unchain

D3.3 – Integration, open interfaces development and lab-testing

UK Research and Innovation

Funded by the European Union

CIVITAS Sustainable and smart mobility for all

Legal disclaimer

This document is issued within the frame and for the purpose of the UNCHAIN project. This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101103812. The UK participant in Horizon Europe Project UNCHAIN, is supported by UKRI grant number 10078841 Lancaster University.

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them.

Copyright statement

Deliverable details

Deliverable No.	D3.3
Deliverable Type	DEM – Demostrator, pilot, prototype
Dissemination Level	PU – Public
Work Package No.	WP3
Work Package Title	Data-driven urban logistics cooperation framework
Task No.	T3.2, T3.3 and T3.4
Task Title	Secure, standardised and interoperable urban logistics data. Smart logistics services marketplace architecture. Software integration, open interfaces and lab testing
Author(s)	Daniel Villalobos
Status (F:final; D:draft; RD:revised draft)	F
File name	D3.3 - Integration, open interfaces development and lab-testing
Version	1.0
Task start date and duration	M6-M18
Due Date	M18 (October 2024)
Delivery Date	31/10/2024

Review History of Deliverable

Version No.	Date	Revision Details
0.1	08/10/2024	Draft version
0.2	14/10/2024	Reviewed version
0.3	15/10/2024	Reviewed version
1.0	31/10/2024	Final version

Reviewers of Deliverable

Reviewers' name	Company	Date
Juan Giménez	IBV	14/10/2024 (v0.2)
Konstantinos Zografos	ULANC	15/10/2024 (v0.3)
Konstantinos Zografos	ULANC	30/10/2024(v1.0)

Table of Contents

List of Figures

Acronyms

Acronym	Meaning
API	Application Programming Interface
CA	Certificate Authority
DAPS	Dynamic Attribute Provisioning Service
DSC	DataSpace connector
EGI	European Grid Infrastructure
GUI	Graphical User Interface
IDS	Internacional Data Space
IDSA	Internacional Data Space Association
MDB	Metadata Broker
REST	Representational State Transfer
UI	User Interface
WP	Work Package

1. Executive Summary

The UNCHAIN project focuses on digitally transforming urban logistics to support sustainable and efficient urban mobility. It addresses the challenges of increased urban logistics demand and digitalization by promoting collaboration among local authorities, logistics stakeholders, and technology providers.

One of the key objectives of this collaboration is to establish a robust infrastructure for data sharing and service integration, contributing to a European urban freight data space.

The objective of this document is to elucidate and provide a comprehensive overview of the system developed in modules T3.2 and T3.3. This system is designed to automate the process of creating connectors and their subsequent implementation within the Data Space. By detailing the architecture, functionalities, and integration mechanisms, this document aims to offer stakeholders a clear understanding of the automated connector creation process and its role in enhancing the efficiency and scalability of the Data Space ecosystem.

The collaborative framework is based on the International Data Spaces Association (IDSA) reference architecture, ensuring secure, standardized, and efficient data transfers. A user-friendly graphical user interface (GUI) modelled after the IDSA interface, along with APIs documented on Swagger, enhances the usability and effectiveness of the data exchange ecosystem.

The standardised data exchange ecosystem and smart services marketplace architecture developed in the UNCHAIN project represent significant progress towards improving urban logistics and promoting sustainable urban mobility. The project's focus on standardization, interoperability, and user-friendly interfaces aims to break down data silos, facilitate better coordination, and improve logistics operations across urban environments. The "D3.3 - Integration, open interfaces development and lab-testing" deliverable advances UNCHAIN's urban logistics Data Space by automating participant onboarding and establishing secure, standardized data exchanges. The continuous integration (CI) system minimizes manual intervention, ensures rapid integration, and builds trust between stakeholders in compliance with IDS standards. This streamlined, secure infrastructure fosters efficient, data-driven urban logistics, optimizing operations and reducing environmental impact amidst rising urban freight demands.

