/{id}

GET objecttype/{id}

Third-party libraries used

Libraries: Spring, Angular

User interface

CLI is text-based, with commands calling the APIs.

Web-based (HTML, JS, Angular).

How to test the system

Manual testing based on test cases.

The success criteria considered will have two components:

In the browser, placed in development mode, in the console there will be no errors

The user will be presented with the results defined in test cases.

5.8.16. M14-COM COMMUNICATION AND INTEGRATION PLATFORM

Provider: SIMAVI

Work Package: WP5, T5.2: Integration of the system components into the SHIFT platform.

Part of tools: T1-IV Image to video, T2-VS Video to Speech.

Short description of the final module:

This module contains the main background elements used to offer basic functionalities for data exchange between modules. Both synchronous and asynchronous communication is considered.

- Asynchronous processing based on KAFKA
- Synchronous data exchange based on REST API.

A full description will be part of D5.2 (SHIFT-D5.2, 2024)

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

Technologies used:

OS: Linux, Windows

Platform: JDK 18, Python 5

Languages: Java, Python

Source code available: Yes. Java, Python

Is containerized

Docker container.

Docker file available

100 MB image size based on Alpine.

Installation and configuration

Copy image file

Docker load <image file> tag <image name>

Docker run -d <image name>

Produced data (Outputs)

JSON format for data requested.

Availability: when it is exposed by dependent modules.

Consumed data. (Inputs)

All the inputs are considered by other modules.

Is asynchronous data exchange necessary:

Yes, for uploading images.

APIs

GET fglist

GET bklist

GET object/{id}

GET objecttype/{id}

Third-party libraries used

Libraries: Spring, Kafka

User interface

None

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

How to test the system

Use Postman to send API requests.

The tests will check if any command returns 200 as the result code.

5.8.17. MAPPING BETWEEN TOOLS AND MODULES

In the previous chapter, we have described the available technical modules (Table 9 SHIFT List of Modules by Owner.). We are now presenting how the modules are mapped on the tools required by each use case.

In the next table, each row represents a TOOL (T1, T7).

Each column numbered from 1 to 10 represents a Module listed in Table 7(Table 9 SHIFT List of Modules by Owner.) and described in the chapter Physical View (5.8) For example, 1 represents Module M1-FOS Foreground/background object segmentation, 2 represents Module M2-MOS Physics Informed Machine Learning Algorithms and Video Generation, .. 10 represents Module M10-ASO Multimedia Cultural Asset Curation Based on Association by Design.

Modules M11-DRM Digital Rights Management, M12-PBK Main Platform, backend, M13-PFB Main platform client (Wallet), and M14-COM Communication and Integration Platform are general usage modules, offering support for the other functional modules, and are not placed in the table.

Table 10 SHIFT Tools – Modules mapping

Code	Description	1	2	3	4	5	6	7	8	9	10
T1-IV Image to video	Tool to enhance Photos / Paintings to Short Videos	x	x	x							x
T2-VS Video to Speech	Audio tool capable of interpreting visual stimuli (e.g., actions explained in visual sequences)			x			x			x	
T3-HA Haptic Interaction	A tool that translates physical objects to digital objects and uses haptics to "feel" the objects. To implement haptic interaction with 3D digital tangible and intangible cultural heritage assets, augmenting the user experience (UX) with new interaction paradigms that can be used in situ or remotely	x						x			

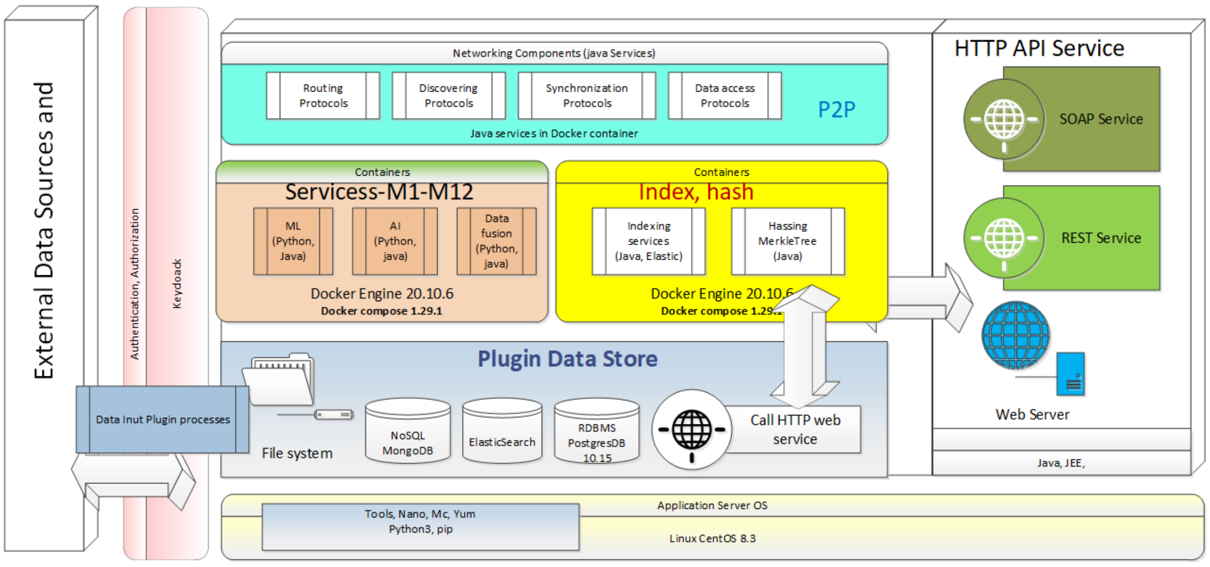
Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

