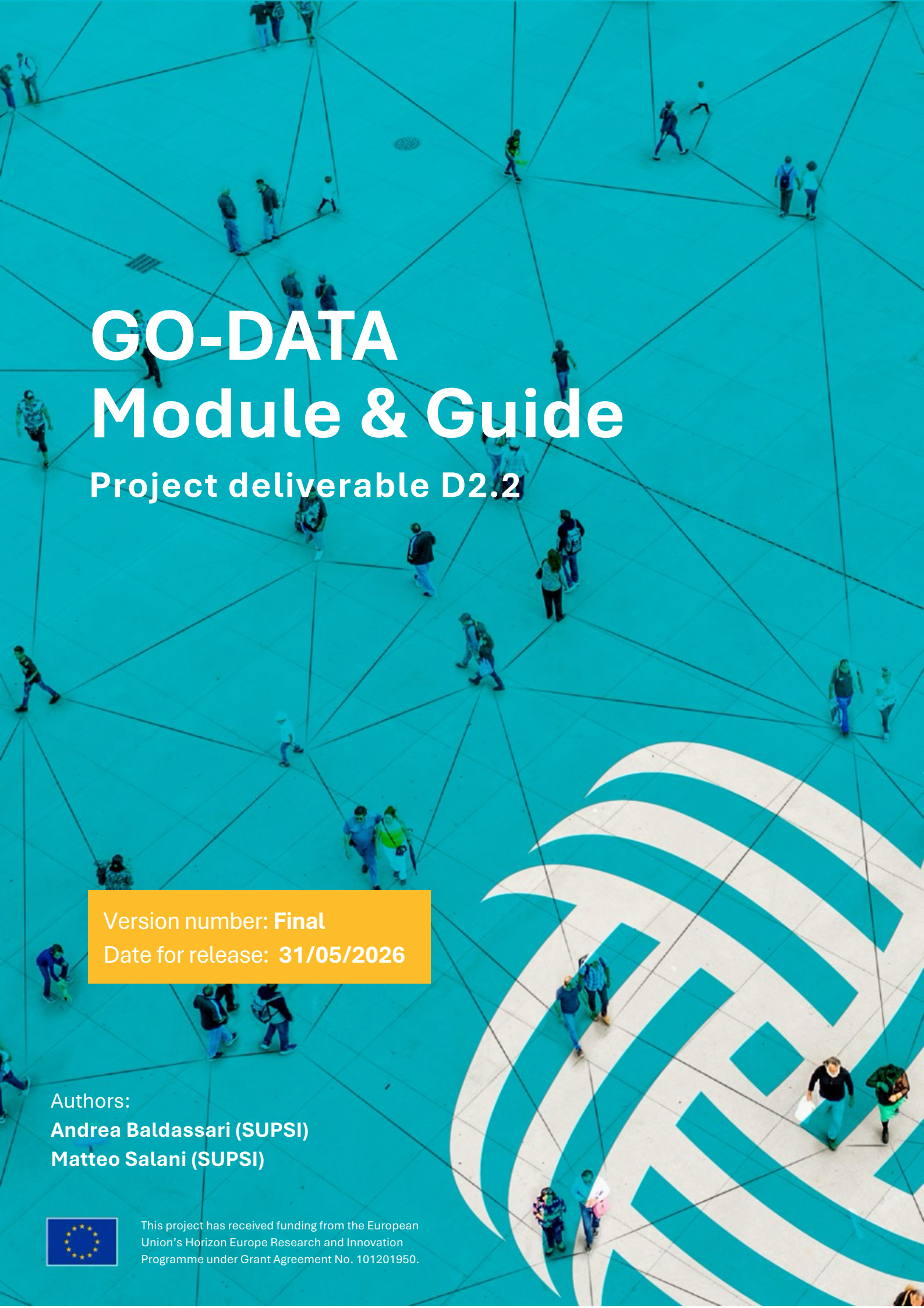




GO-DATA Module & Guide

Project deliverable D2.2

Version number: **Final**

Date for release: **31/05/2026**

Authors:

Andrea Baldassari (SUPSI)

Matteo Salani (SUPSI)



This project has received funding from the European Union's Horizon Europe Research and Innovation Programme under Grant Agreement No. 101201950.

Deliverable administrative information

Deliverable Administration					
Grant Agreement	101201950	Project short name	GOLIA		
Deliverable no.	D2.2	Name	GO-DATA		
Status	Final	Due	31/05/2026	Date	29/05/2026
Author(s)	Andrea Baldassari (SUPSI), Matteo Salani (SUPSI)				
Related tasks	GO-DATA, qualitative and quantitative knowledge base and data space				
Dissemination level	PU - Public				
Document history	Version	Date	Submitted	Reviewed	Comments
	V.01	04/05/2026	Andrea Baldassari	-	First version
	V.02	28/05/2026	Andrea Baldassari	Maria Diez (SINTEF), Simen Støa (SINTEF)	Revised version
	V.03	29/05/2026	Andrea Baldassari Matteo Salani		Final version

Legal Disclaimer

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.

While the information contained in the document is believed to be accurate, the authors or any other participant in the GOLIA consortium make no warranty of any kind regarding this material including, but not limited to the implied warranties of merchantability and fitness for a particular purpose.

Neither the GOLIA Consortium nor any of its members, their officers, employees, or agents shall be responsible or liable in negligence or otherwise howsoever in respect of any inaccuracy or omission herein. Without derogating from the generality of the foregoing neither the GOLIA Consortium nor any of its members, their officers, employees, or agents shall be liable for any direct or indirect or consequential loss or damage caused by or arising from any information advice or inaccuracy or omission herein.

Copyright message

©GOLIA Consortium. This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation, or both. Reproduction is authorised provided the source is acknowledged.