1. Introduction

The UNCHAIN project is an Innovation Action targeting the digital transformation of urban logistics to support sustainable and efficient urban mobility. The project focuses on overcoming the challenges associated with the increasing demand for urban logistics and the digitalization of freight operations. By promoting a collaborative framework that involves local authorities, logistics stakeholders, and technology providers, UNCHAIN aims to establish a

robust infrastructure for data sharing and service integration, ultimately contributing to the creation of a European urban freight data space.

Currently, urban logistics faces significant challenges, including traffic congestion, environmental impact, and the inefficiency of freight operations. The increasing urbanization and demand for quick deliveries have exacerbated these issues. These inefficiencies also have a social impact, as citizens are exposed to increased noise, pollution, accidents, and aesthetic degradation. Therefore, improving freight operations is not only crucial for the logistics sector but also benefits the broader community. Digital transformation in logistics is essential to address these challenges by optimizing routes, reducing emissions, and improving the overall efficiency of freight operations.

1.1. Purpose of this document

- D3.3 aims to build upon the foundations laid in D3.2 within Work Package 3 (WP3) of the UNCHAIN project. The primary focus of this deliverable is to further enhance the interoperability and trust between stakeholders in the urban logistics ecosystem by automating the integration of new participants into the Data Space. The key tasks completed for D3.3 include:
- **Creation of a Continuous Integration (CI) System for Participant Inclusion:** A significant achievement of D3.3 is the development of a CI system that facilitates the seamless onboarding of new participants into the Data Space. This system automates the configuration of services, ensuring that new connectors are rapidly deployed, fully operational, and compliant with the existing standards. The automation process reduces manual intervention and significantly accelerates the integration timeline**.**
- **Establishment of Trust Mechanisms for Data Exchange:** A critical component of the CI system is its ability to automatically establish trust between new connectors and existing participants. This feature ensures that new participants can exchange data securely and efficiently with others in the ecosystem. By simplifying the trust-building process, the created CI system enhances collaboration and data sharing, further breaking down the barriers between urban logistics stakeholders.

This deliverable plays a pivotal role in advancing the digitalization of urban logistics by enabling dynamic scalability and increased flexibility in participant management. Key outcomes of D3.3 include:

- **Automated Data Space Configuration:** By automating the onboarding process, the system developed enables a streamlined and error-free integration of new participants. This ensures that the Data Space can grow organically while maintaining compliance with International Data Spaces (IDS) standards. The automation of service configurations reduces the complexity of the integration process, allowing stakeholders to focus on value-added activities.
- **Enhanced Trust and Security in Data Exchange:** The automatic establishment of trust between participants ensures a secure and reliable exchange of data, fostering greater collaboration within the logistics ecosystem. The system's ability to validate and

authenticate new connectors guarantees that only trusted entities participate in the data exchange, maintaining the integrity of the platform.

- **Scalable and Flexible Data Space Ecosystem:** The CI system provides the necessary infrastructure to support the scaling of the Data Space as more participants join. This ensures that the platform remains agile and adaptable to future growth, allowing cities and stakeholders to manage increasingly complex logistics networks with ease.

D3.3 not only enhances the technical infrastructure of the Data Space but also reinforces the project's commitment to foster a collaborative environment. By streamlining the integration process and promoting trust, this deliverable contributes to the broader objectives of creating sustainable, data-driven urban logistics solutions across Europe.

1.2. Relation with other project activities

Deliverable D3.2 comes as a result of the following tasks:

- **T3.2 - Secure, Standardised and Interoperable Urban Logistics Data:** This task focuses on defining and implementing the data exchange modules and connectors required for secure, standardized, and interoperable data sharing. The primary goal is to ensure that data from various sources can be shared and used efficiently across different platforms and stakeholders within the urban logistics ecosystem. This involves the development of software add-ons compatible with existing systems used by urban logistics operators and public administration smart platforms. The modules are designed to comply with IDS principles and reference architecture proposed by the International Data Spaces Association (IDSA). The outcome of this task is a set of data exchange solutions that provide a secure and standardized way of sharing urban logistics data, thereby enabling better coordination and optimization of logistics operations across cities.
- **T3.3 - Smart Logistics Services Marketplace Architecture:** The objective of this task is to design and implement the architecture of a smart logistics services marketplace in a user-friendly way to create the collaboration framework. This marketplace will integrate various smart logistics services developed within the UNCHAIN project and serve as a central hub for these services. The architecture is designed to be included as a new module within existing local administration platforms, such as multimodal mobility or smart city platforms. This integration will facilitate the management and visualization of logistics metrics and KPIs, thus enhancing the decision-making capabilities of urban planners and policymakers. The marketplace will support services developed in subsequent work packages (WP4 and WP5), providing a comprehensive platform for urban logistics management
- **T3.4 - Software Integration, Open Interfaces, and Lab Testing:** This task concentrated on the integration and testing of the standardized data exchange ecosystem and the smart city logistics services marketplace. In D3.3, extensive lab testing was conducted to ensure the seamless interaction between the various modules developed in T3.2 and T3.3. By employing web services and open interfaces, the marketplace was

validated for smooth integration with both public administration platforms and private logistics operators. The APIs and services were tested to ensure compliance with usability and acceptance criteria, and several validation stages were conducted with key stakeholders to confirm the user-friendliness of the system. The lab testing ensured that all components were fully operational, reliable, and ready for full deployment, providing a robust foundation for future integration in real-world urban logistics environments.

The combined outcomes of T3.2, T3.3, and T3.4 demonstrate the potential of the UNCHAIN project to create a scalable, secure, and interoperable urban logistics framework. The results exposed on the D3.3 represents a major step forward in enabling the automatic and secure integration of new participants into the Data Space, providing enhanced tools for urban planners and logistics operators to optimize their activities in real-time while ensuring the system remains compliant with IDSA standards.

1.3. Structure of the document

This document is structured as follows:

- Section 2 includes a common section to all the UNCHAIN deliverables presenting the gender, ethics and data related issues.
- Section 3 to 5 present the key contribution of this document, detailing the creation of the automated trust process for consortium members. It explains the functioning of the CI system, outlining its architecture and technical implementation, and highlights how it streamlines and automates the trust establishment process within the consortium. Moreover, a lab testing section is considered to validate that everything in the data space works properly.
- Section 6 list the main conclusions the deliverable.

2. Gender, ethics and data related issues

2.1. Gender related issues

Not applicable.

2.2. Ethics related issues

Not applicable.

2.3. Data related issues

In the context of T3.4 Software Integration, Open Interfaces, and Lab Testing, all data-related matters are documented in the *D3.2 Standardised Data Exchange Ecosystem and Smart Services Marketplace Architecture*. This document provides a comprehensive overview of the data standards, exchange protocols, and the architecture that supports secure and efficient data transactions in compliance with the fair data principles and the GDPR requirements stated in the data management plan (d1.2) within the project.

Specifically, for T3.4, a critical aspect of the system involves the management and handling of certificates, which are integral to the Continuous Integration (CI) system. The certificates used and shared in this context are exclusively public certificates, which are strictly employed for verification purposes. These certificates facilitate trust between the systems by ensuring the integrity and authenticity of the data being exchanged. It is essential to highlight that these certificates cannot be used for user identification, maintaining the privacy and security of participants within the system, and avoiding the impersonation.

Through the use of these public certificates, T3.4 ensures a secure environment where system components can communicate reliably, without compromising sensitive user information.

3. Connector complexity

The creation, deployment, and management of connectors within a Data Space is a technically intricate process that involves multiple steps and considerations. In various projects and implementations, significant efforts have been made to streamline the onboarding process for new connectors. However, this remains a challenging task, as it typically requires a central entity within the Data Space to oversee the creation, verification, and validation of the connector's credentials.

A key aspect of the complexity is ensuring the connector can securely and reliably communicate within the ecosystem. The connector must adhere to strict security protocols, such as encryption, authentication, and compliance with data governance policies. This ensures that all parties involved in the Data Space trust the connector's ability to handle data appropriately, whether in terms of sharing, accessing, or processing information.

