

**Collectie van
de Gentenaar**

D7.3.3 - 07.3.1

Visualisation layer: Version 2.0 of the CoGhent software

Project Name: Collections of Ghent

Funding programme: Urban Innovative Actions Call 2019

Contract Number: UIA05-249

Work Package: 7

Contributors: Leen Van Campe, Pieter-Jan Pauwels, Olivier D'Huynslager, Jeroen Budts



This project is funded by the European funding for regional development through Urban Innovative Actions (UIA)

1. Glossary	2
2. Introduction	3
2.1 Contextual information	3
3. Software architecture	3
4. Source code	5
4.1 Back-office	5
4.1.1 LDES	5
4.1.2 DAMS	6
4.2 Front-end interfaces	6
4.2.1. DAMS Front-end	6
4.2.2 Web platform	7
4.2.3 The story wall (180° wall)	8
5. Contact details	9
6. Appendix	9

1. Glossary

Adlib: Adlib is a Collection Management System (CMS) designed for recording and managing museum collection data, used by most Ghent cultural heritage institutions.

CoGent Box: The CoGent-box is a digital experience space that travels through Ghent's neighbourhoods. It revives heritage objects by means of local narratives. Visitors can discover, experience and share their past with other residents of Ghent. From April 2022 to February 2023 the CoGent-box made a voyage in three different districts in Ghent: Wondelgem, Watersportbaan-Ekkerghem and Sluizeken-Tolhuis-Ham.

DAMS: A Digital Asset Management System (DAMS) is a software solution or a set of tools designed to centrally store, organise, manage, retrieve, and distribute digital assets. Digital assets can include a wide range of media files such as images, videos, audio files, documents, design files, and more.

IIIF: IIIF, which stands for the International Image Interoperability Framework, is a set of standardised specifications and best practices for sharing and presenting images and other media on the web. IIIF is primarily used in the cultural heritage and academic sectors to provide easy access to high-quality images and related metadata for a wide range of digital objects, including manuscripts, artworks, photographs, maps, and more. Its goal is to improve the interoperability, accessibility, and reuse of digital images and media.

LDES: Collections of Ghent published its data using Linked Data Event Streams. A Linked Data Event Stream is a collection of immutable objects (such as versioned objects or archived representations) and can be updated anytime at their own pace (slow and fast data). Each object is described in RDF and made accessible via one URI endpoint. This way, consumers can easily replicate the history of a dataset and efficiently harvest the latest changes.

Microservice: A microservice is a software architectural design pattern in which an application is composed of loosely coupled, independently deployable, and independently scalable components or services. These services are small, focused, and typically responsible for a single business capability or function. Microservices are often used to build complex applications by breaking them down into smaller, manageable parts.

2. Introduction

The CoGent Box as an immersive space (O7.2.1) doesn't come to life without the extensive software components that drive the different interfaces, both in the box and online. With this document we want to enable other cities, cultural heritage institutions and interested parties to see how it all connects and can possibly be reused to create similar interactions in their respective environment.

Rather than extensively describing the different components, this document will mainly link to the documentation pages online as well as the source code to replicate or adapt.

The source code has been opened under open source licences by the partners. Components maintained by the City of Ghent / District09 and / or developed by IDLab-imec-UGent are opened up under the European [EUPL License](#). And the source code released by Inuits has been opened up by the internationally recognized [GNU GPLv2 License](#).

