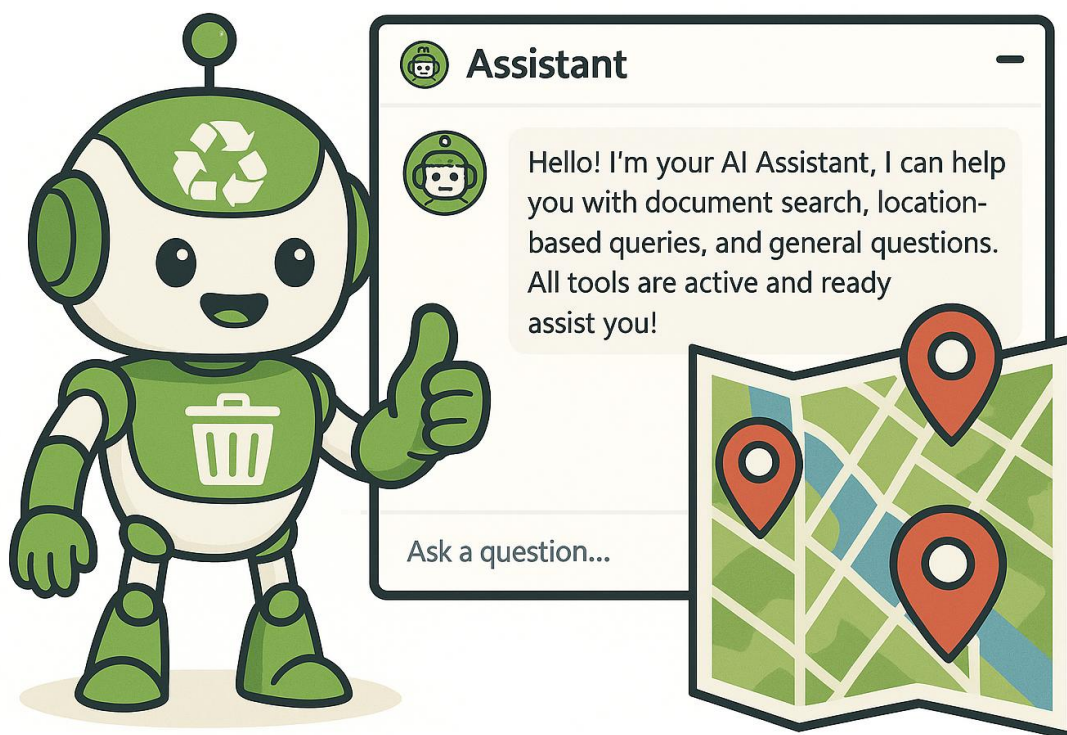


Travail de Bachelor 2025

LLM agent and dynamic GIS integration for waste management



Student : David / Marsoni
Professor : David / Wannier
Filling date : 16 July 2025

Reference for the first image

Image generated by OpenAI model gpt-image-1, <https://platform.openai.com/docs/models/gpt-image-1>

Abstract and Foreword

This bachelor's thesis investigates how advanced large language models (LLMs) understand and use dynamic Geographic Information System (GIS) data alongside building a Retrieval Enhanced Context (RAG) system to be able to transform the way citizens find information about waste management.

We first reviewed existing Swiss and international recycling map services, AI sorting systems and collection information, and new approaches like spatial RAG. Then we conducted a benchmark on the leading LLMs of big companies including OpenAI's GPT-4.1 variants, o3 and o4-mini, Anthropic Claude 4 and Claude 3.7, Google Gemini 2.5, Mistral AI models, DeepSeek, Meta's Llama 3.3, and xAI models. These models have been tested on 3 key metrics: factual accuracy, response latency, and pricing per request.

A prototype has been developed using Flask, Python, Tailwind and JavaScript to build a graphic chatbot and an administration panel to be able to manage all the aspects of the chatbot functionalities. Two agents have been created using LlamaIndex, ChromaDB, and Leaflet to serve, first, the GIS data management and response creation and on the other hand a RAG agent to create precise responses based on ingested documents.

Results show that lightweight models like GPT-4.1-nano, Mistral-Medium offer the best real-time, cost-sensitive performance for GIS queries, while mid-tier reasoning model excel at deeper RAG tasks.

This prototype was at the end packaged using Docker and then deployed on Google Cloud infrastructure.

This work was conducted with the Energy Application and System Integration Laboratory (EASILab) at HES-SO Valais under the supervision of David Wannier, teacher.

Greetings

First and foremost, I would like to thank Professor David Wannier for guiding me during the entire project and for his valuable advice and support.

I would also like to express my gratitude to Abdullah Binjos for his help, for always being available to answer my questions, for assisting me in finding the information I needed, and for encouraging me during the whole project.

I am also grateful to the following people who have assisted me:

- Gwenaëlle Gustin, for her detailed answers to my questions and her helpful advice.
- Jérémie Vianin, for responding to my questions and for granting me access to various APIs.
- Alain Duc, who gave me some advice for the poster, was here to keep me motivated.
- My family, for supporting me and helping me to stay motivated along this bachelor's thesis.

Table of Contents

Abstract and Foreword	ii
Greetings	ii
Table of Contents	iii
List of tables and figures and others	vii
Abbreviation list and specifics words	viii
1. Context	1
1.1. Introduction	1
1.2. Problematics	2
1.3. Objectives	2
2. State of the art	3
2.1. Recycling-map	3
2.1.1. Key features	3
2.1.2. Lacks and problems	3
2.2. Swiss recycle	4
2.2.1. Key features	4
2.2.2. Lacks and problems	4
2.3. Smart waste management in the world	4
2.3.1. Smart waste management with AI	4
2.3.2. Foster Engaged Community Waste Management	5
2.4. Concepts	5
2.4.1. GIS (Geographic Information System)	5
2.4.1.1. GeoJSON	5
2.4.1.2. QGIS	6
2.4.1.3. Online visualization tool for GeoJSON	6
2.4.1.4. Leaflet	6
2.4.1.5. Folium	6
2.4.2. LLMs (Large Language Models)	7
2.4.2.1. History	7
2.4.2.2. Definitions	7
2.4.2.3. Neural networks	8
2.4.2.4. Generative AI	8
2.4.2.5. Transformers, Tokens and embeddings	9
2.4.2.6. GPT (Generative Pre-trained Transformers)	10
2.4.3. RAG (Retrieval-Augmented Generation)	11
2.4.3.1. RAG Description	11
2.4.3.2. Benefits	11
2.4.3.3. RAG explains with a schema	12
2.4.3.4. Spatial-RAG	12
2.4.3.5. Vector Database	13
2.4.3.6. Python pickle files (PKL)	13
2.4.4. Agents	14
2.4.4.1. Definition	14
2.4.4.2. Key features	14
2.4.4.3. Workflow	15
2.4.4.4. Chain of thought	15
2.4.4.5. Prompt engineering	16
2.4.5. Fine Tuning	17
2.5. List of models available	18
2.5.1. Key Technical Factor of Large language Models	18
2.5.2. OpenAI	19
2.5.2.1. GPT-4.1	19

2.5.2.2. o3 and o4-mini	20
2.5.3. Anthropic	21
2.5.3.1. Claude 4	21
2.5.3.2. Claude 3.7 Sonnet	22
2.5.4. xAI	23
2.5.4.1. grok-3.....	23
2.5.5. Google DeepMind	24
2.5.5.1. Gemini 2.5	24
2.5.6. Meta AI	25
2.5.6.1. Llama 4 and Llama 3	25
2.5.7. DeepSeek.....	27
2.5.7.1. DeepSeek-V3 and DeepSeek-R1	27
2.5.8. Mistral AI.....	28
2.5.8.1. Devstral.....	29
2.5.9. Model Comparisons	30
2.5.9.1. Artificial Analysis Intelligence Index.....	30
2.5.9.2. Pricing	31
2.5.9.3. Output speed	32
2.6. List of Embedding models available	33
2.6.1. Generalities	33
2.6.2. Proprietary models	33
2.6.2.1. OpenAI text-embedding and text-ada-002	33
2.6.3. Open-source models.....	34
2.6.3.1. all-MiniLM-L6-v2	34
2.6.3.2. multilingual-e5-large-instruct.....	34
2.7. Available dataset and information's	35
2.7.1. Publicly available online resources	35
2.7.2. Confidential Dataset of Recycling Points	35
2.8. Available frameworks.....	36
2.8.1. LlamaIndex.....	36
2.8.1.1. Key Capabilities	36
2.8.1.2. Advantages	36
2.8.1.3. Downside	36
2.8.2. Langchain.....	37
2.8.2.1. Key Capabilities	37
2.8.2.2. Advantages	37
2.8.2.3. Downsides	37
2.8.3. OpenAI SDK agent.....	38
2.8.3.1. Key Capabilities	38
2.8.3.2. Advantages	38
2.8.3.3. Downsides	38
2.9. Available technologies	39
2.9.1. Ollama	39
2.9.2. Hugging face	39
2.9.3. Infomaniak IA Tools	40
2.9.3.1. Company Overview.....	40
2.9.3.2. Overview of Infomaniak AI Tools	40
3. Methodology	41
3.1. Selection process.....	41
3.1.1. Language Model Selection.....	41
3.1.2. AI Framework Analysis	42
3.1.2.1. Selection Criteria.....	42
3.1.2.2. Detail Analysis	42
3.1.2.3. Comparative analysis.....	45
3.2. Prototype Implementation	46
3.2.1. Technologies	46
3.2.1.1. LlamaIndex	46