The deployment of the connector presents another layer of complexity. Connectors can be deployed in various environments—ranging from local servers to cloud infrastructure or Kubernetes clusters. Each of these deployment options requires careful planning, configuration, and technical expertise. For example, in a Kubernetes deployment, managing containers, networking, and scaling demands specialized knowledge, often making the involvement of technical personnel unavoidable. Despite automation and orchestration tools, the process still demands human oversight to ensure the connector is properly integrated and functioning within the ecosystem.

Once the connector is operational, its typical interaction model is through an HTTP API, which adds a further technical requirement for users who need to engage with it programmatically. While this approach provides flexibility and scalability, it also necessitates a certain level of technical understanding. Our solution mitigates this challenge by providing a graphical user interface (GUI), designed to make the connector more accessible to users who may not be familiar with API-based interactions.

The deployment of the GUI is integrated with the connector itself. However, exposing this GUI for end-users requires additional steps. The interface must be hosted on a publicly accessible URL, and proper security measures must be implemented to ensure that only authorized users can access it. This involves configuring access control policies, securing the connection (e.g., using SSL/TLS certificates), and ensuring the interface is deployed in a manner that is both reliable and scalable.

The technical nature of connectors, coupled with the need for secure, reliable communication in a Data Space, means that some tasks still require significant expertise. Through the integration of user-friendly tools like a GUI and by refining the deployment processes, we aim to lower the barriers for technical users, though some level of complexity will always be part of this dynamic ecosystem.

4. Lab testing

The lab testing phase served as a comprehensive evaluation of the Continuous Integration (CI) system's ability to manage automated onboarding, secure integration, and data-sharing functions within the Data Space. Conducted across cloud servers and local computer environments, this testing aimed to replicate realistic usage scenarios by simulating multiple new participants attempting to join the Data Space simultaneously. This allowed the team to validate the CI system's reliability and robustness in handling concurrent onboarding requests.

A core focus of the testing process was to ensure that the CI system could facilitate a seamless and secure onboarding experience for new connectors, eliminating the need for manual intervention. By automating the configuration, validation, and deployment steps, the system aimed to reduce potential human error and optimize the integration timeline for new participants. The lab setup enabled close monitoring of each automated process, allowing developers to scrutinize the performance of various components, including certificate generation, trust establishment, and connector deployment.

Throughout the testing, the team encountered specific areas requiring improvement, which led to targeted refactoring within the CI system. The initial setup highlighted missing software requirements, unresolved code path errors, and several bugs in the automated scripts, particularly in handling edge cases. These issues were meticulously identified and addressed, resulting in refined script functionality, which were essential for ensuring robust, error-free onboarding in future deployments. This refactoring process underscored the importance of

anticipating edge cases and maintaining flexibility within the system to accommodate a range of potential integration scenarios.

The log system was instrumental in identifying and troubleshooting errors encountered during the lab testing. By capturing detailed logs at each stage of the integration process, the team was able to isolate issues quickly, analyze error patterns, and implement corrective actions.

Testing also validated the fundamental functionality of the connectors, ensuring that they could fulfil their intended role of secure data exchange within the Data Space ecosystem.

5. Continuous Integration System

5.1. Architecture

Data Space In the context of Task T3.4, a Continuous Integration (CI) system was developed to automate the integration of various components within the UNCHAIN project. A Continuous Integration (CI) system is a development practice that automates the process of integrating changes from multiple contributors into a shared repository. It involves automatically building every time a change is made, ensuring that new updates do not introduce errors or incompatibilities within the system.

In the case of the UNCHAIN project, the CI system plays a crucial role in automating the onboarding process for new participants joining the Data Space. This system ensures that new connectors can be seamlessly integrated into the ecosystem. The figure (Figure 1) shown below presents an image and the key steps involved in the CI process:

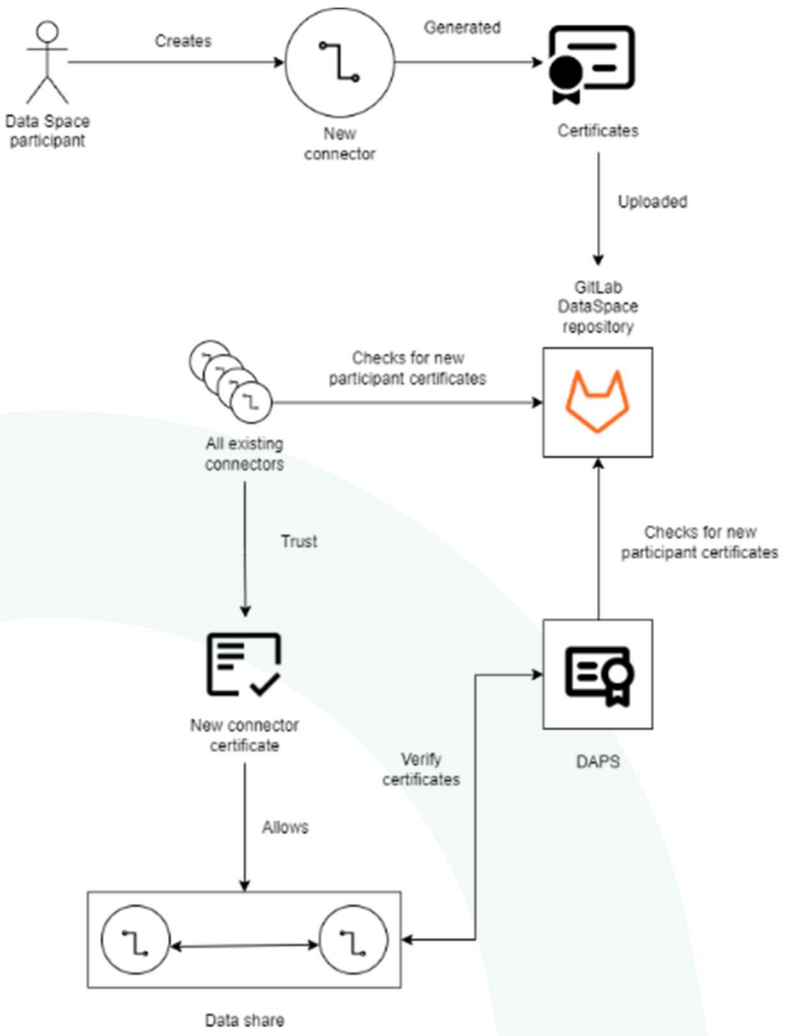

Figure 1 – CI system flow

1. **Private GitLab Repository for Connector Generation**: The CI system utilizes a private GitLab repository that contains a set of automated scripts designed to generate a new connector for a participant joining the Data Space. These scripts are responsible for creating and configuring the necessary components that will allow the new connector to interact with the system.
2. **Certificate Creation and Upload**: As part of the connector generation process, specific certificates are created to secure communications between the new connector and the existing infrastructure. Two of these certificates are uploaded to the GitLab repository. These certificates are essential for establishing trust between the new connector and other components within the Data Space, ensuring that all data exchanges are secure and compliant with the required standards.
3. **Watcher Scripts for Repository Monitoring**: To automate the process, we implemented watcher scripts that continuously monitor the GitLab repository for any changes, such as the addition of new certificates. When changes are detected, these

scripts automatically trigger a process to allow the existing connectors to trust the new one.

4. **Integration with the Dynamic Attribute Provisioning Service (DAPS) Service**: In addition to the watcher scripts, several services are deployed that specifically check for the creation of new connectors. These services handle the implementation of the new certificates into the Dynamic Attribute Provisioning Service (DAPS), a key component that manages access control and trust within the Data Space. Once the new connector's certificates are integrated and trusted by the DAPS service, it is recognized as a secure and compliant participant.
5. **Establishing Trust and Data Exchange**: After the new connector is successfully validated and both the DAPS and existing connectors trust it, the new participant can seamlessly engage in data sharing. The connector can act as both a data provider and a data consumer, participating in the Data Space's secure data exchange ecosystem alongside other trusted connectors.

5.2. How the CI system works?

5.2.1. Connector creation

The connector repository includes a set of scripts created ad-hoc to generate the necessary files to deploy a new connector.

The main script is called *setup_new_connector.sh*, a bash scripts that automates mostly everything in the process, asking the user for the necessary inputs to make everything works.