2.1 Contextual information

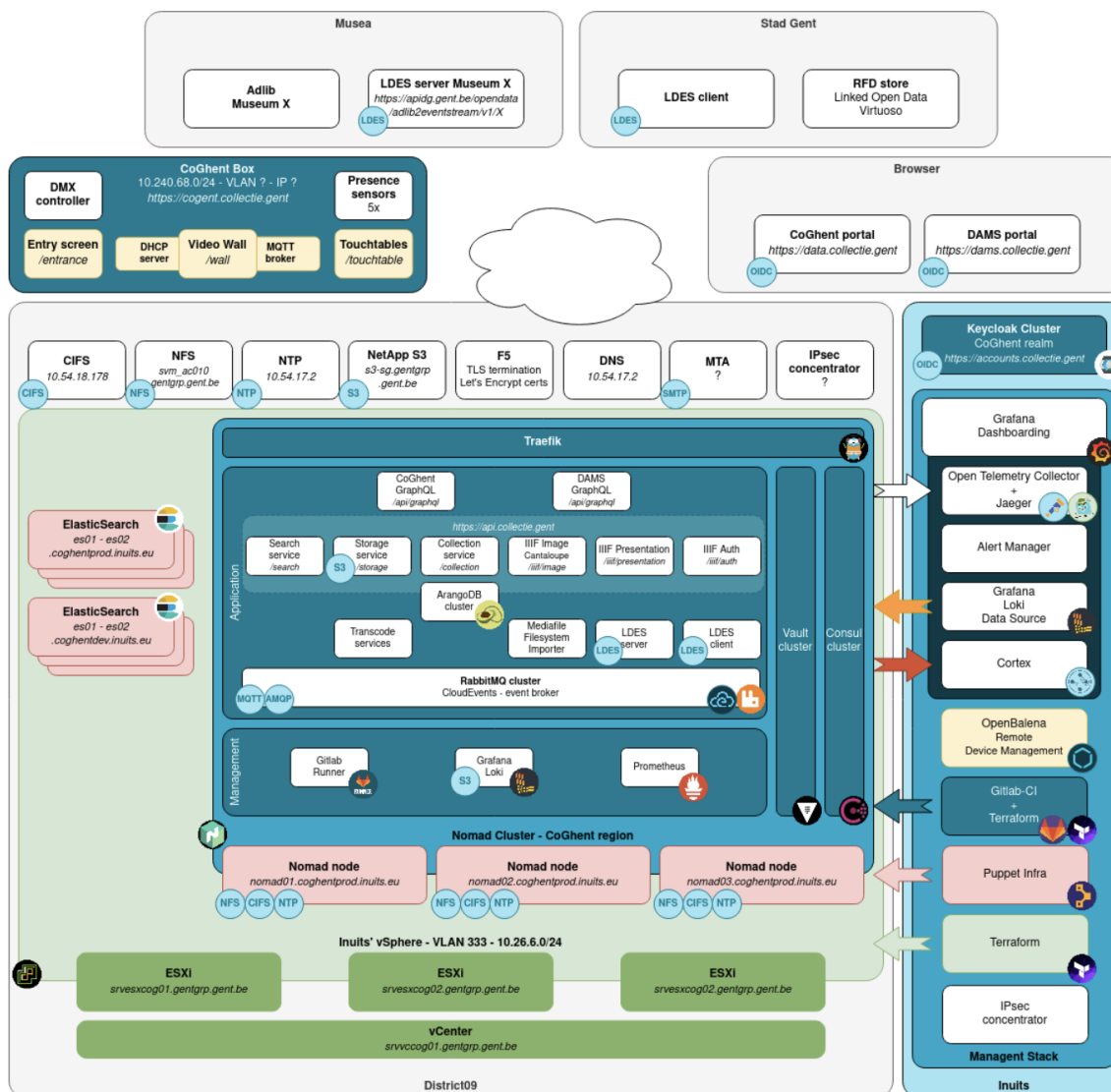
Want to know more about the how and why of this project before you dive in?

Watch the [explainer video](#) to get a general feel about the project. There is a separate [video](#) showing how the CoGent box looked and how the interfaces work. You can also read O7.2.1 to have some more insights in the overall setup of the CoGent box, about the interior, exterior and the design of the different interfaces.

How data is being managed in the project is described in the [data management plan](#). The Data Management Plan describes the lifecycle of the data used and generated within the CoGent project. It sets the framework for the management of the data, from acquisition to curation to dissemination to preservation.

3. Software architecture

The architecture shows how the different components within the project interconnect. A more detailed view can be downloaded separately on the online results page of CoGhent. See image below.



The main grey area on the bottom left entails the complete DAMS system, this is the heart of the CoGhent project where the metadata and assets come together from the different institutions to be fed in the different systems and interfaces.

This DAMS system is set up as a combination of microservices and is interconnected with the management stack that is managed by Inuits, the main IT-partners within the project. You can see this as the blue column on the right. As this management stack is very organisation specific, it is out-of-scope for this document.

This DAMS system gets its metadata feed from the cultural heritage institutions in the upper left corner, which is needed to link with the images. This metadata comes from their respective collection management systems Adlib through the Adlib2eventstream service.

This allows the DAMS to bring everything together and feed the data and assets to both the CoGent box, the blue box on the left, to show the heritage and local narratives to citizens on-site. And it also feeds the web platform data.collectie.gent to show the collection to the citizens online, which is portrayed under the greybox named 'browser on the right of the CoGent Box.