3.2.1.2. Embeddings.....	46
3.2.1.3. Vector Database	46
3.2.1.4. Fronted	46
3.2.1.5. Backend.....	46
3.2.2. General architecture.....	47
3.2.2.1. Presentation layer.....	48
3.2.2.2. Business layer	48
3.2.2.3. Persistence layer	48
3.2.3. File structures.....	48
3.2.4. Configuration files	49
3.2.4.1. Current embeddings	49
3.2.4.2. Files format	49
3.2.4.3. Models	49
3.2.5. RAG System Implementation.....	51
3.2.5.1. Overview.....	51
3.2.6. GIS Integration	52
3.2.6.1. Overview.....	52
3.2.6.2. Process Step breakdown	52
3.2.6.3. Confidentiality	54
3.2.7. Combined tools	54
3.2.8. Embedding management page.....	55
3.2.8.1. Backend.....	55
3.2.9. Index Management pages.....	56
3.2.9.1. Process page.....	56
3.2.9.2. Manage index page	58
3.2.10. Chatbot interface.....	60
3.2.10.1. API endpoints.....	60
3.2.10.2. Backend implementation.....	61
3.2.11. Tracking of token usage	61
3.2.12. Benchmarking Implementation.....	62
3.2.13. Public part	64
3.3. Benchmarking protocol	65
3.3.1. Test questions	65
3.3.2. Test Categories and Actual Questions	65
3.3.2.1. Localization based questions.....	65
3.3.2.2. Informational-Based Questions.....	66
3.3.3. Criteria	67
3.3.3.1. Speed	67
3.3.3.2. Cost	68
3.3.3.3. Accuracy	69
3.4. Benchmarks interpretation	70
3.5. Benchmarks Results	70
3.5.1.1. GIS.....	70
3.5.1.2. RAG.....	70
3.5.2. Criteria analysis.....	71
3.5.2.1. Accuracy	71
3.5.2.2. Speed	72
3.5.2.3. Cost	73
3.5.3. Global analysis of the results	74
3.5.4. Final choice of the model	75
3.6. Prototype Deployment	76
3.6.1. Containerization and Platform Choice	76
3.6.1.1. Docker	76
3.6.1.2. Google Cloud Run	76
3.6.2. Local Configuration Changes.....	77
3.6.2.1. Flask Port Adaptation.....	77
3.6.2.2. Tailwind Installation via NPM.....	77
3.6.3. Deployment with Google Cloud Run.....	78

3.6.4. Build and push the Docker image	78
3.6.4.1. Deploy to Cloud Run	78
3.6.5. Limitation of the current version deployed	79
3.7. Integration to citiwatts	80
3.7.1. Citiwatts	80
3.7.2. Mockup	80
3.7.2.1. User-Interface	80
3.7.2.2. Administration panel	81
3.7.3. Specific adjustment and key point of interest	81
4. Conclusion	82
4.1. Next steps	82
4.1.1. Integration Into citiwatts	82
4.1.2. Keep it up to date	82
5. Project Management	83
5.1. Branching strategies	83
5.2. Sprint 0	83
5.2.1. Sprint 0.1 (May 9 – 15, 2025)	83
5.2.2. Sprint 0.2 (May 16 – 23, 2025)	84
5.2.3. Sprint 0.3 (May 24 – 30, 2025)	84
5.2.4. Sprint 0.4 (May 31 – June 3, 2025)	84
5.3. Sprint 1	85
5.4. Sprint 2	86
5.5. Sprint 3	86
5.6. Sprint 4	87
5.7. End of the project	88
6. Bibliography	89
Annexes	90
Annex I: Complete product backlog	90
Annex II: Time tracking	91
Annex III: Excel file of the detailed benchmarking result	93
7. Author Declaration	95

List of tables and figures and others

Figure 1: Information popup for waste management point.	3
Figure 2: Language Modeling overview	7
Figure 3: Neural Network close up	8
Figure 4: Tokenization example of a text	9
Figure 5: Result number of a text tokenized	9
Figure 6: Embedding example	9
Figure 7: "Attention" process explains	10
Figure 8: RAG schema	12
Figure 9: Agent workflows example	14
Figure 10: Chain of thought example	15
Figure 11: Concrete example of chain of thought prompting	15
Figure 12: The workflow of MCP	16
Figure 13: Pricing of OpenAI GPT-4.1, GPT-4.1-mini, GPT-4.1-nano	19
Figure 14: Pricing of o3 and o4-mini	20
Figure 15: Pricing of Claude Opus 4 and Claude Sonnet 4	21
Figure 16: Pricing of Claude 3.7 Sonnet	22
Figure 17: Pricing of Grok 3 and Grok 3 mini	23
Figure 18: Pricing of Gemini 2.5 Pro and Gemini 2.5 Flash	25
Figure 19: Detailed pricing of DeepSeek-V3 and DeepSeek-R1	28
Figure 20: Pricing of Devstral	29
Figure 21: Artificial Analysis - Artificial Analysis Intelligence Index	30
Figure 22: Artificial Analysis - Pricing: Input and Output Prices	31
Figure 23: Artificial Analysis - Intelligence vs. Price	32
Figure 24: Artificial Analysis - Output pricing	32
Figure 25: Prototype Architecture Diagram	47
Figure 26: File structures	48
Figure 27: Map integration pin popup example	53
Figure 28: Embedding model page	55
Figure 29: Embedding model Information box	55
Figure 30: Screen of the process page	56
Figure 31: Screen of the index page	58
Figure 32: Screenshot of the Chabot admin page	60
Figure 33: Benchmark page choice of model	62
Figure 34: benchmark page GIS and RAG questions	62
Figure 35: Benchmark page general and custom questions	63
Figure 36: Benchmark page result table	63
Figure 37: Screen of a answer of a question from the public chatbot page	64
Figure 38: Map integration response example from public chatbot page with popup opened	64
Figure 39: Total Accuracy score by models	71
Figure 40: Average response time by models	72
Figure 41: Average score for response time by models	72
Figure 42: Average pricing for 1k requests by models	73
Figure 43: Best model comparison	74
Figure 44: Mockup of the chatbot on citiwatts	80
Figure 45: Mockup Admin panel citiwatts	81
Figure 46: Sprint 1 User Stories from Azure DevOps	85
Figure 47: Sprint 2 User Stories from Azure DevOps	86
Figure 48: Sprint 3 User Stories from Azure DevOps	86
Figure 49: Sprint 4 User Stories from Azure DevOps	87

Abbreviation list and specifics words

AI	Artificial Intelligence
API	Application Programming Interface
CDN	Content Delivery Network
CLI	Command Line Interface
CSV	Comma-Separated Values
DB	Database
DOCX	Microsoft Word format
GeoJSON	Geographic JavaScript Object Notation
GIS	Geographic Information System
GPU	Graphics Processing Unit
GPT	Generative Pretrained Transformer
HES-SO	Haute école spécialisée de Suisse occidentale
HTML	HyperText Markup Language
IETF	Internet Engineering Task Force
JSON	JavaScript Object Notation
LLM/LLaMAs	Large Language Model / Meta A
LPD	Loi fédérale sur la protection des données
MCP	Model Context Protocol
MTEB	Massive Text Embedding Benchmark
NLP	Natural Language Processing
NPA	Numéro Postal d'Acheminement
NPM	Node Package Manager
PDF	Portable Document Format
PKL	Python Pickle
PoC	Proof of Concept
QA	Question Answering
QGIS	Quantum Geographic Information System
RAM	Random Access Memory
RAG	Retrieval-Augmented Generation
RL	Reinforcement Learning
SDK	Software Development Kit
SQL	Structured Query Language
URL	Uniform Resource Locator

1. Context

1.1. Introduction

The collection and treatment of waste management are crucial points for the environment and a logistical challenge for the local authorities. With all the different regulations and rules for sorting waste, diverse waste categories, different collection points and varied places to place it, citizens are often faced with a complex system that is not easily accessible. Despite the efforts to digitalize data, relevant information is still scattered across multiple PDFs, websites, or posters at the collection points which makes it tedious and hard for users to find relevant information.