The necessary software to run the script is:

- jq
- yq
- cfssl
- openssl

Once the script is launched the terminal will show information about the current step of the process.

Figure 2 - Connector name input

The script will create a necessary JSON file that includes information about the new connector. This JSON file will be used to generate the necessary certificates.

Figure 3 - JSON file generated

Using the generated JSON file, the script will create the necessary certificates

```
}
Setting up a new PKI...
Reading the JSON file...
2024/09/17 11:17:30 [INFO] generate received request
2024/09/17 11:17:30 [INFO] received CSR
2024/09/17 11:17:30 [INFO] generating key: rsa-2048
2024/09/17 11:17:30 [INFO] encoded CSR
2024/09/17 11:17:30 [INFO] signed certificate with serial number 220289878739253388165458591623058648902562022545
PKI setup completed successfully.
```

Figure 4 - Generated certificates

After that, the script will generate a docker-compose file that will allow a technical partner to deploy and start to work with the connector. In addition, there'll be some files that are needed to run the connector and a keystore where all the other participants public certs will be allocated.

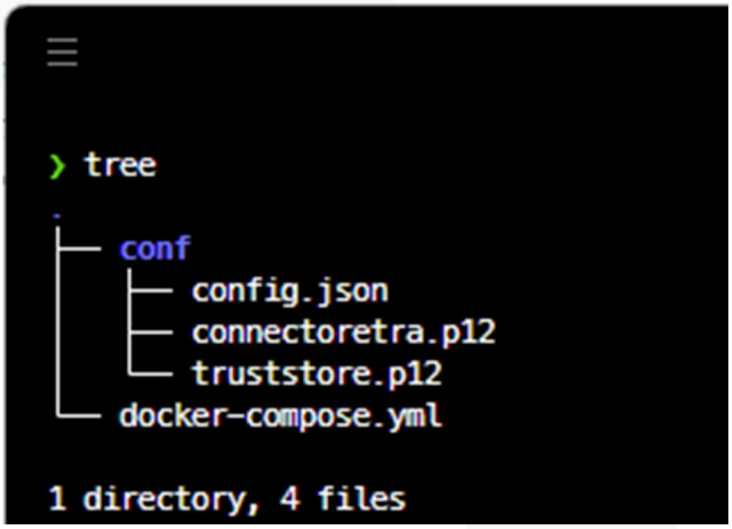

Figure 5 - Generated docker-compose and keystores

The Docker Compose file that is generated provides a straightforward way to deploy and begin using the connector. However, it is important to note that this is not the only deployment option available.

Docker Compose facilitates deployment by leveraging several Docker images to create a working environment, making it a convenient tool for local or small-scale deployments. For larger-scale or more complex environments, such as cloud infrastructure, the connector can also be deployed to a Kubernetes cluster.

This would require some modifications to adapt the Docker Compose setup to Kubernetes-specific configurations, such as defining Kubernetes services, pods, and persistent storage solutions.

With these adjustments, the connector can be effectively scaled and managed in a Kubernetes environment.

5.2.2. Watcher process

Along with the *setup_new_connector.sh* script, in the repository there's also two other scripts that should be used with the connector, *manage_connectors.sh* and *watcher.sh.*

Both of them are bash scripts to include the certificates from created connectors in the keychain of an existing one.