About GOLIA

GOLIA is a project focused on Governing, Optimising and Leveraging Innovation in cities through data-driven and social-optimum led approach. It aims to ensure an adaptive, accessible and just mobility system planned and governed with the help of digital assets, but also with the inclusive and co-creational policy making processes. Through the integration of STEM and SSH expertise and approaches, GOLIA will revolutionize spatial and mobility planning by co-developing and testing vision-led Social Optimum Mobility Index, a metric to assess urban mobility systems based not only on economic variables but also on indicators such as social support, healthy life quality or freedom to make life choices. The index will be location and time dependent to enable dynamic management of transport offer and spaces.

The project will analyse the policy making processes, citizen engagement practices and digital assets of the three Partner cities and with the gained knowledge, it will create an integrated, interdisciplinary and Adaptive Mobility Smart Governance Model. GOLIA aims at understanding to which extent and under which conditions, data generated by Smart Cities supported by AI-based tools novel communication and consultation channels and methodologies to engage citizens and stakeholders might foster the establishment of a new inclusive and adaptive mobility governance. GOLIA's goal is to identify governance models able to deliver policies resilient to political instability, extreme weather events, pandemics or any other more or less disruptive event, as they are co-developed, co-designed, and co-monitored by and for citizens and firms.

1 Deliverable Detailed Information

1.1 Key words

GO-DATA, GO-X, Architecture, modules, integration, cloud, data management

1.2 GOLIA Outputs Mapping to GA Commitments

GOLIA GA Item	GOLIA GA Item Description	GA document Chapter	Justification
DELIVERABLE			
D2.2	<i>D2.2 – GO-DATA module & guide</i>	<i>Chapter 3, section 3.1.4</i>	<i>This document contains the guide of the GO-DATA module. It details the system architecture, data gathering interfaces, and integration protocols designed to harmonize and characterize both qualitative and quantitative digital assets from the partner cities.</i>
TASK			
T2.2	<i>The tasks aim at constituting the backbone necessary as a support for the other GO-X modules.</i> <i>In essence, in this task we will explore existing digital assets available at partner cities and formalize them in the architecture conceived in task T2.1, we will thus perform classification, characterization, interoperability potential assessment.</i> <i>We represent digital assets with the support of standardized ontologies which will be explored, extended when necessary, implemented and tested.</i>	<i>Chapter 3, section 3.1.3 – Detailed Work Package descriptions</i>	<i>This document fulfills the requirements by providing a guide to the GO-DATA module, integrated within the GO-X architecture. It provides a description of the main high-level functionalities, their operation, and a comprehensive guide for their use and integration within the GO-X architecture. It also describes its installation within the project's cloud infrastructure.</i>

Table 1: Deliverable Adherence to Grant Agreement deliverable and work description

2 Executive summary

This deliverable serves primarily as a user guide for the GO-DATA software module, rather than a standard project report, providing a comprehensive technical description of the central data management module within the GOLIA project ecosystem.

As the foundational component of the GO-X architecture, GO-DATA acts as a centralized "Data Hub," ensuring the seamless ingestion, processing, and distribution of heterogeneous data to support advanced decision-making services.

The document highlights the strategic shift towards a Hub-and-Spoke architecture. This modular design allows the platform to maintain a single source of truth (the Hub) while enabling various specialized analytical modules (the Spokes)—such as GO-INFO, GO-KNOW, and GO-WISDOM—to consume normalized data through standardized RESTful APIs. This approach guarantees high scalability, easier maintenance, and robust interoperability across the entire GOLIA framework.

Key highlights of the GO-DATA module include:

- **Diverse Data Ingestion:** The system integrates multiple data streams, including municipal Open Data via a dedicated "Data Bridge," private proprietary datasets, and qualitative social media feeds.
- **Sentiment Analysis Engine:** A sophisticated NLP (Natural Language Processing) pipeline is implemented to monitor social trends (initially focused on Reddit), providing city stakeholders with actionable insights into public perception.
- **Modern Technical Stack:** The module has been entirely engineered in **Python** to leverage modern data science libraries and ensure a lightweight, high-performance execution environment.
- **Privacy and Security by Design:** Full compliance with **GDPR** is ensured through automated data anonymization and by hosting all infrastructure and storage instances within the European Union (Google Cloud Platform, Spain region).

In conclusion, GO-DATA establishes the necessary infrastructure to power the upcoming pilot phases of the GOLIA project. Future developments will focus on expanding social media coverage (integration with Facebook and X), onboarding additional urban datasets, and providing continuous technical support to the specialized analytical modules as the project moves toward large-scale validation in partner cities.

Table of contents

DELIVERABLE ADMINISTRATIVE INFORMATION.....	1
LEGAL DISCLAIMER	2
COPYRIGHT MESSAGE.....	2
ABOUT GOLIA	3
1 DELIVERABLE DETAILED INFORMATION	4
1.1 KEY WORDS	4
1.2 GOLIA OUTPUTS MAPPING TO GA COMMITMENTS	4
2 EXECUTIVE SUMMARY.....	5
3 LIST OF ABBREVIATIONS AND ACRONYMS	8
4 PURPOSE OF THE DELIVERABLE.....	10
4.1 ATTAINMENT OF THE OBJECTIVES AND EXPLANATION OF DEVIATIONS.....	10
4.2 INTENDED AUDIENCE	10
4.3 LINKS WITH OTHER WORK PACKAGES/DELIVERABLES	11
4.3.1 WP1 – CONTRIBUTOR.....	11
4.3.2 WP3, WP4 AND WP5	11
4.3.3 WP6	11
4.3.4 WP7	11
5 ARCHITECTURE AND MODULE DESCRIPTION	12
5.1 GO-DATA ROLE IN THE GO-X MODEL	12
5.1.1 GO-X ARCHITECTURE RECAP	12
5.1.2 GO-DATA AS A HUB.....	14
5.2 PRINCIPLES AND FEATURES OVERVIEW	15
5.2.1 DATA BRIDGE	15
5.2.2 CROSS-MODULE DATA HANDLING.....	16
5.2.3 SENTIMENT ANALYSIS.....	16
6 GUIDE.....	18
6.1 CONNECTORS CONFIGURATION	18