At the same time, new advances in artificial intelligence, particularly in large language models (LLMs) have opened new opportunities for improving the accessibility and accuracy of complex information. These models can understand the questions expressed by users in natural language and generate an adequate response.

More recently, new approaches such as context enhancement, Retrieval Augmented Generation (RAG), and agent tools have significantly improved the abilities of these models to respond accurately to context-specific questions in real time.

In the context of waste management where information is fragmented or in different-format text, GIS data for the position of the waste management plan and these improvements are key advantages. They allow the creation of a complex and accurate chatbot that also not only understands the user's intent but can also *provide accurate information based on up-to-date information.

This bachelor thesis is dedicated to exploring and evaluating different models in order to determine the most suitable model for this task. A second part focuses on the development of a module that encapsulates the chatbot and its GIS capabilities, with the goal of integrating this module on the Citiwatts platform.

The following report is structured as follows: The next section will present to you the state of the art related to waste management, LLM agents, technologies and GIS solutions. This is followed by model selection, context enhancement using RAG and the development of the module that will integrate the chatbot with its GIS capabilities and RAG enhancement. The report then details the implementation within the citiwatts platform, as well as the project management.

1.2. Problematics

Despite the abundance of data available online, accessing relevant information about waste management remains difficult and a complex task for citizens. In most cases, this information is buried in long PDF documents, static web pages, or printed boards at the waste management point. This information is also not always up-to-date and rarely complete on its own. This lack of accessibility discourages proper waste sorting and reduces the effectiveness of environmental care.

Furthermore, it is difficult to know where to dispose of specific types of waste, such as batteries, heavy equipment, paint and other special materials. The diversity of the rules to dispose safely of certain types of dangerous waste and some lack of treatment of waste for certain municipalities create more confusion for citizens.

This situation reveals a clear problem: despite the data existing, it is not presented in a way that is intuitive, interactive, or personalized for the end user. There is a need for a solution that can understand user queries in natural languages, retrieve context-specific data, and guide users clearly, especially for the localization of the correct waste management point.

1.3. Objectives

The main objective of this bachelor thesis is to design, implement and integrate an AI chatbot with GIS and enhance public data to make information personalized, instant and easier to understand for each citizen. To achieve this objective, we will pass through several phases involving research, evaluation, development and integration.

First, a detailed analysis and an evaluation of the different LLMs available on the market will be conducted with a focus on their ability to manage GIS information, confidentiality, and their prompt following capabilities.

Second, the project will make a benchmark of the selected LLMs with a prototype that integrates both RAG and GIS integration in order to see which model will perform the best. This prototype will create a real case to see how the different models react and be able to collect and analyze results.

Thirdly a prototype will be developed and deployed that will encapsulate both the RAG enhancement and GIS integration in order to test all the functionality and provide already a place where citizens can ask questions about waste management.

Finally, we must analyze how it would be possible to integrate the prototype developed into the cititwatts platform and determined what would have been the necessary changes to have a good user experience for the citizen.

2. State of the art

2.1. Recycling-map

Recycling-map (<https://recycling-map.ch/>) is an online platform and mobile application dedicated to help user in Switzerland to easily locate recycling collection points. Recycling-Map.ch is operated by the Igora Cooperative for Aluminum Recycling, which is a member of Swiss Recycle.¹

2.1.1. Key features

This platform has key features that make it easier for users to find information about the location of a waste management plan and the types of waste that can be disposed of. Here is a detailed list:

- Localization-based search: users on the platform can search for a collection point based on their localization or with the postal code of their home.
- Filter of Materials: Users can do research on the platform for a specific type of waste like batteries and the website will only show points that accept batteries.
- Detailed Collection Point Information: For each collection point on the map a dedicated button opens a new interface where the user can get detailed information about the point. This information includes all the types of waste you can dispose of at this point, at the full address of the point, in some cases the phone number of the point, and details about the opening hours or notes.
- Ad-Free: The service is free of advertisements, and user login is optional.
- Mobile App: A mobile app is also available to access the same information but directly on the phone on the user's phone.

Other smaller features like the small popup to quickly see the information related to a point and be able to directly create an itinerary from your position to the plan, are really nice for the user.

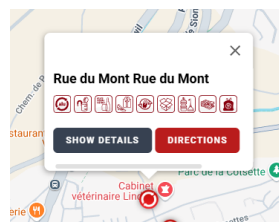


Figure 1: Information popup for waste management point.
Captured on May 2025 from the recycling-map website, <https://recycling-map.ch/en/map>

2.1.2. Lacks and problems.

First the biggest problem with this platform is the mobile application. This mobile application has some big problems. One of them is easy access to the map, which is the main feature of the platform. When the application is opened the map is not available straight away and we have to click on a small zone on the main screen to access the map. The main screen of the app is filled with quizzes and a leaderboard that make it not easy to navigate and find relevant information on the mobile app.

The second biggest problem is it is not possible to see rapidly where to dispose of a specific type of waste if we don't know in which categories it belongs. The user must know what the type of waste is before the research which makes this not very convenient for users. For the map itself, it provides interesting features such as regrouping points in a similar zone in the same box to make it easier to find a place and avoid having a map with thousands of pins all over the place. However, the way this system has been implemented makes it a problem rather than a solution. For instance, to be able to see near points not regrouped on the map we need to be at a zoom level that makes the research not easy at all. Moreover, it also does not change the layout of the map to have, for example, a satellite view of the area to better visualize where the waste management plant is.

¹ Swipe GmbH Zürcherstrasse 310, Ch-8500 Frauenfeld. (n.d.). *Recycling Map*. Recycling Map. Retrieved May 15, 2025, from <https://recycling-map.ch/en/info/general-business-terms>

The platform also has some other minor problems such as it is not possible to research in a zone on the web platform but only with an NPA or a locality. However, some small localities do not have all the necessary equipment to dispose of specific waste and the user can really be confused and think that this type of waste cannot be recycled and dump it in a trash can.

2.2. Swiss recycle.

Swiss Recycle is the Swiss competence center for recycling different type of waste. Its main purpose is to support and promote any recycling advances or efforts make in Switzerland. The website swissrecycle.ch which will be analyzed in this section is Managed by Swiss Recycle.²

2.2.1. Key features

This website has some key features:

- **Extensive information about recoverable substances:** The website provides extensive information about all the different types of waste that can be recycled. For each type of substance, it provides general information about waste, what type of product is accepted for this type of waste and a lot of information and advice about what this waste becomes.
- **Help section dedicated to enterprise:** The platform provides support for companies with a complete toolbox that includes documents, pictograms and more.
- **Events and promotion:** Swiss Recycle promotes initiatives around the recycling theme and provides guides to help citizens sort their waste.

2.2.2. Lacks and problems.

The biggest problem with this platform is it doesn't provide any information about where to dispose of specific waste. There is not a list of available points that does not map to display waste management points. This can, however, be explained as this site's main purpose is not to explain where to dispose of specific waste but to explain what the recoverable substances are.

2.3. Smart waste management in the world

2.3.1. Smart waste management with AI

Recent studies, such as "Smart waste management: A paradigm shift enabled by artificial intelligence" (Olawade et al., 2024) show a shift in waste management thanks to AI. Traditional waste management has some limitations such as logistics problems, lack of waste sorting, and poor recycling efforts. AI introduces new solutions for every step in this process:

- **Smart Bin Systems:** Sensors analyzed by AI algorithms allow real-time monitoring of bin fill levels that enables dynamic collection schedules and minimizes overflows.
- **Route Optimization:** AI can also analyze historical and real-time data such as traffic and waste production patterns to create efficient collection routes to reduce fuel consumption and operational costs.
- **Automated Waste Sorting:** New Multimodal models with vision capabilities replace manual sorting by identifying and classifying different waste types with accuracy, which improves recycling rates and reduces miss sorting problems.
- **Recycling Process Optimization:** AI models can adjust operational parameters like temperature, pressure, composition to increase material recovery, including detecting contamination or defects in recycled products.
- **Demand Forecasting:** AI can anticipate waste generation trends, which can improve the allocation of resources efficiently.