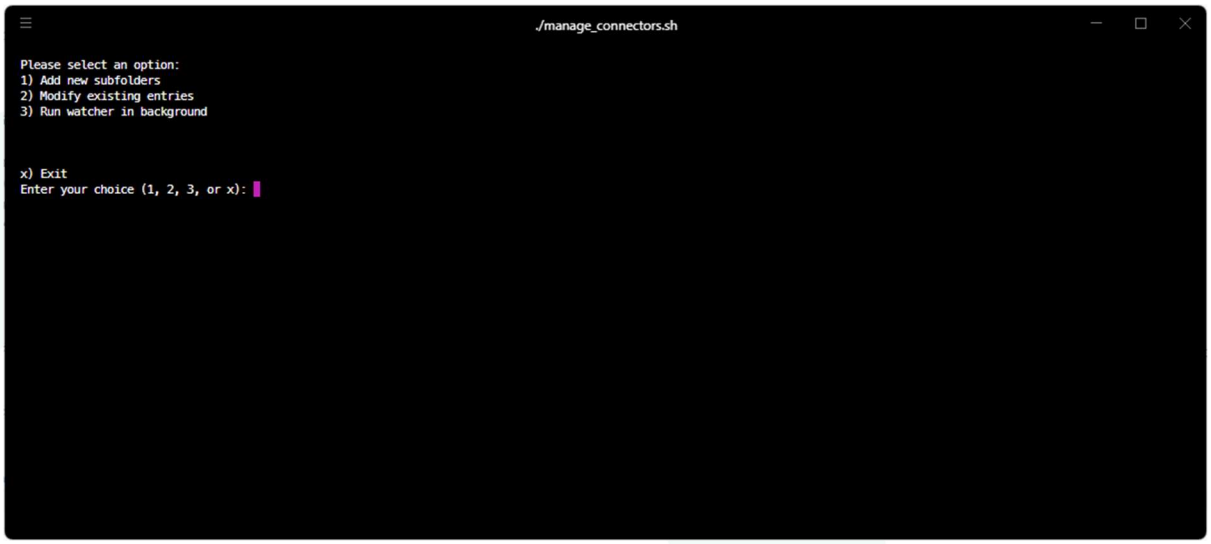

Figure 6 - Main manage connectors screen

The first script, *manage_connectors.sh*, is responsible for generating an SQLite database that keeps a record of all certificates that have already been added to the keychain. This database is used to ensure that no duplicate certificates are processed. In addition to tracking certificates, the script also creates a list of connectors that need to be monitored for updates.

Once a connector is selected for monitoring, the second script, *watcher.sh*, comes into play. This script is designed to regularly check the repository where the certificates are stored. It attempts to update the local repository by pulling any changes, and if new certificates are detected, or if the content of an existing certificate has been modified, it will automatically add them to the keystore.

The *watcher.sh* script performs these checks at regular intervals, updating the repository every 10 seconds to ensure that any new or modified certificates are quickly identified and processed. This ensures that the keychain remains up to date and that all connectors are continuously trusted.

5.2.3. DAPS CI

In addition to the connector repository, there is a separate repository dedicated to managing the deployed DAPS (Dynamic Attribute Provisioning Service).

The DAPS service plays a critical role in the Data Space by verifying whether a certificate is correctly signed and can be trusted.

To ensure that newly generated certificates are properly integrated and recognized within the Data Space, a comprehensive workflow has been established.

This workflow tracks each new certificate and ensures that it is validated and ready for use. The process ensures seamless trust management across the system, allowing connectors to operate securely.

The image below illustrates the details of this workflow.