6.2	AUTHENTICATION.....	19
6.2.1	OBTAINING AN ACCESS TOKEN.....	19
6.2.2	EXAMPLE RESPONSE.....	19
6.2.3	USING THE TOKEN.....	19
6.3	REST API FOR DATA CONSUMPTION	20
6.3.1	LIST AVAILABLE RESOURCES.....	20
6.3.2	FETCH JSON DATASET	20
6.3.3	DOWNLOAD BINARY RESOURCE FILE	21
6.4	DATA UPDATE NOTIFICATION SYSTEM	21
6.4.1	MESSAGE BROKER CONFIGURATION	21
6.4.2	EVENT PAYLOAD STRUCTURE	22
6.5	REST API FOR DATA PUBLICATION	23
6.5.1	STORE CUSTOM DATA	23
6.5.2	RETRIEVE STORED DATA	23
6.6	CREATION AND CONFIGURATION OF A SENTIMENT ANALYSIS INSTANCE	24
6.6.1	REGISTER A REDDIT POST FOR TRACKING	24
6.6.2	THE POLLING ENGINE CONFIGURATION	24
6.6.3	RETRIEVE TRACKED POSTS AND REPORTS	25
7	CONCLUSIONS	26
7.1	SUMMARY OF THE PROPOSED ARCHITECTURE	26
7.2	FUTURE DEVELOPMENT	26

List of figures

Figure 1 - GO-X Architecture recap	13
Figure 2 - GO-DATA logo	14

List of tables

Table 1: Deliverable Adherence to Grant Agreement deliverable and work description.....	4
---	---

3 List of abbreviations and acronyms

Acronym	Meaning
AI	Artificial Intelligence
API	Application Programming Interface
CINEA	European Climate, Infrastructure and Environment Executive Agency
CSV	Comma-Separated Values
EC	European Commission
EU	European Union
GA	Grant Agreement
GCP	Google Cloud Platform
GDPR	General Data Protection Regulation
GIS	Geographic Information System
GTFS	General Transit Feed Specification
HTTP	Hypertext Transfer Protocol
IoT	Internet of Things
JSON	JavaScript Object Notation
Maas	Mobility As a Service
MQTT	Message Queuing Telemetry Transport
NLP	Natural Language Processing
PU	Public
REST	Representational State Transfer
SIRI	Service Interface for Real Time Information
SOMI	Social Optimum Mobility Index
SSH	Social Sciences and Humanities
STEM	Science, Technology, Engineering and Mathematics

SUPSI Scuola Universitaria Professionale della Svizzera Italiana

ToS Terms of Service

TTL Time To Live

URL Uniform Resource Locator

WP Work Package

YAML Yet Another Markup Language

4 Purpose of the deliverable

4.1 Attainment of the objectives and explanation of deviations

The objectives associated with this deliverable have been achieved in full and in accordance with the project's original timeline. The GO-DATA module successfully establishes the foundational data layer of the GO-X architecture, enabling the seamless ingestion and management of mobility and sentiment data. It is important to note that the integration of additional data sources is an iterative process that will continue throughout the following months, during which the GO-DATA team will provide ongoing technical support for the integration and development of subsequent modules.

Regarding the technical implementation, no significant deviations from the project's strategic scope occurred. However, two main technical refinements were implemented to optimize the system's performance and maintainability:

- **Software Language:** While the module was initially envisioned in Java, it has been completely refactored using Python. This pivot ensures more agile management, easier maintenance, and better compatibility with modern data processing libraries.
- **Cloud Infrastructure:** The cloud provider has been updated from the initial suggestion in D2.1 (Amazon AWS) to Google Cloud Platform (GCP). This transition was conducted with rigorous adherence to GDPR regulations; specifically, all data centers and storage instances are located within the European Union (specifically in the metropolitan area of Madrid, Spain), ensuring full compliance with European data sovereignty and privacy standards.

4.2 Intended audience

Given its nature as a technical specification and operational manual, this deliverable is primarily intended for software developers and system architects responsible for the implementation of the various components within the project's technical ecosystem.

The document serves as a comprehensive Technical Guide and User Manual, providing the necessary instructions for interacting with the GO-DATA module. It is specifically designed to support technical teams in:

- Understanding the underlying data structures and the storage logic implemented in the system;
- Configuring and deploying the connectors required for data ingestion;
- Seamlessly integrating downstream components with the GO-DATA REST APIs for efficient data consumption and publication;
- Implementing and managing real-time notification listeners for event-driven interactions.

While this deliverable may provide useful context for project coordinators and stakeholders, its core content is tailored to provide the technical expertise and procedural details required for the successful interoperability and integration of the platform's data services.

4.3 Links with other work packages/deliverables

4.3.1 WP1 – Contributor

GO-DATA implements the technical requirements and data models defined in WP1. It translates the identified "city digital assets" and SSH (Social Sciences and Humanities) requirements into functional data connectors.

4.3.2 WP3, WP4 and WP5

This deliverable provides the data infrastructure necessary for the execution of the pilot projects in different cities. GO-DATA acts as the primary data provider for all services developed within these work packages.

4.3.3 WP6

WP6 is dedicated to the comprehensive evaluation and validation of the GOLIA platform and the GO-X modules across the designated pilot sites. To ensure a scientifically rigorous assessment of the project's outcomes, the evaluation framework ingests and analyzes two distinct categories of data:

- **Quantitative Evaluation Data:** This includes high-resolution mobility metrics and Key Performance Indicators (KPIs) automatically generated by the platform or provided by local transport authorities. Key data streams consist of traffic flow measurements, environmental indicators (e.g., emissions reduction estimates), modal split shifts, and system performance logs (such as the processing times and predictive accuracy of the GO-KNOW engine).
- **Qualitative Evaluation Data:** This encompasses user acceptance, socio-economic perceptions, and behavioral insights. This data is collected directly from stakeholders, transit operators, and citizens through structured surveys, focus groups, and post-pilot interviews.

4.3.4 WP7

GO-DATA handles data storage in compliance with the GDPR requirements and ethical guidelines established in this WP, specifically regarding the data center localization and data anonymization processes.