T4-AN Audio Narrative	The tool automatically can provide complementary information regarding the cultural heritage assets (books, paintings, photos)				x	x	x				
T5-CT Contemporary Translation	A tool that translates historical meaning into more contemporary language and for auto-tagging/ auto-categorization of cultural heritage resources.				x	x				x	x
T6-AF Accessibility Framework	Comprehensive intuitive and accessible tool for all (including individuals with disabilities) multimodal storytelling of cultural heritage assets.			x	x	x	x		x	x	x
T7-AT Accessible Text-to-Speech	Comprehensive intuitive and accessible tool for all (including individuals with disabilities) multimodal storytelling of cultural heritage assets.				x	x	x			x	x

Modules are placed in the layered architecture in the Services block (Services M1-M12).

Each layer contains the implementations of the concepts and is explained in this subchapter.



Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

Figure 19. Technology stack, full node

5.8.18. NODES

The nodes can have the following 5 components:

- Data Input Plugin. It is a tool elaborated especially for each site and is used to get data from the site and present it to the Platform.
- Plugin DataStore. It has a storage system (File system RDBMS, or NOSQL), An access to the storage via secured REST API
- Index Hash. It is a summary of data available on the site and in all the sites having nodes linked in the P2P network
- Services. They are transformation and presentation services, developed in the project and identified in the list of modules as M1-M12.
- P2P Network Component. It is a software implementing communication protocols specific to Peer To Peer.
- HTTP Web services (API). They are REST or SOAP services used to expose the data.

The components mandatory for each node are:

- Index Hash
- Services
- P2P Network Component
- HTTP Web services (API)

That means that we can have a node not linked to an existing system, but able to expose data from other nodes, via the P2P link.

A Full node is created in the following way:

Step 1. Install the Docker engine.

Step 2. Download the Seed Node docker image.

Step 3. Create the Data Storage (manually install MongoDB, or create a file system).

Step 4. Install data Access plugins. (manually, based on the individual installation procedures).

Step 5. Install the procedures used to access data (M1-M12). .(manually, based on the individual installation procedures)

Step 6. Place the node in P2P. The address of another node should be known. Using this address, the network handshake procedure will be launched. The procedure will publish the newly created node in the P2P network, and update the node lists.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

Step 7. Synchronize nodes. The procedure will copy on the newly created node all indexes and hash contents (All simplified nodes).

Step 8. Create workspaces.

5.8.19. SEED NODE.

It is an important component used to generate a node. It contains all the mechanisms used by a node. It can be used to initiate a full node, a simplified node, or a knowledge node.

The Seed node is stored in a repository in the format of a Docker image.

The docker image is based on Ubuntu and contains the following preinstalled elements:

- Java SDK 18
- Apache Tomee
- Postgres DB
- PostgREST
- Index Hash mechanism
- P2P Network Component
- HTTP Web services (API)
- Initial Block (hardcoded initial bloc, defined by an initial hash).

5.8.20. DATA STORAGE

Data Storage can be added to Full nodes. The data storage proposed is based on a File system, or RDBMS (PostgreSQL) or NoSQL (MongoDB), and can store structured and unstructured heterogeneous information.

Access to the Data Storage is possible using the specific plugins for specific sources of data.

5.8.21. P2P

From an implementation perspective, the P2P network is composed of several equal nodes, all connected, each node is a package of services with the following functionality:

- Connection mechanism: used to link the node itself to the network.
- Discovery mechanism: used to find all the nodes in the network.
- List of all nodes.
- Communication protocol.
- Synchronization protocol.
- Data access protocol.
- Verification protocol.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

The architectural pattern implemented is Microservices.

5.8.22. P2P DATA MESH

It is the data structure defined by several full nodes placed in a P2P network.