² Swiss Recycling. (n.d.). *Swiss Recycling*. Swiss Recycling. Retrieved May 18, 2025, from <https://swissrecycle.ch/fr/swiss-recycle>

2.3.2. Foster Engaged Community Waste Management

The researcher developed an AI-powered waste management solution to improve waste management and environmental awareness for the Takhian Tia community in Chonburi, Thailand. The solutions created integrate a deep learning image classification model with a LINE-based chatbot to create an educational tool for the residents. (Kulkarineetham et al., 2025)

The model has been trained using labeled images of waste items categorized into four categories. It achieved 91% classification accuracy, that allows to identify reliably waste types from photos. The team also developed user interface and offers multiple functionalities including an image-based recognition, upcycling and recycling tips, quizzes and learning module and a progress tracking system. (Kulkarineetham et al., 2025)

This user interface was tested on 30 primary schools and has given these results:

- 88% average accuracy in waste categorization by the students using the chatbot.
- Big improvement in user's knowledge about waste management

2.4. Concepts

2.4.1. GIS (Geographic Information System)

2.4.1.1. GeoJSON

Definition

GeoJSON is a file format for representing geospatial data, based on the standard JSON Format. It is a popular way of sending and receiving points, line or polygon in a text format. It has become standard to exchange Geospatial data over the web, and it is maintained by the Internet Engineering Task Force (IETF).³

In the context of our bachelor's, the geolocation data provided about waste management plants that are confidential are encoded into GeoJSON format.

Format

The basic structure of a GeoJSON file is a JSON object. This JSON object includes first a type of field for either a "geometry" or "features" field. This type allows to determine if this is a single geometry object or a "FeatureCollection". Each geometry includes a coordinate that specifies its position using the WGS84 format containing longitude and latitude values. Features can also contain properties that object to storing additional information about the geographic element. These properties are not imposed with a specific format.³

```
{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [7.0741, 46.1928]
  },
  "properties": {
    "name": "Sion"
  }
}
```

³ Butler, H., Daly, M., Doyle, A., Gillies, S., Schaub, T., & Hagen, S. (2016). *The GeoJSON Format* (Request for Comments No. RFC 7946). Internet Engineering Task Force. <https://datatracker.ietf.org/doc/rfc7946/>

2.4.1.2. QGIS

Description

QGIS (Quantum Geographic Information System) is an open source, geographical information system (GIS) that has become a widely used tool for geospatial analysis, and cartographic production. It is developed by the community, and it offers a wide range of vector, raster and database format like GeoJSON. It also offers an extensive plugin platform where people in the community can either use the plugins or create a new one specialized for a specific task.⁴

InteliGeo

InteliGeo is a plugin developed for QGIS software that facilitates interaction with LLMs. It offers a chatbot directly integrated into QGIS that allows users to ask questions about the current project. This solution has been developed using Python and Langchain to manage the agent tool and workflow.⁵

In concrete terms, the InteliGeo agent has a tool that can retrieve information on the different layers available on the projects and retrieve their relative information. However, it cannot access to the content of one layer. This is one of the main limitations we have found when we have tested the chatbot.⁵

There are also some other limitations such as it is not possible to set a local model on the configuration of the chat conversation, and it is also not possible to choose your model and you can only chose some model of OpenAI. For instance, right now new models dedicated to agentic apps are being released and they are more cost-effective than previous versions.

2.4.1.3. Online visualization tool for GeoJSON

On the internet there are multiple visualization tools to show GeoJSON. One of them is <https://geojson.io>, which allows you with a minimal interface to visualize some GeoJSON rapidly. This is useful to test the functionalities of GeoJSON and to use it to previsualize the structure of one of them.

2.4.1.4. Leaflet

Leaflet is a lightweight, open-source JavaScript library that creates interactive maps. It has been designed for simplicity and performance. Leaflet support GeoJSON format and allow us to add markers, popups, line and more. Leaflet supports multiple providers of maps including OpenStreetMap.⁶

2.4.1.5. Folium

Folium is a python library that manipulates interactive leaflet maps easily. It has been created to bridge data manipulation with interactive map.⁷

It allows users to connect complex structures like Pandas objects to geographical layers for dynamic visual output. Folium supports a lot of open-source map providers. It supports GeoJSON format and their relative objects like marker tooltips and more.⁷

⁴ qgis/QGIS. (2025). QGIS. <https://github.com/qgis/QGIS> (Original work published 2011)

⁵ Farnaghi, M. (2025). MahdiFarnaghi/intelli_geo. https://github.com/MahdiFarnaghi/intelli_geo (Original work published 2024)

⁶ Leaflet — an open-source JavaScript library for interactive maps. (n.d.). Retrieved June 4, 2025, from <https://leafletjs.com/>

⁷ Folium — Folium 0.19.7 documentation. (n.d.). Retrieved June 4, 2025, from <https://python-visualization.github.io/folium/latest/>

2.4.2. LLMs (Large Language Models)

2.4.2.1. History

Language modeling is a foundational concept in natural language processing (NLP), that has goals to understand and predict language patterns. This last decade has been full of breakthroughs in this domain. This reflects new technological progress and understanding of linguistics. Here below is an overview of the language modeling map. (Hadi et al., n.d.)

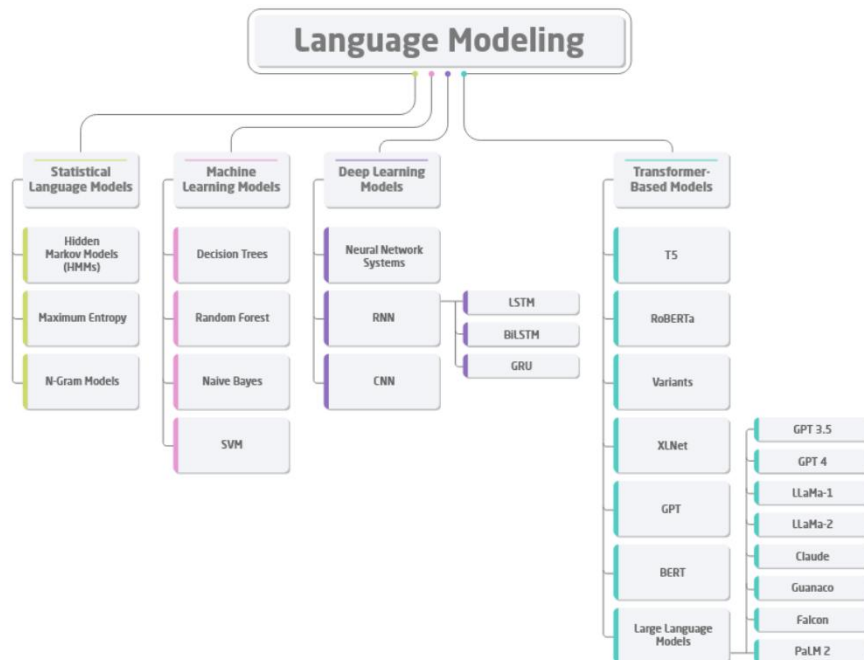


Figure 2: Language Modeling overview

Retrieved from Large Language Models: A Comprehensive Survey of its Applications, Challenges, Limitations, and Future Prospects <https://doi.org/10.36227/techrxiv.23589741.v3>

Language models started with statistical and classic machine learning approaches that understood the basic word pattern but had difficulties in sequence understanding. Advances in the deep learning field, especially Transformer architecture, allow models to process language with global context and task parallelization. Today Large language models build on these advances and on enormous datasets, allowing us to have a better understanding of the natural language. (Hadi et al., n.d.)

2.4.2.2. Definitions

LLMs stand for large language models, which is an AI (Artificial Intelligence) model that has been trained on very large and various datasets.⁸ This makes the model capable of analyzing, processing and generating an answer to a given query. This answer will try to mimic the natural languages of humans. (Mohaimenul Azam Khan et al., 2024)

LLMs are deep learning algorithms that use different architectures such as neural networks or Generative Pretrain transformers ([GPT](#)) to analyze and generate text.