5 Architecture and module description

5.1 GO-DATA role in the GO-X model

5.1.1 GO-X Architecture recap

The GO-X Architecture represents the technical and conceptual framework designed to support the GOLIA project's vision of a unified urban mobility ecosystem. Its primary objective is to overcome the limitations of traditional, siloed data systems by providing a scalable, interoperable, and secure environment where heterogeneous digital assets - ranging from real-time IoT sensor data to qualitative social media streams - can be seamlessly integrated and leveraged.

To ensure maximum flexibility and resilience, the architecture adopts a Hub-and-Spoke pattern. This paradigm shift is central to the project's ability to manage complex data lifecycles while maintaining high performance and ease of maintenance.

- **The nexus: GO-DATA (The Hub)** At the core of the GO-X ecosystem lies GO-DATA, serving as the central Hub and the single source of truth for the entire platform. Unlike a mere storage layer, the Hub acts as a dynamic "Data Engine" responsible for the orchestration of ingestion, validation, normalization, and persistence. By centralizing these critical functions, GO-DATA abstracts the technical complexity of raw data handling from the rest of the system. It ensures that any data consumed by the peripheral modules is already standardized, clean, and compliant with the project's data models, thereby reducing the computational overhead on the « Spokes ».
- **The analytical layers: The spokes** Radiating from the GO-DATA Hub are the specialized modules, or "Spokes," each dedicated to a specific stage of value extraction. This decoupling allows each spoke to operate independently, using the Hub as its primary interface for data acquisition and publication:
 - **GO-INFO (information spoke):** This layer is dedicated to situational awareness. It consumes raw streams from the Hub to generate real-time information, such as traffic density alerts or environmental monitoring, providing an immediate snapshot of the city's status. Crucially, it acts as a bidirectional communication channel, allowing the platform to interact directly with the public.
 - **GO-KNOW (knowledge spoke):** This module is dedicated to assessing the potential for Mobility as a Service (MaaS) implementation and scalability across different settlement environments. GO-KNOW provides a structured evaluation framework that generates a MaaS Readiness Index, across technical, organizational, and social dimensions. Target Users: City planners, transport technicians, local authorities, and policymakers who require a structured tool to evaluate their current capacity for MaaS integration and need tailored strategic guidelines for future deployment.
 - **GO-WISDOM (decision support spoke):** The most advanced layer of the architecture, GO-WISDOM, acts as the decision-support system. It integrates the outputs from the other spokes to run simulations, perform "what-if" analyses, and provide strategic recommendations for policy-making and urban optimization. Target Users: Policy makers, city planners, and municipal executives responsible for long-term strategic decisions and urban governance.

The adoption of the Hub and Spoke model provides the GOLIA project with three fundamental advantages:

1. **Scalability:** New data sources or specialized processing modules can be added as new "spokes" without requiring modifications to the existing infrastructure, simply by connecting them to the GO-DATA API.
2. **Maintainability:** By centralizing data governance within the Hub, updates to data schemas or security protocols (such as GDPR-compliant anonymization) can be implemented in a single location.
3. **Interoperability:** The Hub ensures that all modules speak the same "language," facilitating a seamless flow of information across the different layers of the GO-X ecosystem through standardized RESTful services and event-driven notifications.

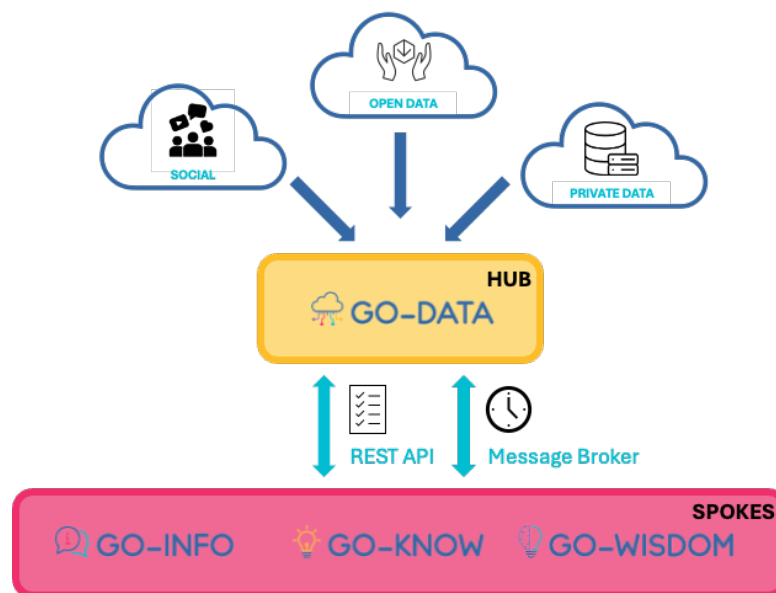


Figure 1 - GO-X Architecture recap

5.1.2 GO-DATA as a hub



As the central nexus of the GO-X Architecture, GO-DATA operates as a sophisticated data orchestrator, managing the flow of information across the entire ecosystem. Its role as a "Hub" is not merely limited to storage; it acts as a gatekeeper and a normalization engine for four distinct types of data streams and interactions:

- **Management of Open Data (City Portals) :** GO-DATA is designed to interface seamlessly with public administration infrastructures. It automates the ingestion of real-time and static datasets from municipal Open Data portals (e.g., those provided by the city partners in the GOLIA consortium). By implementing dedicated connectors for standard formats such as GTFS for public transport schedules and SIRI for real-time updates, the Hub ensures that public mobility data is constantly refreshed and available for downstream analysis without manual intervention.
- **Management of Private and Proprietary Data Sources** Beyond public feeds, the Hub manages high-value private datasets that are not accessible to the general public. These assets are typically provided by commercial partners or transport operators in the form of specific files (e.g., CSV, JSON, or proprietary logs). GO-DATA provides a secure environment for these assets, ensuring that they are handled according to specific licensing agreements or bilateral "Data Sharing Agreements." The Hub manages access rights and ensures that proprietary information is used strictly within the bounds of the project's legal framework.
- **Social Media Ingestion for Sentiment Analysis** To support the "Social Science and Humanities" (SSH) requirements of the project, GO-DATA acts as the primary collector for qualitative data. It performs automated scraping and API-based collection from social media platforms to capture public perceptions regarding urban mobility. These raw qualitative streams are ingested, timestamped, and prepared for the Sentiment Analysis engines, allowing the project to correlate physical mobility patterns with citizen satisfaction and feedback.
- **Inter-Module Data Exchange (Spoke Synchronization)** Following the strict Hub and Spoke pattern, GO-DATA serves as the exclusive intermediary for all communications between the specialized modules (GO-INFO, GO-KNOW, and GO-WISDOM). Direct "Spoke-to-Spoke" data transfers are prohibited to maintain system integrity and a unified audit trail. When a module like GO-INFO generates a situational alert, it must "publish" this information back to the GO-DATA Hub, which then "notifies" or provides the data to other interested spokes. This centralized exchange ensures data consistency and prevents the creation of information silos.
- **GDPR Compliance and Data Sovereignty** A core principle of the GO-DATA Hub is the **strict** enforcement of GDPR regulations. As the single entry and exit point for data, the Hub implements "Privacy by Design" protocols. All ingested data—especially sensitive mobility patterns or social media feeds—undergo rigorous control processes, including:
 - **Anonymization and Pseudonymization:** Ensuring that no personal identifiers are propagated to the analytical spokes.

- **Geographic Sovereignty:** All data processing and persistence occur within the European Union, specifically utilizing the project's designated infrastructure in the Spain region, ensuring full compliance with EU data protection laws and the project's Ethics requirements (WP7).

5.2 Principles and features overview

5.2.1 Data bridge

The Data bridge is the operational interface of the GO-DATA module, specifically designed to bridge the gap between the GO-X internal ecosystem and the external data world. Its primary function is to provide a standardized ingestion path for heterogeneous datasets, regardless of their origin, format, or delivery method.

The Data Bridge manages two primary categories of external integration:

- **Remote source connectors:** These are dynamic bridges established with external servers, such as Municipal Open Data portals, IoT brokers (MQTT), or third-party REST APIs. The bridge handles the authentication, scheduled polling, or web-socket subscriptions required to keep the data flow active.
- **Direct file ingestion:** For datasets that are not available through live services (e.g., historical logs, proprietary spreadsheets, or static mobility surveys), the bridge allows for direct file uploads. These files are parsed and normalized by the GO-DATA engine to be made available to the "Spoke" modules in a structured format.

The principle of data minimization and caching Logic In alignment with the project's commitment to efficiency and GDPR compliance, GO-DATA follows a "minimal storage" policy. Unlike traditional data warehouses that mirror entire external databases, the GO-DATA Hub acts primarily as a high-performance cache for raw external data.

- **Ephemeral data:** Raw data fetched from external bridges is kept in the system only for the duration necessary to serve the immediate needs of the downstream modules (GO-INFO, GO-KNOW, GO-WISDOM). This significantly reduces the platform's storage footprint and mitigates privacy risks.
- **Persistent storage exceptions:** Persistence is strictly reserved for value-added data. This includes high-level aggregations, historical trends processed by the analytical spokes, and strategic outputs that must be preserved for long-term policy evaluation.

To enable the "Spokes" to operate reliably, the Data Bridge associates every ingested dataset with a comprehensive set of metadata. This metadata layer is crucial for the internal governance of the Hub and includes:

- **Source health and availability:** Real-time status of the external connection (e.g., whether a city portal is currently reachable).
- **Data freshness (TTL):** Timestamps indicating when the data was last refreshed and its expected expiration or "Time-to-Live" (TTL).
- **Reliability index:** Information regarding the frequency of updates and the historical consistency of the source.

By querying these metadata, the downstream modules can autonomously decide whether the available data is sufficient for their specific logic or if a manual refresh/alternative source is required, ensuring the overall robustness of the GO-X Architecture.

5.2.2 Cross-module data handling

The architecture is designed around a "hub-and-spoke" model, ensuring data consistency, persistence, and real-time responsiveness across all specialized modules.

GO-DATA acts as the central orchestrator for data exchange. Unlike a peer-to-peer network where each module would need to interface with every other module, the GO-X architecture mandates that:

- **Spoke Integration:** All specialized modules (spokes) communicate exclusively with GO-DATA for cross-cutting data needs.
- **Standardization:** This centralized approach simplifies the integration of new modules and ensures a single source of truth for the entire platform.

GO-DATA is responsible for the lifecycle of shared information generated throughout the ecosystem. Specifically:

- **Information Storage:** When a module generates new data, such as **SOMI (Social Optimum Mobility Index)** or specific **MaaS (Mobility as a Service) score indexes**, these are sent to GO-DATA for persistence.
- **On-Demand Retrieval:** These persisted datasets are made available to other modules upon request, allowing for a decoupled environment where modules do not need to be active at the same time to share information.

Beyond simple storage, GO-DATA serves as an intelligent data layer. It provides advanced processing capabilities to reduce the computational load on individual spokes:

- **Aggregations:** GO-DATA can perform complex data summaries and statistical aggregations across datasets originating from multiple sources.
- **Filtering:** Modules can request specific subsets of data by applying granular filters, ensuring that they only receive the information relevant to their specific functional domain.

To ensure the ecosystem remains synchronized and reactive to changing conditions, a real-time notification system is implemented:

- **Subscription Model:** Modules can "subscribe" to specific data topics or events managed by GO-DATA.
- **Push Notifications:** Whenever new information is generated or existing data is updated, GO-DATA automatically broadcasts notifications to all subscribed modules. This eliminates the need for constant polling and ensures that all components of the system operate on the most current data available.

5.2.3 Sentiment analysis

Sentiment analysis is the process of computationally identifying and categorizing opinions expressed in a piece of text. The primary objective is to determine the user's attitude toward a specific topic—whether it is positive, negative, or neutral.

By aggregating these sentiments over a defined period, the system provides a clear metric of public response, allowing project stakeholders to visualize trends and shifts in opinion regarding specific mobility themes or urban interventions.