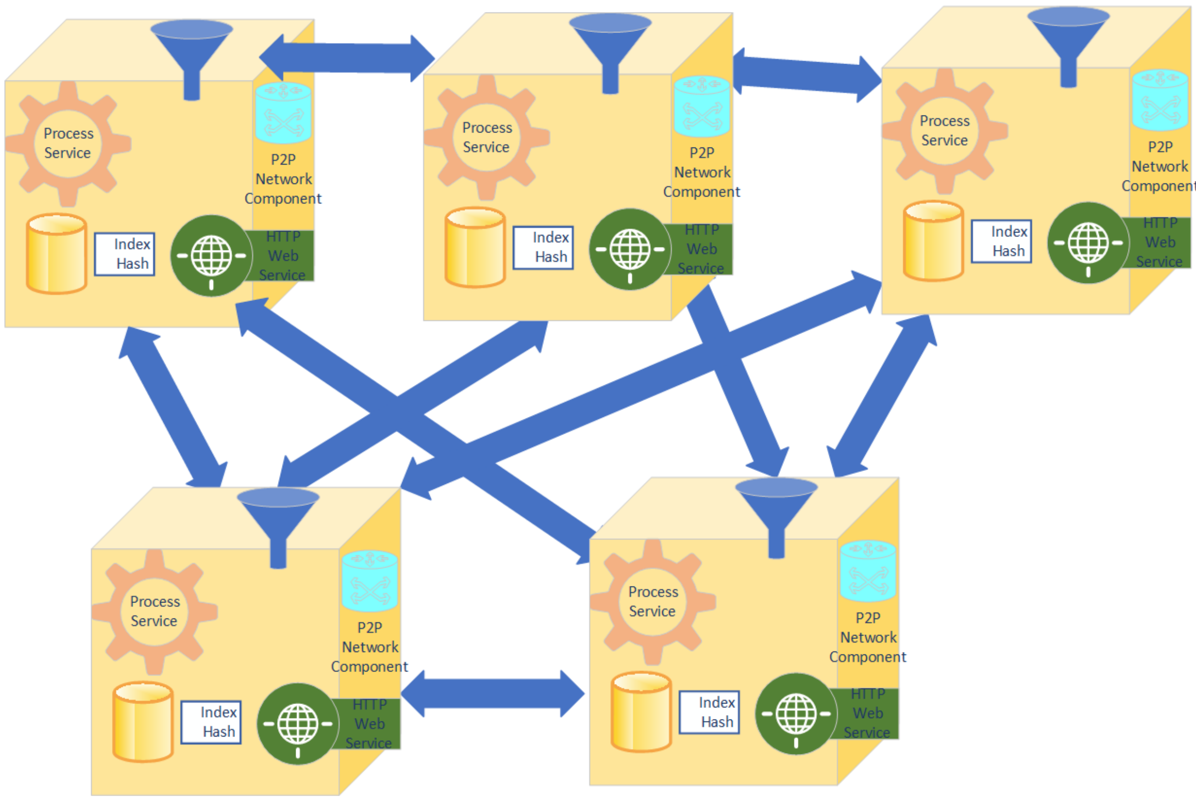


Figure 20. P2P data mesh

5.8.23. API

The APIs are exposing functionality for external clients to fully manage the Open repository.

We are grouping the APIs into the following categories:

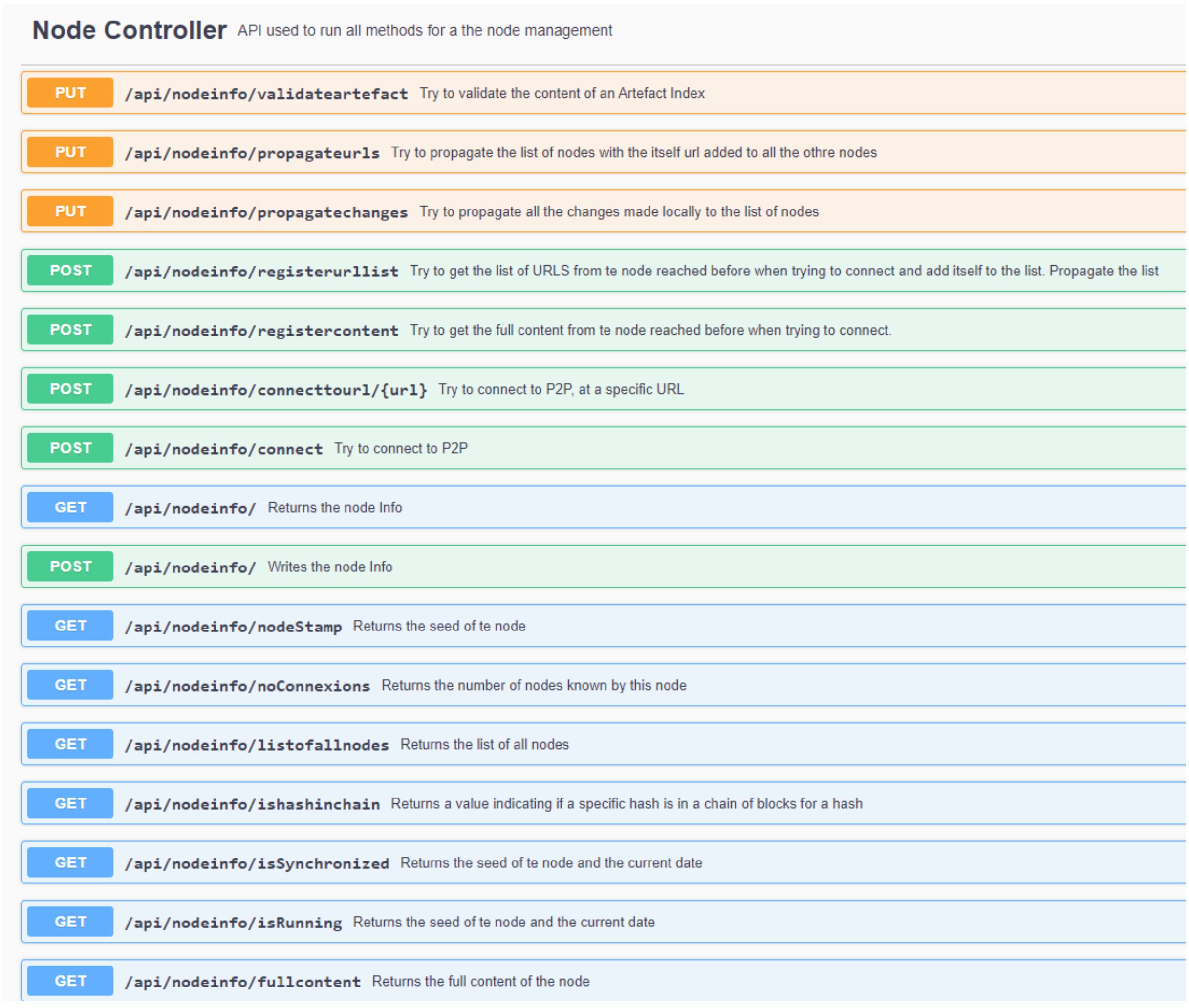
1. Node manipulation
   a. Add node to the network
   b. Node Configuration
   c. Synchronize nodes
   d. Detach a node
2. Data and knowledge storage and archiving:
   a. Add data
   b. Archive data
3. Data and knowledge retrieval and discovery:
   a. Search for data
   b. Get the data
   c. Verify data validity