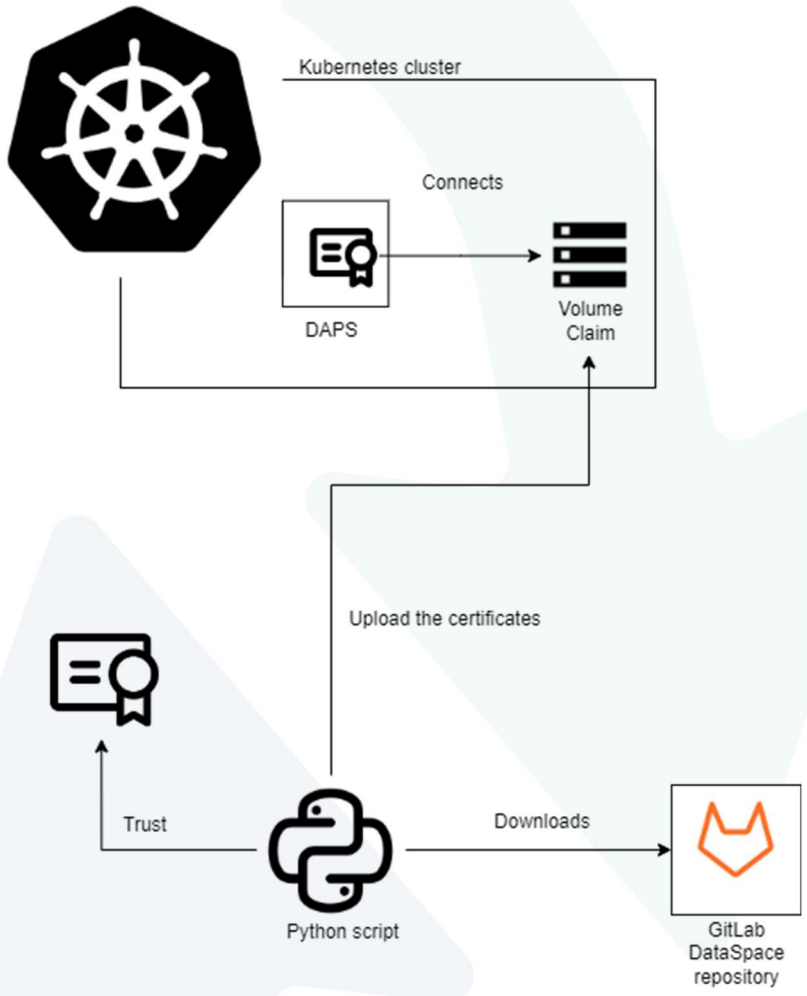

Figure 7 - DAPS workflow

The workflow begins with the Python scripts, which include an HTTP server featuring a secured webhook endpoint. This endpoint is protected by a secret key that must be passed as a parameter in the request header to ensure only authorized calls are accepted.

Once the webhook is triggered with the correct parameters, the script attempts to pull the latest changes from the connector's GitLab repository. If the repository update is successful, any new or modified certificates are added to the truststore of the DAPS, which is stored in a private repository to maintain security.

After the DAPS has successfully added the new certificates to its truststore, the certificates are uploaded to the Persistent Volume Claim (PVC) associated with the DAPS deployment in the Kubernetes cluster. Following this, the necessary services are restarted to ensure the new certificates are fully integrated and the DAPS can recognize and trust the updated connectors for secure data exchange.

6. Conclusions

The Continuous Integration (CI) system developed in the UNCHAIN project represents a key step forward in enabling the seamless and secure integration of new participants into a standardized urban logistics data exchange ecosystem. Through a combination of automation, secure certificate handling, and the deployment of trust mechanisms, this deliverable demonstrates how the project is tackling one of the central challenges in the digital transformation of urban logistics: fostering effective collaboration between diverse stakeholders while maintaining the integrity and security of the data being shared.

One of the key advantages of the CI system is its ability to fully automate the configuration of services for new participants entering the Data Space. In more traditional approaches, the process would require a central authority or figure to manually create the necessary certificates for each new participant, adding an extra step that increases complexity and time. This manual intervention introduces the possibility of delays and human error, as well as a potential bottleneck in the system. However, with the CI system's automated scripts and certificate management, the entire process is streamlined. Participants can be quickly and securely integrated without the need for manual oversight, as the responsibility to generate new connectors and certificates is distributed and automated. This approach not only eliminates additional steps but also enhances the efficiency of building trust between participants, enabling faster collaboration and data exchange. Furthermore, the system aligns with the International Data Spaces Association (IDSA) architecture, ensuring that all participants adhere to the highest standards of security and interoperability while retaining the flexibility to manage their own integration when needed.

The work presented in this deliverable not only provides technical solutions but also contributes to a broader strategic goal of the UNCHAIN project: fostering a sustainable, data-driven approach to urban logistics. By enabling real-time data sharing between local

authorities, logistics providers, and other stakeholders, the system provides valuable insights that can inform decision-making and lead to more efficient logistics planning and operations. This is particularly relevant in the context of modern urban challenges, where congestion, emissions, and inefficiency remain persistent issues. The ability to securely exchange data on freight operations allows cities to optimize routes, reduce environmental impact, and improve the overall flow of goods through urban areas.

As the system continues to evolve and scale, it will play an increasingly important role in shaping the future of urban logistics, helping cities meet the challenges of increased freight demand and environmental sustainability.