⁸ Marsoni, D., Pitteloud, M., Fernández Rodríguez, Z., & Marques Conde, C. (2025, April 15). Basic Concepts. Retrieved May 24, 2025, from Llog Wiki. <https://github.com/davidmarsoni/Llog/wiki/Basic-Concepts>

2.4.2.3. Neural networks

A neural network is a machine learning algorithm that makes decisions similar to the human brain by using nodes and layers and weights.⁹ The following diagram shows a basics implementation of a neural network.

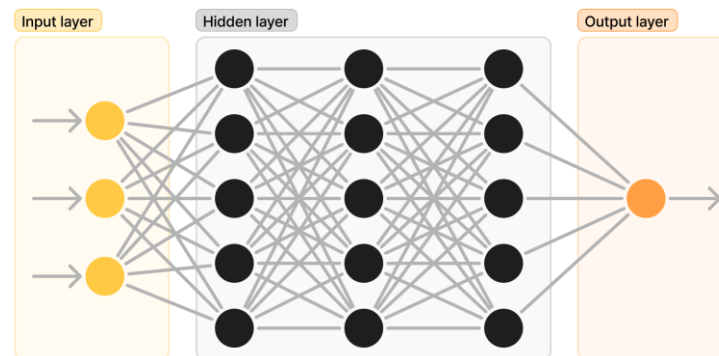


Figure 3: Neural Network close up

Author's source, originally created for a project at the HES-SO, <https://github.com/davidmarsoni/Llog/wiki/Basic-Concepts>

“Neural networks learn by adjusting the weights of connections (determining their importance) between neurons during training to refine their predictions. Thus, it learns efficiently and thereby minimizes the errors between its predictions and actual data.”^{8,10}

“These kinds of AI are well suited for small tasks such as images guessing or simply playing a game.”¹⁰ A lot of videos have been created demonstrating neural network models trying to beat the original Donkey Kong or create a model to learn how to drive. Here below are 3 videos on this topic:

- <https://youtu.be/v3UBIEJDXR0> “AI Agent Learns to Escape (deep reinforcement learning)”
- https://youtu.be/r428O_CMcpI “A.I. Learns to DRIVE”
- <https://youtu.be/DmQ4Dqxs0HI> “AI Learns to play JUMP KING”

“Opposed to the deterministic output of games or image guessing where a result is either good or bad, language is much more fluid. And while multiple movement inputs are easy to map, parameters such as context, goal, clarity, language register, use of idioms, etc..., are much harder to list and classify.”¹⁰

Instead, we will use what is commonly called a “transformer”.

2.4.2.4. Generative AI

“Generative AI refers to algorithms that are capable of generating new content, such as text, images, video, and more based on existing data that they have learned. LLMs are increasingly becoming more and more multimodal, which allows the user to upload an image and receive responses in different formats. e.g., if a user uploads an image about a specific schema, the LLM will be able to understand the image and respond to a coherent response to the user.”^{8,11}

⁹ IBM. (2021, October 6). *What is a Neural Network?* <https://www.ibm.com/think/topics/neural-networks>

¹⁰ Marsoni David, Gloor Gian-Luca, Mathias Pitteloud, & Salamin Alexandre. (2023). ChatGPT Report, It Security project.

¹¹ *What is generative AI?* (2021, February 9). IBM Research. <https://research.ibm.com/blog/what-is-generative-AI>

2.4.2.5. Transformers, Tokens and embeddings

To make languages manageable for an AI model we need to transform it into small quantifiable and countable blocks. To do that we will use the concept of token. It consists of breaking words into simple bricks that are represented with numbers.^{10,12}

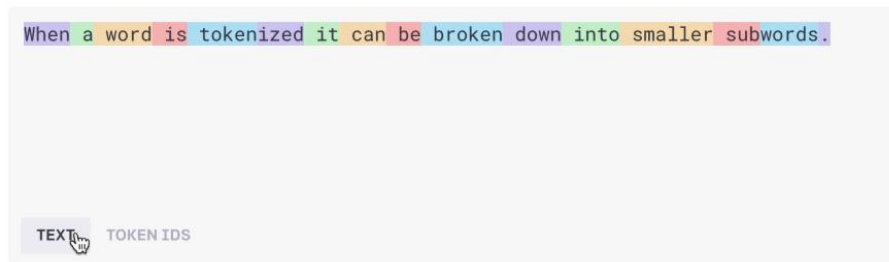


Figure 4: Tokenization example of a text

Captured on May 2025, How ChatGPT works, <https://typefully.com/DanHollick/how-chatgpt-works-a-deep-dive-yA3ppZC>

[2215, 257, 1573, 318, 11241, 1143, 340, 460, 307, 5445, 866, 656, 4833, 850, 10879, 13]



Figure 5: Result number of a text tokenized

Captured on May 2025, How ChatGPT works, <https://typefully.com/DanHollick/how-chatgpt-works-a-deep-dive-yA3ppZC>

"We can see that word like "tokenized" is broken into "token" and "ized". Same for "subwords" which become "sub"- "words". Each "bricks" are then assigned an ID."¹⁰

After this step, we store these into a multi-dimensional vector that indicates the relationship between certain words by using the distance between each other. For example, "dog" and "puppy" need to be at the same distance as "cat" and "kitten." It is called "embedding".^{10,12}

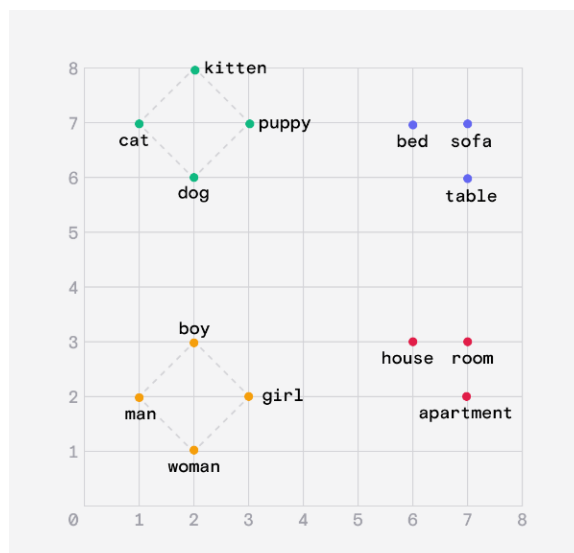


Figure 6: Embedding example

Captured on May 2025, How ChatGPT works, <https://typefully.com/DanHollick/how-chatgpt-works-a-deep-dive-yA3ppZC>

¹² Hollick, D. (2023, April 13). How ChatGPT works: a deep dive | Dan Hollick. Retrieved May 24, 2025, from Typefully. <https://typefully.com/DanHollick/yA3ppZC>

The next step needed call “attention” add even more context to embeddings. Attention is essential in natural language processing. Because embedding struggles to capture words with different meanings. For example, word “bank” is ambiguous without context, meaning both the edge of a river and a place where money is stored. “Attention” resolves this ambiguity by looking at the overall context. The process will look back for words inside the given context and then adjust the weight toward it. ^{10,12}