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

4. Data process
    a. Call one or more of the M1-M12 modules
5. Statistics
    a. Get the volume of data
    b. Get the volume of usage
    c. Get frequency of usage
6. Logs
    a. Get logs for a period

An example of API is presented in the next figure (Figure 21. Node API):



Node Controller API used to run all methods for a the node management

PUT	/api/nodeinfo/validateartefact	Try to validate the content of an Artefact Index
PUT	/api/nodeinfo/propagateurls	Try to propagate the list of nodes with the itself url added to all the othre nodes
PUT	/api/nodeinfo/propagatechanges	Try to propagate all the changes made locally to the list of nodes
POST	/api/nodeinfo/registerurllist	Try to get the list of URLS from te node reached before when trying to connect and add itself to the list. Propagate the list
POST	/api/nodeinfo/registercontent	Try to get the full content from te node reached before when trying to connect.
POST	/api/nodeinfo/connecttourl/{url}	Try to connect to P2P, at a specific URL
POST	/api/nodeinfo/connect	Try to connect to P2P
GET	/api/nodeinfo/	Returns the node Info
POST	/api/nodeinfo/	Writes the node Info
GET	/api/nodeinfo/nodeStamp	Returns the seed of te node
GET	/api/nodeinfo/noConnexions	Returns the number of nodes known by this node
GET	/api/nodeinfo/listofallnodes	Returns the list of all nodes
GET	/api/nodeinfo/ishashinchain	Returns a value indicating if a specific hash is in a chain of blocks for a hash
GET	/api/nodeinfo/isSynchronized	Returns the seed of te node and the current date
GET	/api/nodeinfo/isRunning	Returns the seed of te node and the current date
GET	/api/nodeinfo/fullcontent	Returns the full content of the node

Figure 21. Node API

5.9. Development View

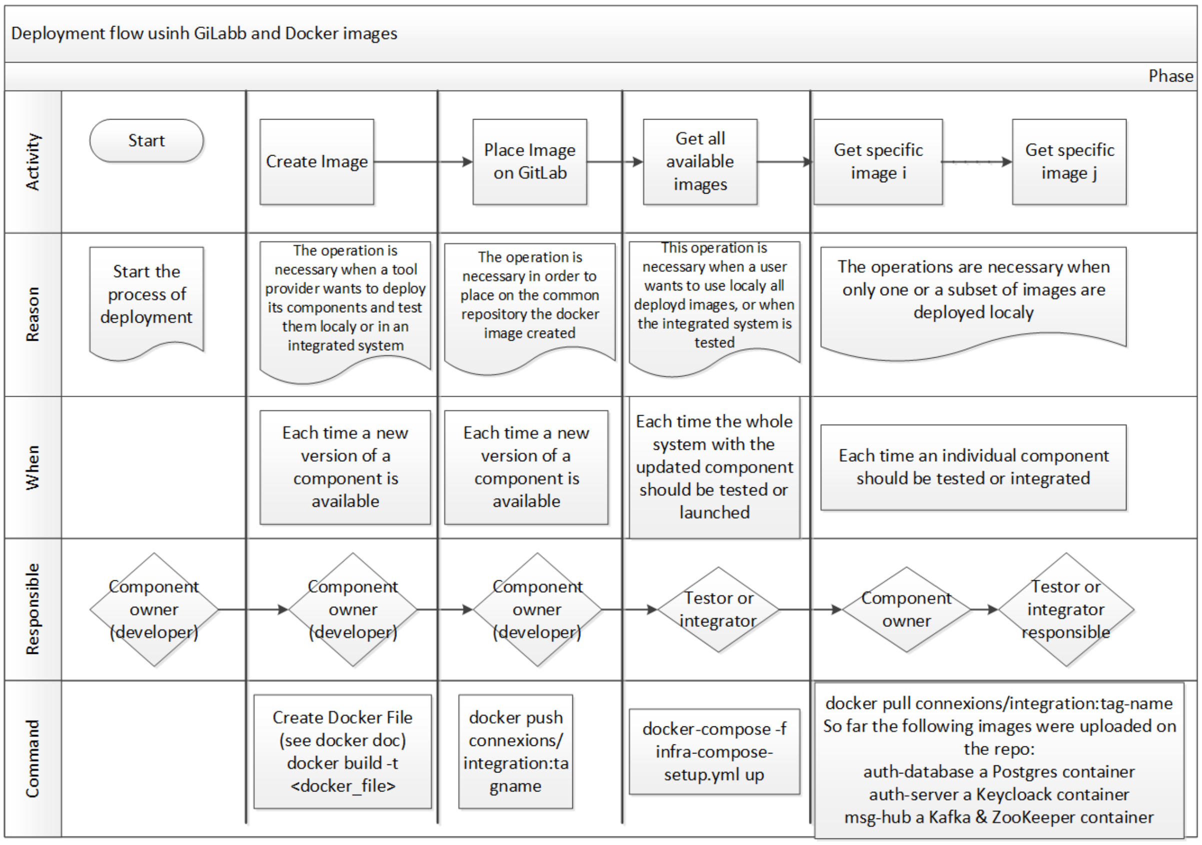
We are presenting in this chapter the development view of the architecture. We are referring to the technologies and tools used, and to the methodologies used for development and deployment.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