The system is highly flexible in how it targets data sources for sentiment analysis, allowing the consuming module to specify exactly which topics or specific posts should be monitored. The system can be configured to track two main types of public engagement:

- **Existing Discussions:** Tracking pre-existing threads or posts generated by external users and the general public that are relevant to the selected topic.
- **Ad-hoc Discussions:** Monitoring targeted posts specifically created to trigger, stimulate, and observe public debate on a particular subject of interest. In this case, the responsibility for

initiating and creating the post lies entirely with the specific module that will consume and analyze the resulting information.

To account for the diverse demographic of users, the underlying analytical model is language-agnostic. It is capable of processing and analyzing interactions in any language. Furthermore, the model can handle **code-switching**, where multiple languages are used within the same post or comment thread, ensuring that the sentiment score remains accurate regardless of the linguistic complexity.

The monitoring parameters are fully customizable to meet the needs of different study cases:

- **Configurable Duration:** Users can define the total window of time for which a post remains under active monitoring.
- **High-Frequency Updates:** The system performs checks multiple times per day. The resulting data is refreshed every **X hours** (where X is defined in the configuration), providing a near-real-time view of how the conversation is evolving.

At the current stage of development, the system interfaces exclusively with Reddit, utilizing its structured thread format to gather high-quality discussion data. However, the roadmap includes the expansion of this capability to other major social networks:

- **Planned Extensions:** Integration with Facebook and X (formerly Twitter).
- **Timeline:** These additional sources are scheduled to be fully operational by M18 (Month 18) of the project.

Data integrity and user privacy are fundamental to the Sentiment Analysis module. The system operates under a "processing-on-the-fly" principle:

- **No Data Retention:** The system does not save or store raw text or content from social media platforms. It processes the data in real-time to extract the sentiment score and then discards the source text.
- **Anonymization:** Personal data and user identifiers are systematically ignored as they are irrelevant to the aggregate sentiment analysis.
- **Compliance:** This methodology ensures full compliance with GDPR regulations and strictly adheres to the Terms of Service (ToS) of the utilized social networks.

6 Guide

6.1 Connectors configuration

The GO-DATA platform connects to various external city OpenData APIs through a series of dedicated adapters. The core configuration for these connectors is centralized in a YAML file located at [app/adapters/config.yaml](#). In the current version of the module, this setup requires system administrators or developers to manually configure the connectors within the service. This file dictates how the system interacts with each external municipality, defining connection parameters, caching rules, and data formats.

To maintain a standardized access pattern across different municipalities, the configuration relies heavily on identifiers:

- **City Identifier:** This is the top-level key in the YAML structure (e.g., florence, antwerp, pilsen). It uniquely identifies the municipality within the platform and is directly used by consumers in the API path to route requests to the correct adapter.
- **Resource Identifier:** Nested under the city, this key (e.g., river_crossing, ztl_daytime) uniquely identifies a specific dataset or file provided by that city. Like the city identifier, this string is used in the REST API URL to specify exactly which data to consume.

Below is a descriptive excerpt of how a city is configured within the [config.yaml](#) file:

```
pilsen:
  # Base URL of the external city OpenData API
  base_url: "https://opendata.plzen.eu/api"
  # Maximum HTTP request timeout in seconds
  timeout: 10

  resources:
    # 'transit_stops' is the Resource Identifier
    transit_stops:
      # 'endpoint' is appended to the base_url for JSON APIs, 'URL' an absolute URL
      url: "https://opendata.plzen.eu/public/opendata/detail/9?detail-fileId=20&do=detail-downloadFile"
      method: "GET"
      format: "shp"
      # Time-to-live for the internal cache (in seconds)
      cache_ttl: 86400
```

Configuration Properties Breakdown:

- **base_url:** The root URL of the target city's OpenData API.
- **timeout:** The maximum allowed time (in seconds) for HTTP requests to the external API.
- **resources:** A dictionary mapping Resource Identifiers to their specific retrieval rules.
 - **endpoint:** Used for standard JSON APIs. It represents the specific API path appended to the **base_url**.
 - **url:** Used for binary resources (like Shapefiles or ZIP archives). It allows specifying an absolute download link that ignores the **base_url**.
 - **format:** Describes the file format (defaults to json for endpoints, but can be shp, zip, etc. for raw file downloads).
 - **cache_ttl:** The Time-To-Live in seconds. It defines how long the GO-DATA platform will cache this resource before fetching a fresh copy from the external API.

6.2 Authentication

All REST API endpoints exposed by the platform (with the exception of the `/health` check) are secured and require authentication. The system implements a secure authentication model using standard OAuth2 flows, issuing signed JSON Web Tokens (JWT) for access control.

6.2.1 Obtaining an Access Token

To interact with the data consumption, publication, or sentiment analysis APIs, client applications must first request a short-lived access token by calling the dedicated authorization endpoint.

- **Endpoint:** `POST /auth/token`
- **Content-Type:** `application/x-www-form-urlencoded`
- **Payload Requirements:** The request must include the registered **username** (acting as the Client ID) and **password** (acting as the Client Secret).

```
POST /auth/token HTTP/1.1
Host: go-data.go-modules.com
Content-Type: application/x-www-form-urlencoded
username=my_registered_client&password=my_secure_secret
```

6.2.2 Example Response

Upon successful validation of the credentials against the internal client store, the server generates a JWT. This token contains the required operational scopes (such as the machine scope used for system-to-system interactions).