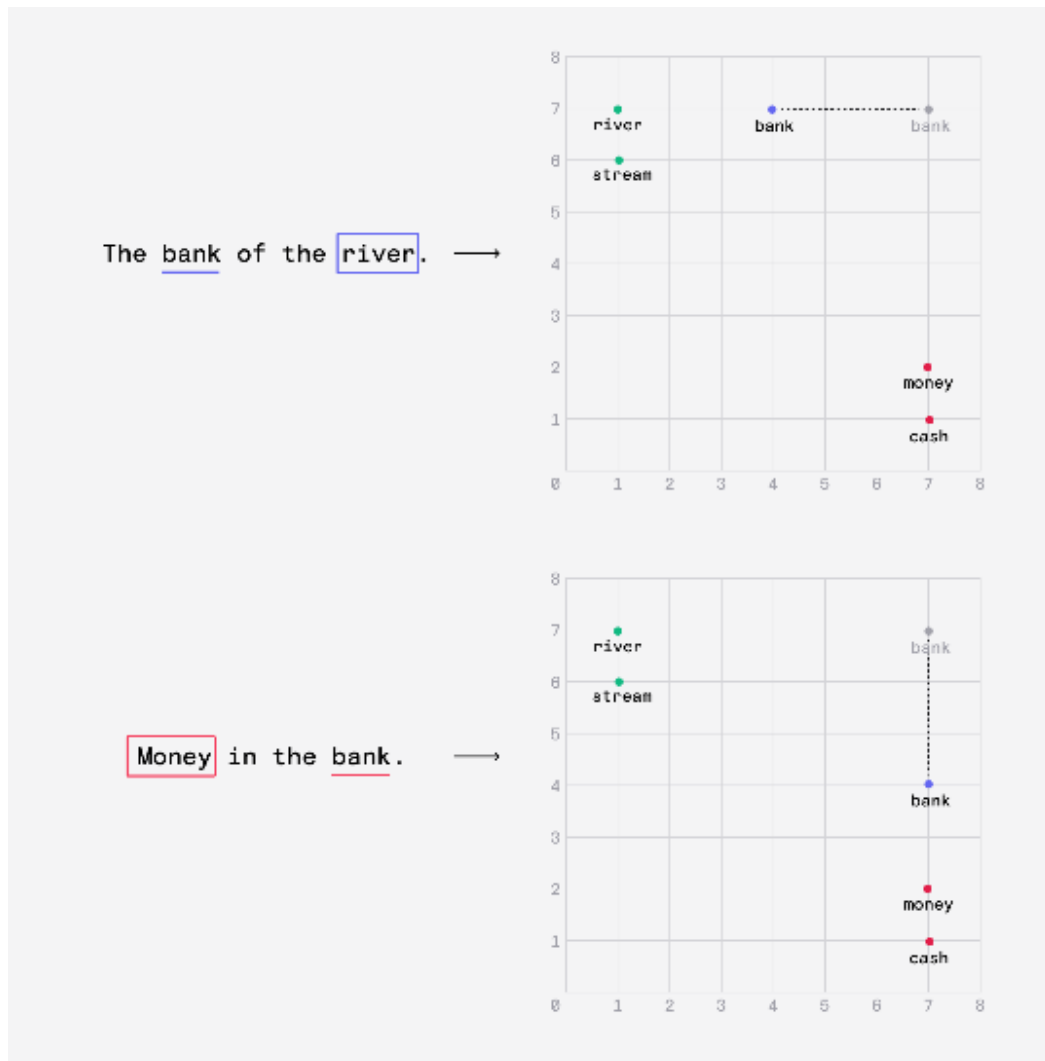


Figure 7: "Attention" process explains

Captured on May 2025, How ChatGPT works, <https://typefully.com/DanHollick/how-chatgpt-works-a-deep-dive-yA3ppZC>

This process will be repeated many times to be able to capture all the context linked to a sentence. The resulting embedding (which has been weighted by the attention process) is then fed into a neural network which will predict the next word of the sentence. ^{10,12}

2.4.2.6. GPT (Generative Pre-trained Transformers)

Generative Pre-trained Transformers (GPT) is a model developed by OpenAI to understand and generate natural language. GPT models are based on the transformer's architecture, with has been explained in more detail in the upper section about transformer architecture tokens and embeddings. ¹³

This new model architecture has accelerated generative AI and natural language generation. This architecture has also created a new boom in this field with multiple companies like Google and Anthropic to building their IA models with either this GPT architecture or a completely custom one. ¹³

¹³ *What is generative AI?* (2021, February 9). IBM Research. <https://research.ibm.com/blog/what-is-generative-AI>

2.4.3. RAG (Retrieval-Augmented Generation)

2.4.3.1. RAG Description

“Retrieval-Augmented Generation (RAG) is an approach that solves some of the challenges of LLMs.”^{8,14}

One of the main limitations of LLMs is their lack of access to a user's specific data. Most LLMs are typically trained in a large variety of publicly available information, enabling them to handle a lot of situations when responding to user questions. However, once a model is trained, Most LLMs remain static as they don't have the ability to access new data beyond their training data. As a result, they may provide outdated answers or even generate completely inaccurate answers when queried about topics that are not in their training data.^{8,15}

Other challenges that LLMs face are that they often produce irrelevant or imprecise results when they try to answer some specialized domain questions. To be able to give relevant answers, the model needs to understand the specific domain or organization. To do that you can provide the LLM with company-related or subject-related data. For example, in our bachelor thesis we will need a way to let the chatbot respond with high accuracy to questions related to waste management.^{8,14}

“The common solution to these problems is to provide the custom data for the question as part of AI's prompt. This is called Retrieval-Augmented Generation (RAG), as you retrieve data from your own sources to augment the capabilities of the LLMs.”^{8,14}

2.4.3.2. Benefits

“The main benefits of using RAG are the following:”⁸

- **“Reduced hallucinations:** RAG allows for controlled information retrieval, which is finely tuned with generated content to maintain the best coherence and avoid hallucination. Hallucinations can be especially problematic in sensitive domains such as healthcare or law, where incorrect information can have serious consequences.”^{8,14}
- **“Up-to-date results:** The RAG approach relies on an external knowledge base to retrieve relevant information for the user. These bases can be updated all the time rather than being limited to the specific training the LLM has received. This allows the response to no longer be limited to the model's current knowledge.”^{8,14}
- **“Better context relevance:** As we can rely on external specific resources like PDF databases or files to enhance our context, it reduces the LLM's lack of context in specific context.”^{8,14}

¹⁴ Databricks. (n.d.). What is Retrieval Augmented Generation (RAG)? | Databricks. Retrieved May 22, 2025, from <https://www.databricks.com/glossary/retrieval-augmented-generation-rag>

¹⁵ Shone, O. (2025, April 25). 5 key features and benefits of retrieval augmented generation (RAG). Retrieved May 24, 2025, from The Microsoft Cloud Blog. <https://www.microsoft.com/en-us/microsoft-cloud/blog/2025/02/13/5-key-features-and-benefits-of-retrieval-augmented-generation-rag/>

2.4.3.3. RAG explains with a schema

RAG can be a bit hard to represent without a proper schema, that way the best way to represent and explain RAG is with a schema.

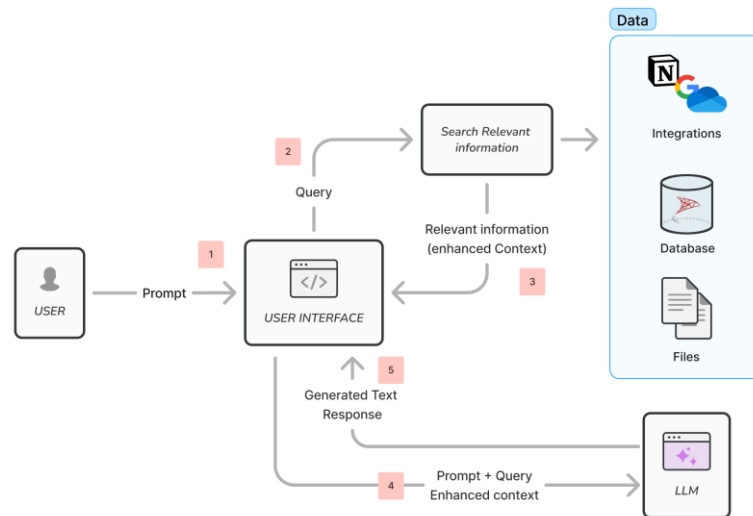


Figure 8: RAG schema

Author's source, originally created for a project at the HES-SO, <https://github.com/davidmarsoni/Llog/wiki/Basic-Concepts>, Inspired by a AWS schema of the <https://aws.amazon.com/what-is/retrieval-augmented-generation/>

Schema description

1. "First, the user asks a prompt (question) to the user interface."⁸
2. "Next, this transformed question into a user prompt optimized to retrieve relevant information (query) will be used to search for relevant information using the data it has available."⁸ This information can include database file integration and more.
3. After that, the relevant information via [embedding](#) or other methods (enhanced context) will be returned to the user interface.⁸
4. This enhanced context will then serve as a base and a context to be able to generate a response by our LLM.⁸
5. "Finally, the generated enhanced response is shown to the user."⁸

2.4.3.4. Spatial-RAG

Spatial-Rag (Spatial Retrieval-Augmented Generation) is a newly developed framework designed to enhance capabilities of a LLMs to understand and work with spatial interaction or data with natural languages. This approach helps to solve the limitations of LLMs to work with complex spatial queries by combining spatial databases with some semantic analysis. (Yu et al., 2025)

This framework combines two retrieval mechanisms:

- **Sparse Spatial Retrieval:** It uses spatial SQL queries to extract relevant information such as point polygons from a spatial database based on user queries.
- **Dense Semantic Retrieval:** It uses a ranking system based on the semantic relevance to extract relevant information.