5.9.1. CI/CD

The development and deployment are governed by the Continuous Integration/Continuous Deployment paradigm. The description of the process, with responsibilities and schedule is presented in the next figure (Figure 22. CI/CD)



Activity	Start	Create Image	Place Image on GitLab	Get all available images	Get specific image i	Get specific image j
Reason	Start the process of deployment	The operation is necessary when a tool provider wants to deploy its components and test them localy or in an integrated system	The operation is necessary in order to place on the common repository the docker image created	This operation is necessary when a user wants to use localy all deployd images, or when the integrated system is tested	The operations are necessary when only one or a subset of images are deployed localy	
When		Each time a new version of a component is available	Each time a new version of a component is available	Each time the whole system with the updated component should be tested or launched	Each time an individual component should be tested or integrated	
Responsible	Component owner (developer)	Component owner (developer)	Component owner (developer)	Testor or integrator	Component owner	Testor or integrator responsible
Command		Create Docker File (see docker doc) docker build -t <docker_file>	docker push connexions/ integration:tagname	docker-compose -f infra-compose-setup.yml up	docker pull connexions/integration:tag-name So far the following images were uploaded on the repo: auth-database a Postgres container auth-server a Keycloack container msg-hub a Kafka & ZooKeeper container	

Figure 22. CI/CD

In detail, the steps are presented in the next figure (Figure 23. Development and deployment flow)

1.1 Enter the code (edit) using an IDE (IDEA, NetBeans, PYChaRM, Visual Studio)
1.2 Place the code in the GitLab repository (Commit)
1.3 Merge changes
1.4 Build the process (maven, Gradle, npm)
1.5 Run scripts
1.6 Inspect the code (IDEA, Sonar Cube)

2. Take components from a repository (maven, GitLab, Apache)

3. Take the plugins if available (database drivers, libraries)

4. Place and edit configuration files

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

5. Unit tests
6. Place the results/issues in Jira (Issue management system)
7. Link and call available web services
8. Build the whole system (call Gitlab pipeline)
9. Run integration tests
10. Deploy one node
11. Run the node on the pilot site

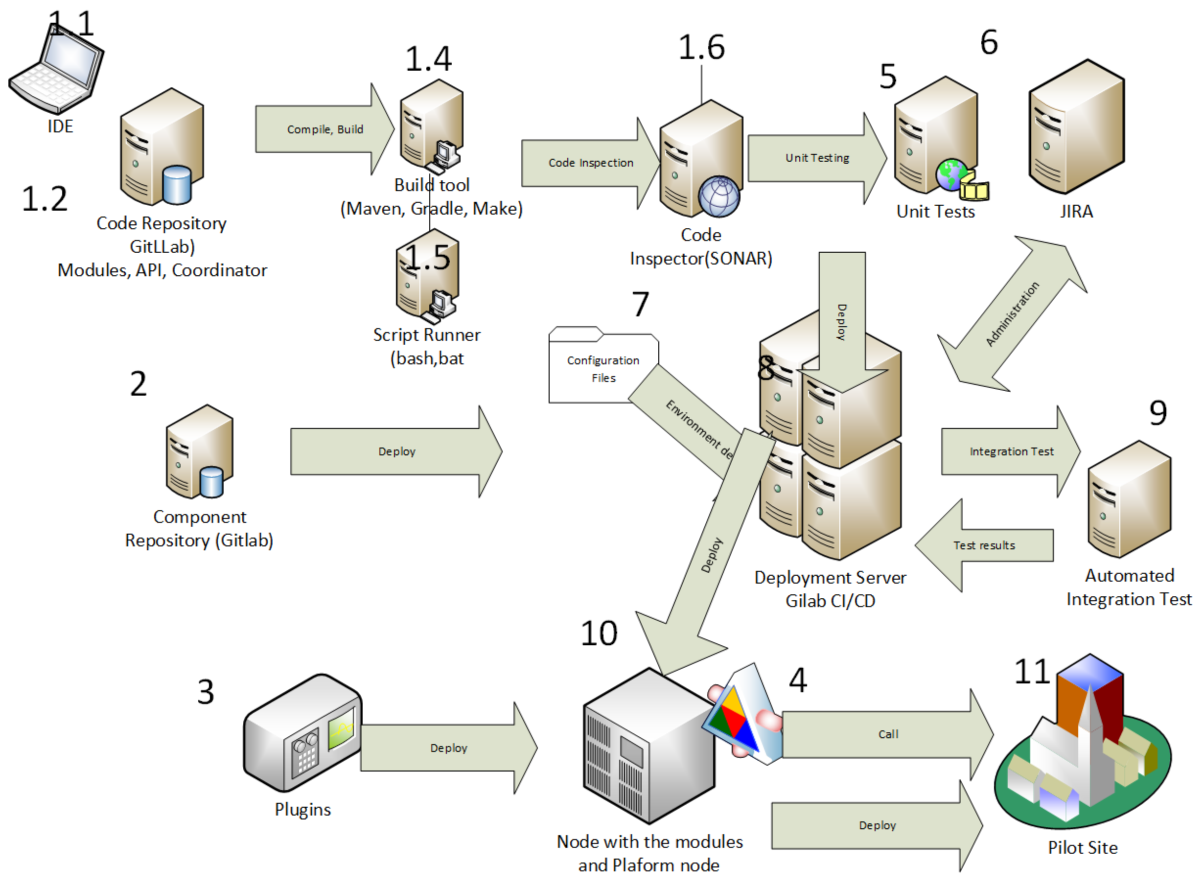


Figure 23. Development and deployment flow

5.9.2.TECHNOLOGIES

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

We are mentioning here the technologies used, with a short description of their role and functionality. For details of each technology used, we are indicating the references.