A LDES client consumes the metadata of the different LDES streams and feeds these to the RDF store (Virtuoso) of the City of Ghent. Thus the data become discoverable on the sparql endpoint <https://stad.gent/sparql/>. An example query can be found on the open data portal <https://data.stad.gent/explore/dataset/coghent-nova-objecten/information>.

4. Source code

As the software in CoGhent is set up as a combination of microservices and APIs, there are different repositories that contain the source code for these different applications. We'll divide them into back-office applications and front-end interfaces.

4.1 Back-office

To open up digitised cultural heritage on a city level, there are two big back-office systems at play. The LDES (Linked Data Events streams) and the DAMS (Digital Asset management system).

4.1.1 LDES

The adlib2eventstream API exposes the Adlib databases as event streams and is composed of 2 open-source components:

1. Adlib backend: First, an Adlib database is retrieved, and the data is mapped to Linked Data in JSON-LD format before being stored in a database.
2. Event stream API: Subsequently, the Linked Data fragments are accessed from this database following the Eventstream specification.

The documentation of the API's and the source code itself can be found on the GitHub of the City of Ghent. All applications are available as Open Source repositories.

Documentation of the API's: <https://github.com/StadGent/CoGhent.github.io> and the swagger file on <https://apidg.gent.be/opendata/adlib2eventstream/v1/api-docs/>

The NodeJS applications that support the adlib2eventstreamproces

Eventstream API: https://github.com/StadGent/node_service_eventstream-api

The objective of a Linked Data Event Stream is to allow consumers to replicate all of its items and to stay in sync when items are added, updated or deleted. A Linked Data Event

Stream is a collection of immutable objects (such as version objects, sensor observations or archived representation). Each object is described in RDF.

Adlib Backend: https://github.com/StadGent/node_service_adlib-backend

The adlib-backend extracts, transforms and loads from an Adlib API into a database. This database is then used by [eventstream-api](#) that exposes [event streams](#)*.

4.1.2 DAMS

The different DAMS consists of different microservices that are all gathered on this Gitlab: <https://gitlab.com/inuits/dams>

It contains 12 repositories:

Collection API

This is the core of the DAMS system. The Collection API contains all metadata of all assets, entities, and taxonomies.

Storage API

The Storage API is responsible for writing and making individual media files available. For this purpose, the S3 server of District09 is used within CoGhent.

Search API & Elastic Indexer service

The Search API indexes all assets and entities and updates the index when changes occur.

Transcode service

Automatically, one or more transcode services generate derivatives (transcodes). An example of this is TIFFs automatically converted to JPGs to better suit a web environment.

Graphql Service

An intermediary layer between the management interface and the various services. The GraphQL layer ensures that the management interface doesn't have to connect directly to all the different services but only needs to communicate with GraphQL.

Csv-import-service & Antivirus are the services

The service allows us to import various media files to the DAMS system.

Cantaloupe IIIF image API

The IIIF image API provides a standardised way to request and deliver images. This can be as simple as, give me the original image to give me an upside tiled version of the image in a gif format. The API is restful and allows for images to be served dynamically.

IIIF Manifest Mapper

The IIIF manifest contains the metadata about the image object. This library is being used for creating these manifests.