This framework employs a hybrid optimization strategy that integrates the 2 retrieval mechanisms. Finally, an LLM-based step creates a coherent answer for the user based on all the context available.

It has been tested on real-world tourism data (OpenStreetMap + TripAdvisor) and the results show that the Spatial-RAG significantly outperforms RAG and QA models, particularly in tasks with spatial constraint, such as nearby recommendations. This approach makes a significant step forward in spatial LLMs capabilities (Yu et al., 2025)

2.4.3.5. Vector Database

A vector database is a special kind of database used to store embeddings. Embeddings are numerical representations of data like text, images or files. To have more information about embeddings see this section: 2.4.2.5 Transformer, Token, Embeddings.^{16*17}

Unlike relying on an exact keyword like a traditional database, a vector database will use the approximate neighbor algorithm (ANN) to find the vectors that are the most similar to the query. This can be done by using cosine similarity or the Euclidean distance metrics.^{16*17}

Vector databases are often used on AI projects with RAG systems because these vector databases are an more efficient way to store and use vectors for retrieval purposes.

ChromaDB

ChromaDB is an open-source vector database built to work with embeddings. It is primarily used for retrieval IA applications. It is really well suited for RAG workflows where we need to use embedding to retrieve information on documents.¹⁸

2.4.3.6. Python pickle files (PKL)

A PKL (Pickle) file is a binary files format used in Python to serialize and store Python object. It is commonly used in the machine learning field to store and save complex python objects to avoid the need to recompute or retrain a model. It can also be used as a simple way to store the precomputed embedding of files.¹⁹

¹⁶ Holdsworth, J., & Kosinski, M. (2024, July 29). What Is A Vector Database? | IBM. <https://www.ibm.com/think/topics/vector-database>

¹⁷ What is a vector database? (n.d.). Cloudflare. Retrieved June 6, 2025, from <https://www.cloudflare.com/learning/ai/what-is-vector-database/>

¹⁸ Chroma. (n.d.). Chroma. Retrieved June 6, 2025, from <https://trychroma.com>

¹⁹ pickle — Python object serialization. (n.d.). Python Documentation. Retrieved June 6, 2025, from <https://docs.python.org/3/library/pickle.html>

2.4.4. Agents

2.4.4.1. Definition

AI agents are autonomous systems that use AI to resolve goals and complete tasks instead of users without the need of the users who do have direct interaction with the system. These agents have shown reasoning, planning, and memory management capabilities and have a certain level of autonomy that allows agents to make decisions and adapt to a situation.²⁰

One of the main strengths of modern AI agents is their multimodality handling, such as text, images and files. This multimodality allows agents to interpret complex queries more effectively. They can select the tool they want for each task, which will generate a more accurate and elaborate answer. For instance, a multimodal agent can analyze an image along with some text and a file, choose the right tools for each type of data, and provide a comprehensive response.²⁰

Here below is an example of an agent that generates an image on a Telegram channel. This workflow is part of a guide on the platform n8n that allows the creation of agents graphically. This workflow is composed of a memory section that is a database and 2 tools to first generate the image and then send it back on the Telegram channel. For more information about this particular workflow, follow this link: <https://blog.n8n.io/ai-agentic-workflows/>.

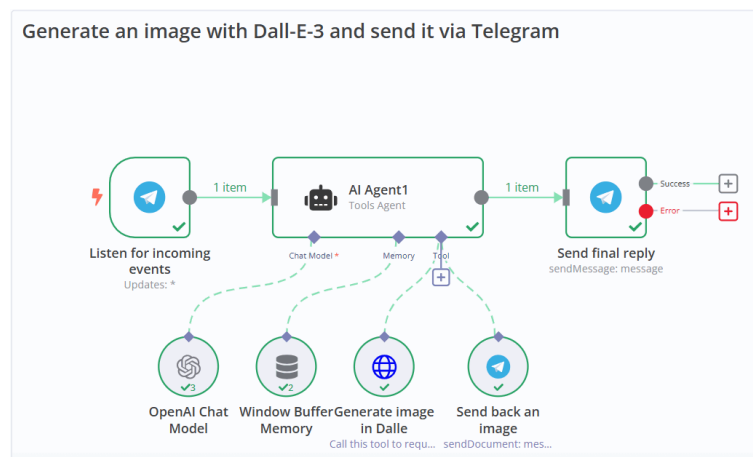


Figure 9: Agent workflows example

Captured on May 2025, on the Agentic AI practical guide of n8n, <https://blog.n8n.io/ai-agentic-workflows/>

2.4.4.2. Key features

AI agents have some main features that enable them to be more autonomous and flexible in complex environments. Here below are the key features:

- **Autonomy:** AI agents can perform tasks and make some decisions by themselves without the need for the intervention of a user in the process. This independence allows us to handle effectively unpredicted situations.²⁰
- **Reactivity:** The agent can interpret the environment and respond in real-time to change or new information. This enables them to adapt their actions based on the current situation. For example, if an agent cannot retrieve information for generating a response, it will automatically say that he was not able to get the information in his output response.²⁰
- **Planning:** Some agents can identify the necessary steps to perform a task by analyzing the requests. Then with this plan in mind, execute the steps in order producing a more coherent answer.²⁰
- **Collaboration:** AI agents can collaborate with humans or other agents to achieve a common goal. This is especially important for complex queries requiring multiple steps.²⁰

²⁰ What are AI agents? Definition, examples, and types. (n.d.). Google Cloud. Retrieved May 29, 2025, from <https://cloud.google.com/discover/what-are-ai-agents>

2.4.4.3. Workflow

Workflow is a sequence of steps or tasks that are required to complete a specific task. In our context, workflow really refers to a process that is done by one or multiple agents that are participating in the workflows. Workflow helps to stay organized, efficient, and consistent by defining who to do what, when and how. A workflow can be manual but is often fully automated.²¹

2.4.4.4. Chain of thought

Chain of Thought (CoT) categorized as a prompt engineering technique is a process that enhances the output of LLMs particularly in complex tasks. It facilitates the query understanding and solving with a step-by-step reasoning process. It is often used to refine the user prompt to get the most accurate information to answer the query.²²

Here below is a graphical representation of the chain of thought compared to a basic prompt chaining.

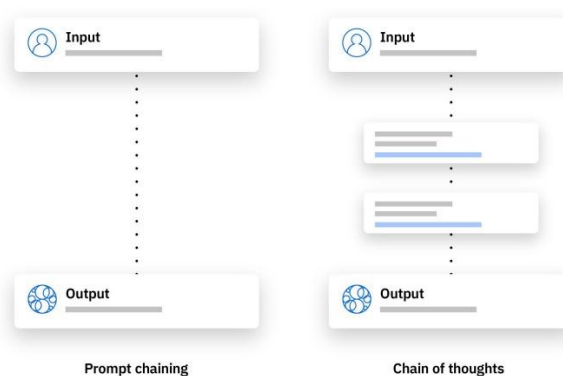


Figure 10: Chain of thought example

Captured on May 2025, IBM chain-of-thought, <https://www.ibm.com/think/topics/chain-of-thoughts>

CoT prompting has shown very great results compared to prompt chaining with a significant upgrade in the accuracy of the answer (Wei et al., 2023). Here below is a concrete example of a chain of thought prompting and their results. Now new models now directly generate a chain of thought during their process without the need of CoT prompting.