5.9.2.1. P2P

"The term peer-to-peer, or P2P, means that the computers that participate in the network are peers to each other, that they are all equal, that there are no "special" nodes, and that all nodes share the burden of providing network services. The network nodes interconnect in a mesh network with a "flat" topology. There is no server, no centralized service, and no hierarchy within the network. Nodes in a P2P network both provide and consume services at the same time with reciprocity acting as the incentive for participation. P2P networks are inherently resilient, decentralized, and open. A preeminent example of a P2P network architecture was the early internet itself, where nodes on the IP network were equal. Today's Internet architecture is more hierarchical, but the Internet Protocol still retains its flat-topology essence." (Antonopoulos, 2017)

Network discovery

When a new node boots up, it must discover other P2P nodes on the network to participate. To start this process, a new node must discover at least one existing node on the network and connect to it.

The geographic location of other nodes is irrelevant; the network topology is not geographically defined. Therefore, any existing P2P nodes can be selected at random.

To connect to a known peer, nodes establish a TCP connection, usually to port 68444 (the port generally known as the one used by SHIFT), or an alternative port if one is provided. Upon establishing a connection, the node will start a "handshake" by transmitting a version message, which contains basic identifying information.

5.9.2.2. SUMMARY DATA

Summary data is the normalization, indexing, and hashing of existing data in a site, created in such a manner that, it offers quick data retrieval, and presents uniformly the data to the processes implemented in modules M1-M12.

Summary data is created for bunches of real data no larger than an established amount (for example 10 MB). If real data is larger then it is split into chunks.

5.9.2.3. BLOCKS

A block is the full representation of one version of a data summary from a site.

The block is made of a header, containing metadata, followed by a list of summary data versions that make up the bulk of its size. The block header is 80 bytes,

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

whereas the average summary data is at least 256 bytes, and the average block is expected to contain about 10 summary data.

The proposed structure of a block is:

- 4 bytes Block Size The size of the block, in bytes, following this field.
- 80 bytes Block Header Several fields form the block header (see below).
- 1-9 bytes (VarInt) Transaction Counter How many workspace versions follow.
- Variable Workspaces The versions of workspaces recorded in this block.

5.9.2.4. BLOCK HEADERS (ANTONOPOULOS, 2017)

The block header consists of the following information:

- 4 bytes Version A version number to track software/protocol upgrades.
- 32 bytes Previous Block Hash A reference to the hash of the previous (parent) block in the chain.
- 32 bytes Merkle Root A hash of the root of the Merkle tree of this block's workspaces.
- 4 bytes Timestamp The approximate creation time of this block (seconds from Unix Epoch).
- 8 bytes. Reserved

5.9.2.5. BLOCK IDENTIFIERS - BLOCK HEADER HASH AND BLOCK HEIGHT

The primary identifier of a block is its cryptographic hash, a digital fingerprint, made by hashing the block header twice through the SHA256 algorithm.

The resulting 32-byte hash is called the *block hash*, but is more accurately the *block* ***header*** *hash*, as only the block header is used to compute it.

The block hash identifies a block uniquely and unambiguously and can be independently derived by any node by simply hashing the block header.

Note that the block hash is not actually included inside the block's data structure, neither when the block is transmitted on the network, nor when it is stored on a node's persistence storage as part of the node. Instead, the block's hash is computed by each node as the block is received from the network. The block hash may be stored in a separate database table as part of the block's metadata, to facilitate indexing and faster retrieval of blocks from disk.

A second way to identify a block is by its position in the blockchain, called the *block height*. The first block ever created is at block height 0 (zero)

5.9.2.6. MERKLE TREES

"A *Merkle tree*, also known as a *binary hash tree*, is a data structure used for efficiently summarizing and verifying the integrity of large sets of data. Merkle trees are binary trees containing cryptographic hashes.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

The term "tree" is used in computer science to describe a branching data structure, but these trees are usually displayed upside down with the "root" at the top and the "leaves" at the bottom of a diagram." (Antonopoulos, 2017)

5.9.2.7. BLOOM FILTERS (ANTONOPOULOS, 2017)

A bloom filter is a probabilistic search filter, a way to describe a desired pattern without specifying it exactly. Bloom filters offer an efficient way to express a search pattern while protecting privacy. They are used to asking their peers for data matching a specific pattern, without revealing exactly which addresses they are searching for.

5.9.2.8. NODES

A node implements all the elements necessary for the Platform to manage the summary data on all sites, to call the processes implemented by modules (M1-m12), and to manage the links to raw data on the sites. It can work in standalone mode too. In other words, the minimum implementation of the Platform is one node where there is summary information, and links to data on the site and also to the output of the processes (M1-M12) implemented.

The five elements of a full node are:

- Data Store. Is the place where all data is stored. Typically, it is a Mongo DB database, with multimodal content.
- Index and hash. Is a file structure containing all the blocks created on the data from the data store. It is a tree structure of blocks.
- Processes. It is a group of modules, which can be run on the data from the data Store. The implementation resides in modules M1-M12.
- P2P component. It is a microservice used to search for, discover, and link to other nodes.
- API. It is a microservice exposing functionality for Open repository clients. It is REST API.