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "token_type": "bearer",
  "expires_in": 3600
}
```

*Note: The **expires_in** value indicates the token's time-to-live in seconds before a new one must be requested.*

6.2.3 Using the Token

Once the **access_token** is obtained, it must be included in the HTTP Authorization header of all subsequent API requests using the Bearer authentication scheme:

```
Authorization: Bearer <your_access_token>
```

If the token is missing, structurally invalid, expired, or lacks the appropriate internal scope to access a specific endpoint, the API will safely reject the request and return either a **401 Unauthorized** or a **403 Forbidden** error response.

6.3 REST API for Data consumption

The platform abstracts the complexity of interacting with various external municipal APIs by exposing a unified, standardized set of REST API endpoints. All data consumption APIs are protected and require a valid JWT Bearer token with machine scope, obtainable via the OAuth2 client credentials flow (**POST /auth/token**).

To request data, clients do not need to know the underlying base URLs, specific external endpoints, or authentication mechanisms of the target municipality. Instead, the routing is entirely based on the **Identifiers** defined in the configuration (as explained in Chapter 6.1):

1. **City Identifier:** Used to select the correct external adapter (e.g., florence, pilsen).
2. **Resource Identifier:** Used to specify the exact dataset or file to retrieve (e.g., **ztl_daytime**, **transit_stops**).

By injecting these two identifiers directly into the API path parameters, the GO-DATA platform acts as an intelligent proxy. It checks the internal high-performance cache and either returns the stored data instantly or dynamically fetches, processes, and caches the fresh data from the external provider.

6.3.1 List Available Resources

- **Endpoint:** **GET /v1/{city}/resources**
- **Description:** This endpoint allows clients to discover which datasets are currently configured and available for a specific municipality.
- **Path Parameters:**
 - **city** (string): The valid City Identifier (e.g., pilsen).
- **Response:** Returns a JSON object containing the city name and an array of all available Resource Identifiers strings.

6.3.2 Fetch JSON Dataset

- **Endpoint:** **GET /v1/{city}/{resource_type}**
- **Description:** Retrieves standard JSON data for a given dataset. It heavily utilizes the internal cache based on the adapter's TTL configuration to ensure fast response times and prevent overloading the external city APIs.
- **Path Parameters:**
 - **city** (string): The City Identifier.
 - **resource_type** (string): The Resource Identifier.
- **Query Parameters:**
 - **limit** (integer, optional): The number of records to return per request for pagination. Default is 100, maximum is 1000.
 - **offset** (integer, optional): The number of records to skip. Default is 0.
 - **fields** (string, optional): A comma-separated list of fields to include in the response payload.
- **Response:** Returns a standardized JSON structure containing:
 - **metadata:** A block detailing the city, resource_type, current timestamp, last_updated time, pagination counts (count, total), the original source URL, and a **cache_hit** boolean flag useful for debugging.
 - **data:** The actual array of JSON records fetched from the external source.

6.3.3 Download Binary Resource File

- **Endpoint:** `GET /v1/{city}/{resource_type}/download`
- **Description:** Designed for GIS or raw file datasets (like Shapefiles or ZIP archives). It retrieves the binary file either from the internal cache or directly from the external provider's absolute URL.
- **Path Parameters:**
 - **city** (string): The City Identifier.
 - **resource_type** (string): The Resource Identifier for a binary resource (e.g., `transit_stops` in Pilsen).
- **Response:** Returns a raw binary file stream (e.g., `application/zip`). To provide auditing context without modifying the file content, the platform injects several custom HTTP headers into the response:
 - **X-GO-DATA-City:** The requested city.
 - **X-GO-DATA-Resource-Type:** The requested resource type.
 - **X-GO-DATA-Timestamp:** The time of the request.
 - **X-GO-DATA-Last-Updated:** The timestamp when the file was last fetched from the external source.
 - **X-GO-DATA-Cache-Hit:** `true` or `false`.
 - **X-GO-DATA-Source:** The origin URL of the file.

6.4 Data update notification system

To ensure clients always have access to the most up-to-date municipal data without the need to continuously poll the REST APIs, the platform features an asynchronous, event-driven notification system. Behind the scenes, a background process (the **TTLWatcher**) continuously monitors all configured city resources. At the end of every cache TTL period, it fetches the latest data from the external OpenData provider and compares it against the last known state using a SHA-256 hash. If a change is detected, the internal cache is silently updated, and a real-time notification is broadcasted.

6.4.1 Message Broker Configuration

The platform utilizes **RabbitMQ** to distribute these update events. Clients wishing to receive real-time notifications must connect to the broker and bind their queues to the platform's main exchange.

- **Exchange Name:** By default, notifications are published to the `go_data.resource_updates` exchange (this can be customized by the platform administrator via the `rabbitmq_exchange` environment variable).
- **Exchange Type:** TOPIC. This allows subscribers to perform advanced filtering using routing keys.
- **Subscribing to Events (Routing Keys)**

When a change is detected, the platform publishes the message using a specific routing key format: `{city}.{resource_type}`

By declaring a queue and binding it to the Topic Exchange, client applications can choose exactly what to listen to:

- **Specific Resource:** Bind with `florence.ztl_daytime` to receive updates only when the daytime ZTL boundaries in Florence change.
- **City-wide Updates:** Bind with `pilsen.*` to receive an event whenever *any* monitored dataset in Pilsen is updated.
- **Global Updates:** Bind with `#` to receive all notifications across the entire platform.

6.4.2 Event Payload Structure

The message body is delivered in standard JSON format (application/json) and contains essential metadata about the update.

```
{
  "event": "resource_updated",
  "city": "pilsen",
  "resource_type": "transit_stops",
  "timestamp": "2026-04-30T14:01:26.123456+00:00",
  "count": 48256,
  "source": "https://opendata.plzen.eu/public/opendata/detail/9?detail-fileId=20&do=detail-downloadFile"
}
```

- **event:** A static string identifier for the event type (currently resource_updated).
- **city:** The City Identifier where the change occurred.
- **resource_type:** The specific Resource Identifier that was updated.
- **timestamp:** ISO-8601 formatted UTC timestamp of when the change was detected and processed by the watcher.
- **count:** For JSON datasets, this represents the number of records. For binary files (like Shapefiles), this represents the size of the file in bytes.
- **source:** The original external API endpoint or file URL where the fresh data was retrieved from.

6.5 REST API for Data publication

In addition to consuming external municipal data, the platform provides a robust, generic, and secure data store. Other modules can use these APIs to publish and persist custom JSON payloads directly within the GO-DATA ecosystem.

As with the consumption APIs, all publication endpoints require a valid JWT Bearer token with machine scope.

6.5.1 Store Custom Data

- **Endpoint:** `POST /v1/data/store`
- **Description:** Accepts any valid JSON payload in the request body and persists it to the platform's internal database.
- **Request Body:** `application/json`

Example Request:

```
POST /v1/data/store HTTP/1.1
Host: go-data.go-modules.com
Authorization: Bearer <your_access_token>
Content-Type: application/json

{
  "sensor_id": "AQ-104",
  "location": "Piazza della Signoria, Florence",
  "metrics": {
    "pm10": 12.4,
    "temperature": 22.1
  },
  "status": "active"
}
```

Example response :

Upon successful storage, the system generates and returns a universally unique identifier (UUID) associated with the stored data.