LDES Consumer

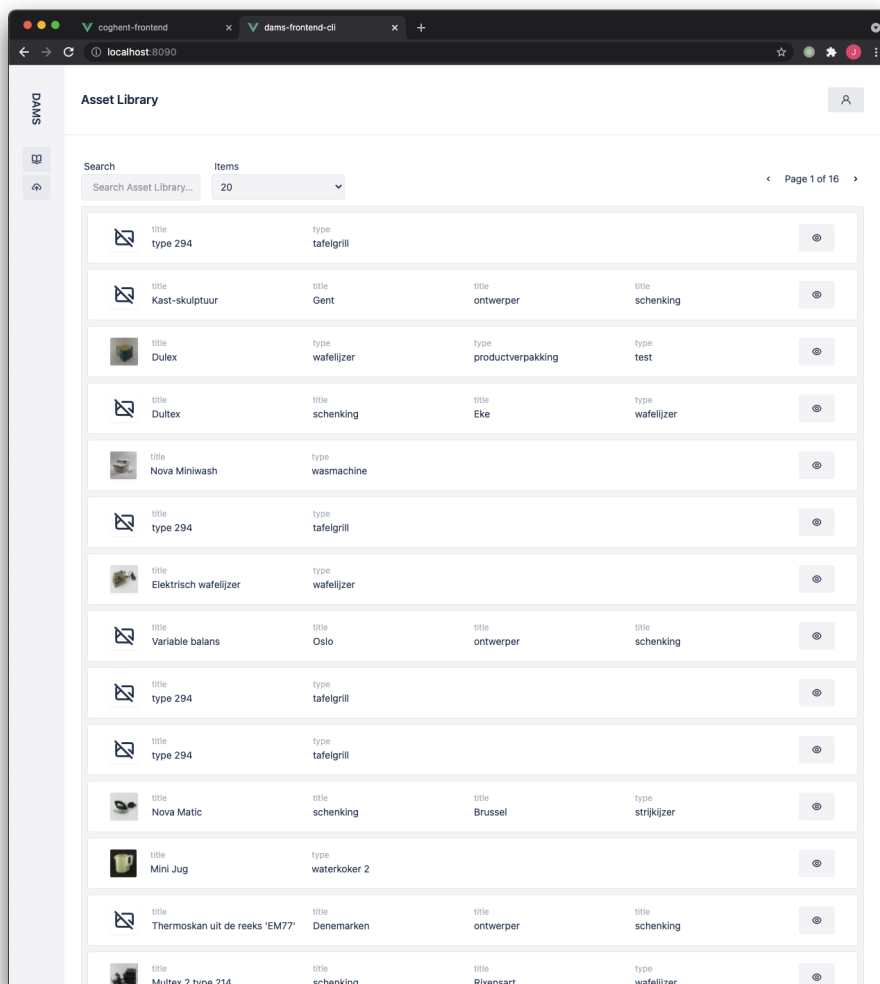
The LDES consumer feeds the different LDES streams into the RDF store of the City of Ghent (Virtuoso) and thus opens up the Coghent metadata to a public sparql endpoint.

4.2 Front-end interfaces

4.2.1. DAMS Front-end

This front-end aids employees of institutions and services in managing digital materials. This is achieved by ensuring that media files can be imported in the simplest and most efficient way possible. On the other hand, it also involves automatically and comprehensively retrieving associated metadata from source systems and establishing the connection between metadata and visual content.

<https://gitlab.com/inuits/dams/dams-frontend>



Asset overview in the DAMS Front-end

4.2.2 Web platform

The web platform is an online platform for citizens to browse and search through the different cultural heritage collections. You can view this online webplatform on <https://data.collectie.gent/>



Frontpage view of the web platform

Components for the web platform:

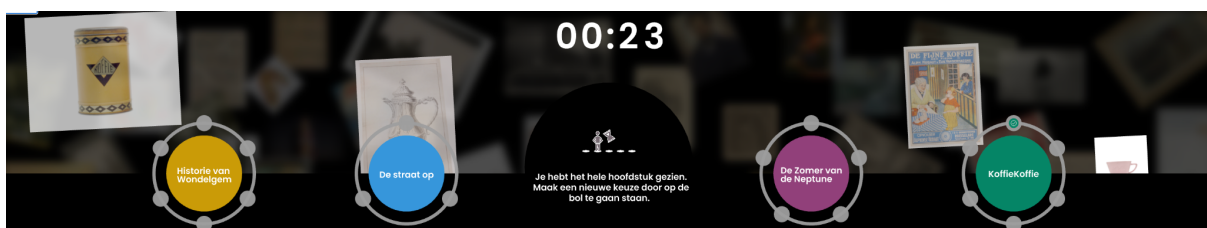
The main source code is to be found on https://github.com/StadGent/vue_site_coghent-web-portal

The front-end component library (in Vue) is available on https://github.com/StadGent/vue_webcomponent_coghent-component-library

The library for OpenID Connect (authentication of users) is available on: https://github.com/StadGent/vue_webcomponent_coghent-oidc-library

4.2.3 The story wall (180° wall)

The Coghent 180° wall revives heritage objects by means of local narratives. These narratives are fed with data from the DAMS which is retrieved via the Coghent GraphQL.



The chapter selection menu of the story wall interface after viewing a (first) chapter.

The source code is available on

https://github.com/StadGent/node_site_coghent-box-frontend

5. Contact details

Do you have any questions about the architecture or source code? Feel free to reach out via info@district09.gent. State your questions via e-mail and we'll make sure we connect you with the right technical partners.

6. Appendix

Full resolution architecture image, see below.

Architectuur / Infrastructuur