5.9.2.9. INITIAL NODE

It is a node used to initialize any node in the system. It is part of the Seed Node. The content is hardcoded and contains:

- 4 bytes Block Size The size of the block, in bytes, following this field. Hardcoded 128).
- 80 bytes Block Header Several fields form the block header (see below).
- 2 bytes (VarInt) value 0.
- The predefined hash of the seed block (32 bytes)
- 14 bytes Reserved.

The Block header has the content:

- 4 bytes Version A version value 01.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

- 32 bytes Previous Block Hash A reference to the hash of the previous (parent) block in the chain. It has the value 0 (no previous block)
- 32 bytes Merkle Root A hash of the root of the Merkle tree of this block's workspaces. The 32 bytes of the predefined seed node hash.
- 4 bytes Timestamp The approximate creation time of this block (seconds from Unix Epoch).
- 8 bytes. Reserved

5.9.2.10. P2P COMPONENT OF A NODE.

This component is mandatory for all nodes. It has two main elements:

- The list and addresses of all the other nodes in the P2P network. It is used when nodes are added or removed, or when data synchronization is necessary.
- A microservice is used to search for and discover data and knowledge.

5.9.2.11. PLUGINS

Plugins are used to access data from heterogeneous systems and to make them available in the processes implemented in a node.

A plugin will be developed for each site and is specific to the site.

5.9.3. TOOLS

In the development phase, we will use the following tools and technologies, as presented in the next diagram (Figure 24. Technology stack, development)

- Operating systems:
  - Linux Ubuntu 20 for development server
  - Windows 11 for local development environments
- Languages:
  - Java 18
  - Python 3
- Databases
  - PostgreSQL
  - MongoDB
- Indexing and search mechanism
  - ElasticStack
- Web servers:
  - Apache Tomcat, Apache Tomee
- Container management
  - Docker
  - Docker compose
- Communication protocols
  - HTTP, HTTPS
- Services

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

- Web Soap
- Web REST

- Authentication, Authorization, Identity, and Access management
  - Keycloack
- IDEA
  - IntelliJ IDEA
  - Apache Netbeans
  - PyCharm
  - Visual Studio
  - Jupiter
- Libraries, frameworks
  - Spring
  - Springboot

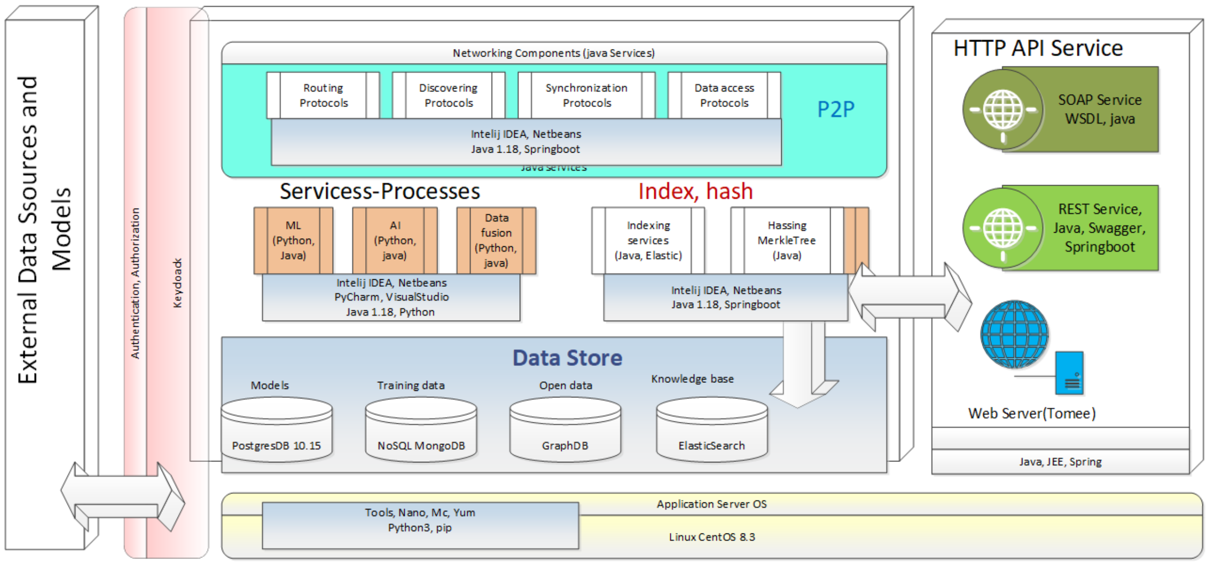

External Data Ssources and Models
Authentication, Authorization
Keydoack
Networking Components (java Services)
Routing Protocols
Discovering Protocols
Synchronization Protocols
Data access Protocols
P2P
Intelij IDEA, Netbeans Java 1.18, Springboot
Java services
Servicess-Processes
ML (Python, Java)
AI (Python, java)
Data fusion (Python, java)
Intelij IDEA, Netbeans PyCharm, VisualStudio Java 1.18, Python
Index, hash
Indexing services (Java, Elastic)
Hassing MerkleTree (Java)
Intelij IDEA, Netbeans Java 1.18, Springboot
Data Store
Models
Training data
Open data
Knowledge base
PostgresDB 10.15
NoSQL MongoDB
GraphDB
ElasticSearch
HTTP API Service
SOAP Service WSDL, java
REST Service, Java, Swagger, Springboot
Web Server(Tomee)
Java, JEE, Spring
Application Server OS
Tools, Nano, Mc, Yum Python3, pip
Linux CentOS 8.3

Figure 24. Technology stack, development

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

6. Conclusion

After an introductory part considering the methodologies used, we have presented in the document the project's technical requirements and the proposed End-To-End Platform Architecture.