```
{
  "id": "550e8400-e29b-41d4-a716-446655440000"
}
```

6.5.2 Retrieve Stored Data

- **Endpoint:** `GET /v1/data/store/{item_id}`
- **Description:** Retrieves a previously stored JSON payload using the unique identifier provided during the publication step.
- **Path Parameters:**
 - `item_id` (string): The UUID of the stored item.

Example request:

```
GET /v1/data/store/550e8400-e29b-41d4-a716-446655440000 HTTP/1.1
Host: api.go-data.example.com
Authorization: Bearer <your_access_token>
```

If the item exists, the server returns the exact JSON payload that was originally published. If the provided UUID does not correspond to any record in the database, a standard **404 Not Found** error is returned.

6.6 Creation and configuration of a Sentiment Analysis instance

The platform includes a specialized background service to track public sentiment about specific cities by analyzing discussions on Reddit. Once a discussion thread is registered, the system periodically polls for new comments and performs Natural Language Processing (NLP) to generate an aggregated sentiment report.

As with other internal APIs, interacting with the sentiment service requires a valid JWT Bearer token with the machine scope.

6.6.1 Register a Reddit Post for Tracking

To begin tracking a new discussion, you must provide the Reddit post URL and link it to a valid, supported city identifier.

- **Endpoint:** `POST /v1/sentiment/track`
- **Request Body:** `application/json`

Example request:

```
POST /v1/sentiment/track HTTP/1.1
Host: api.go-data.example.com
Authorization: Bearer <your_access_token>
Content-Type: application/json
```

Example response:

```
{
  "post_url": "https://www.reddit.com/r/travel/comments/example/florence_trip/",
  "city": "florence"
}
```

Upon successful registration, the API returns a **201 Created** status with the tracking details. If the provided URL is invalid or the post is already being tracked, the API will return a **400 Bad Request** or **409 Conflict**, respectively.

6.6.2 The Polling Engine Configuration

The sentiment analysis operates asynchronously in the background. The polling frequency is controlled globally via the `sentiment_poll_interval` environment variable (which defaults to 86400 seconds, or 24 hours).

Every time the interval elapses, the `sentiment_service` automatically connects to Reddit (using the internal `reddit_client_id` and `reddit_client_secret`), fetches new comments, runs the NLP model, and updates the internal metrics without blocking any standard API request.

6.6.3 Retrieve Tracked Posts and Reports

You can retrieve the current state of the sentiment analysis using two dedicated GET endpoints:

- **List all tracked posts (GET /v1/sentiment)** Returns an array of all Reddit posts currently monitored by the platform, giving a high-level overview of available analyses.
- **Get a specific sentiment report (GET /v1/sentiment/{post_id})** Retrieves the detailed sentiment report for a specific tracked post. If the `post_id` is unknown, the API returns a **404 Not Found**.

Example request:

```
GET /v1/sentiment/example_post_id HTTP/1.1
Host: go-data.go-modules.com
Authorization: Bearer <your_access_token>
```

Example response :

```
{
  "post_id": "example_post_id",
  "city": "florence",
  "post_url": "https://www.reddit.com/r/travel/comments/example/florence_trip/",
  "total_comments_analyzed": 142,
  "sentiment_score": 0.85,
  "distribution": {
    "positive": 110,
    "neutral": 20,
    "negative": 12
  },
  "last_updated": "2026-04-30T14:20:00Z"
}
```

7 Conclusions

7.1 Summary of the proposed architecture

The GO-DATA module serves as the central Hub and the single source of truth for the entire GO-X ecosystem. Operating as a dynamic "Data Engine," it orchestrates the ingestion, validation, normalization, and persistence of heterogeneous digital assets.

The transition to a **Hub and Spoke pattern** represents a strategic shift toward a more scalable and resilient environment. This architectural choice provides three fundamental advantages:

- **Scalability:** New data sources or specialized processing modules can be added as "spokes" without infrastructure modifications.
- **Maintainability:** Data governance and security protocols are centralized within the Hub.
- **Interoperability:** Standardized RESTful services and event-driven notifications ensure a seamless information flow between all modules.

The platform successfully integrates diverse data streams, including municipal Open Data , high-value private datasets , and qualitative social media streams for sentiment analysis. To ensure long-term agility, the module has been refactored using **Python**, enhancing compatibility with modern data processing libraries. Furthermore, the system implements "**Privacy by Design**" protocols, ensuring full GDPR compliance with data centers and storage instances strictly located within the European Union (Spain region).

7.2 Future development

The development of GO-DATA is an iterative process that will continue to evolve alongside the GOLIA project. Key future activities include:

- **Data Source Expansion:** Ongoing integration of additional city digital assets and data sources to support the platform's growth.
- **Support for Analytical Spokes:** Providing the necessary data infrastructure and technical support for the development of GO-INFO, GO-KNOW, and GO-WISDOM.
- **Enhanced Sentiment Analysis:** Expanding social media ingestion beyond Reddit to include Facebook and X (formerly Twitter), with full operations scheduled by Month 18 (M18).
- **Pilot Support and Impact Assessment:** Serving as the primary data provider for pilot projects in partner cities (WP3, WP4, WP5) and supporting evaluation activities within WP6.
- **Continuous Compliance:** Maintaining strict adherence to ethical guidelines and GDPR requirements, particularly regarding data anonymization and secure cross-module data exchange.