The system is described in terms of 7 tools implementing the functionalities of 10 specific and 4 general business modules. The tools, the modules, and the map tools/modules or tools/use-cases are described.

The End-To-End Platform Architecture is based on the Data Mesh concept defined in Chapter 4.2. and to be implemented in a distributed system around a Peer-to-peer (P2P) network.

The system will place business modules on a backend platform. The main physical elements of the Platform backend are Nodes. A node offers data storage, retrieval, validation, collaboration, governance, and distribution functionality. Nodes can be added or removed at any moment from the system and also can work in standalone mode. Each node contains the business main entities grouped in workspaces.

This is the basis for the development of an entire ecosystem that offers cultural heritage institutions the necessary impetus to stimulate growth and embrace the latest technical innovations.

All the data or knowledge is exposed via APIs, presented at the level of each node.

Some of the modules offer themselves specific data representations (for example 3D, haptic, and video interfaces).

The architecture of the whole system is defined to allow the fulfillment of the objectives of the project and validate this on the use cases defined.

The document will be used as input for the development of the whole system, especially the tools in WP3 and WP4, and the integration mechanisms in WP5.

Possible changes in the technical requirements or system architecture, after the interaction with the other WPs, will be reflected in D5.2 – Integration and functional testing, D5.3 Pilots evaluation strategy plan, and D5.4 Pilots final report.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

References

Antonopoulos, A. M. (2017). *Mastering Bitcoin. - Programming the Open Blockchain.* O’Reilly Media, Inc.

Data-Mesh-General. (2023). Retrieved from Data-Mesh: https://martinfowler.com/articles/data-mesh-principles.html

Dehghani, Z. (2022). *Data Mesh: Delivering Data-Driven Value at Scale.* O'Reilly Media.

Jacek Majchrzak, S. B. (2022). *Data Mesh in Action.* Manning .

Kleppmann, M. (2017). *Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems.* O'Reilly Media.

Kruchten, P. (1995). Architectural Blueprints—The “4+1” View. *IEEE Software 12 (6)*, 42-50.

MatCHMaker. (2023). *D5.1 Technical specifications of the Open Repository.* HORIZON-CL4-2022-RESILIENCE-01-19.

MES-CoBraD. (2021). *D7.1 System requirements & architecture.* MES-CoBraD project.

Nick Rozanski, E. W. (2005). *Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives.* Addison-Wesley Professional.

Per Kroll, P. K. (2003). *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP: A Practitioner's Guide to the RUP.* Addison-Wesley Professional.

Richards, M. (2015). *Software Architecture Patterns.* O'Reilly Media, Inc.

SHIFT Consortium. (2022). DOA-SHIFT. Bruxelles.

SHIFT-D1.1. (2023). *D1.1 - SHIFT requirements, user evaluation.*

SHIFT-D2.1. (2023). *Automatic generation of motion sequences.* SHIFT Consortia.

SHIFT-D3.1. (2023). *Tool for the textual representation of CH.* SHIFT Consortiu.

SHIFT-D3.2. (2023). *Text and video to affective speech synthesis.* SHIFT Consortia.

SHIFT-D3.3. (2023). *Haptic based interaction with CH assets.* SHIFT Consortia.

SHIFT-D3.4. (2024). *Accessible framework of inclusive museum.* SHIFT Consortia.

SHIFT-D4.1. (2023). *Tools for Cultural Asset Curation and.* SHIFT Consortia.

SHIFT-D4.2. (2024). *Distribution SHIFT Curation Repository for Cultural Assets with DRM capabilities.* SHIFT Consortia.

SHIFT-D5.2. (2024). *Integration and functional testing.* SHIFT Consortia.

SIMAVI. (2021). QA-Procedures and Work Instructions (In Romanian). Bucharest, Romania.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.

The Members of the SHIFT Consortium:

Organizations	Country	Role
SIMAVI - SOFTWARE IMAGINATION & VISION	Romania	Coordinator
FORTH - IDRYMA TECHNOLOGIAS KAI EREVNAS	Greece	Partner
MDS - MASSIVE DYNAMIC SWEDEN AB	Sweden	Partner
AUD - audEERING GmbH	Germany	Partner
UAU - UNIVERSITAET AUGSBURG	Germany	Partner
SOMKL - MAGYAR NEMZETI MÚZEUM – SEMMELWEIS ORVOSTÖRTÉNETI MÚZEUM	Hungary	Partner
ANBPR - THE NATIONAL ASSOCIATION OF LIBRARIANS AND PUBLIC LIBRARIES IN ROMANIA	Romania	Partner
SPK - STIFTUNG PREUSSISCHER KULTURBESITZ	Germany	Partner
BMN - THE BALKAN MUSEUM NETWORK	Bosnia and Herzegovina	Partner
HERITAGE - HERITAGE MANAGEMENT	Greece	Partner
ERC - ETICAS RESEARCH AND CONSULTING	Spain	Partner
DBSV - GERMAN FEDERATION OF THE BLIND AND PARTIALLY SIGHTED	Germany	Partner
QMUL - QUEEN MARY UNIVERSITY OF LONDON	United Kingdom	Associated Partner

Contact:

Project Coordinator: **Purcarea Razvan** **SIMAVI**- SOFTWARE IMAGINATION & VISION	razvan.purcarea@simavi.ro

Disclaimer:

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 101060660.