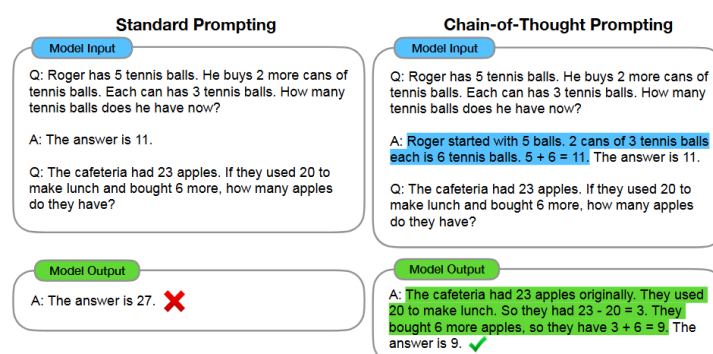


Figure 11: Concrete example of chain of thought prompting

Captured on May 2025, Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, <https://arxiv.org/pdf/2201.11903>

²¹ AI Workflow | IBM. (2024, November 11). <https://www.ibm.com/think/topics/ai-workflow>

²² Gadesha, V., Kavlakoglu, E., & Winland, V. (2025, April 23). What is chain of thought (CoT) prompting? | IBM. <https://www.ibm.com/think/topics/chain-of-thoughts>

A recent study by Google DeepMind shows that explicit CoT prompting is not always needed. Instead, we can achieve similar results by changing the decoding methods, using the top-k sampling that greedy decoding uses. This results in the creation of a natural reasoning path in the model output. This really means that the model is now able to reason by itself without the need for additional prompting techniques. For instance, new reasoning models use this base to create a complete chain of thought and augment the accuracy of their answers. (Xuezhi & Zhou, 2024)

2.4.4.5. Prompt engineering

Prompt engineering is a method that consists of creating precise and effective prompts to guide the LLM system in producing more accurate and relevant answers. All models based their responses on the prompt of the user, so using a well-formatted and structured prompt helps the AI to better understand the intent, context, and slight particularities of the questions.²³

Recently the system prompt of Claude 4 has been leaked and revealed around 10,000 words of instruction, advice, and queries. This really shows how important and complex prompt engineering can be. Claude 4 system prompts extensive information for safety purposes, like legal ages of multiple countries.²⁴ To see the entire Claude 4 system prompt, you can follow this link: [https://github.com/elderplinius/CL4R1T4S/blob/main/ANTHROPIC/Claude 4.txt](https://github.com/elderplinius/CL4R1T4S/blob/main/ANTHROPIC/Claude%204.txt)

MCP (Model Context Protocol)

The Model Context Protocol (MCP) is a standardized interface designed to connect AI assistants with external systems such as APIs, databases, and software tools. It defines a common structured way to give contextual data and functions to large language models (LLMs). This enhanced LLMs ability to reason, and act based on real information. (Hou et al., 2025)

MCP was originally developed by Anthropic to address the growing need for LLMs to interact with external sources in a consistent and extensible manner. Instead of relying on custom integrations for each external implementation, MCP introduces a unified approach to exposing external capabilities to AI systems in a machine-readable format. (Hou et al., 2025)

One of the main strengths of MCP rely on its ability to make external tools “visible” to the model. This allows the AI assistant to reason about which tools to use at which time and interpret the returned data. All of this is a coherent, standardized structure. Anthropic have open-sourced MCP, to encourage its adoption of this standard by AI researchers and the community. (Hou et al., 2025)

Here below is a graphical representation of an MCP workflow where you can clearly see that MCP provides a unified way to build tools that interact with the LLMs. This was the LLM’s ability to select the right tool autonomously.

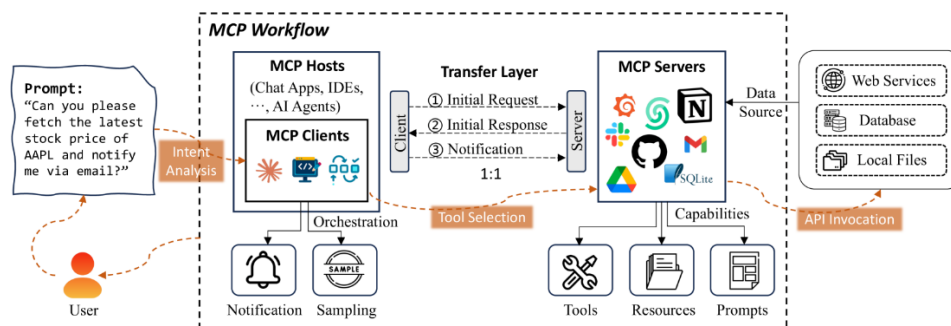


Figure 12: The workflow of MCP

Retrieved from: Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions, section 3.1, <https://www.doi.org/10.48550/arXiv.2503.23278>

²³ Bergaman, D. (2023, November 27). What Is Prompt Engineering? | IBM. <https://www.ibm.com/think/topics/prompt-engineering>

²⁴ Willison, S. (2025, May 25). Highlights from the Claude 4 system prompt. Simon Willison's Weblog. <https://simonwillison.net/2025/May/25/claude-4-system-prompt/>

2.4.5. Fine Tuning

Fine tuning is a machine learning algorithm used to adapt an LLM to a specific task or improve its ability to a specific domain. It is a form of transfer of learning with an existing model trained with a large dataset that is refined (fine-tuned) with a smaller specific dataset. This approach helps to improve model performance for specialized use cases and avoid the need to create a completely new model from scratch.²⁵

Fine-tuning is especially valuable with LLMs as it allows them to adjust to the new specific domain quite easily. It is commonly used in applications with customizable conversational AI, refined image generation, or the use of a model in a particular sector.²⁵

²⁵ What is Fine-Tuning? | IBM. (2024, March 15). <https://www.ibm.com/think/topics/fine-tuning>

2.5. List of models available

The following section focused on the most recent and widely available models. We made this choice to ensure relevance and to be up-to-date with models that are constantly evolving. Most recent models have more capabilities, such as multimodality, a deeper understanding of complex queries, and built-in tools like web research, and often come with lower operational costs. To keep a certain balance between the paid and open-source models, we have also included some state-of-the-art local models to be able to compare closed-source models and open-source models in our context of waste management.

A crucial feature of our future application is the management of geographical data and the multilingual aspect of the chatbot. These two points will influence the choice of models that we will analyze more deeply.

Ever since these models started constantly evolving with new models announced practically every week this section has been becoming outdated relatively quickly. Moreover, scientific documentation and transparent performance data are nearly impossible to find for many of the closed-source models, which limits our analysis. Please note that the model presented below was announced before June 2025, models presented after this have not been included. However, information like pricing and key features of those models has been updated until the end of the project.

Previous generation models have also already been analyzed and explained in detail by Abdullah Binjos in his bachelor's thesis, "Integration of Artificial intelligence for the exploration of energy transition solutions" which is closely related to the theme of my bachelor's thesis for the technical part. For detailed information on those earlier models, please refer to his work, in particular Section 2.1. In short, the analysis focuses on the most advanced model from 2023, which included GPT-4, GPT-3.5, LLaMA, OpenLLaMA, and others. Each model was analyzed with first a small description of the model followed by an overview of its capabilities and limitations. (Binjos, 2023)

During our analysis, we will keep the general format used by Abdullah Binjos but we will include more information about the new features and capabilities of these models regarding their pricing and practical information.

Most of the descriptive content is based on the information on the official websites of the respective models. As these sources can induce some positive bias or marketing overstatement, we added a last section with a third-party comparative analysis, offering a more objective vision of real-world model performance.

2.5.1. Key Technical Factor of Large language Models

LLMs are complex architectures where performance depends on multiple technical parameters. One of these parameters is the context length or context windows that represent the amount of text the model can process at once. On the other hand, the number of parameters practically determines the model's size and complexity. These 2 parameters are especially important.

Context length determines how much information the model can process on a single path. Models with longer context windows are better for understanding and generating a more coherent response based on the context and conversation provided. This can especially be relevant for the historical context of the chatbot during a conversation. Many researchers have published scientific papers on how to extend context windows with various methods. For instance, SelfExtend is a method that extends LLMs' context windows by combining grouped and neighbor attention during inference, enabling long-context understanding without fine-tuning. This greatly improves the size of the context that can be given to LLMs. (Kaplan et al., 2020)

The number of parameters reflects the scale of understanding of a model and its capacity to learn language patterns. Generally, larger models with more parameters tend to have better performance on complex tasks including multilingual processing and reasoning. However, these larger models need more computational resources and that can induce more costs. (Kaplan et al., 2020)

Other parameters such as the diversity of the data to train the model the built-in multimodality and the costs of the models are also really important points.

2.5.2. OpenAI

OpenAI, which is one of the most well-known AI companies in the world with its product ChatGPT, is an AI research company focused on ensuring that artificial general intelligence benefits all humanity. Its economic model is a capped-profit company, limiting investor return to prioritizing safety, ethics, and broad accessibility.²⁶ They are currently looking to change their economic model to profit company.²⁷

2.5.2.1. GPT-4.1

Description

OpenAI GPT-4.1 model is the new flagship LLM model from OpenAI, introduced on April 14, 2025. It offers a significant improvement over previous models, offering better capabilities in coding, instruction following, and long-context comprehension. With its new context that has been upgraded to 1 million tokens (corresponding to around 2,500 pages) and a newer context cutoff of June 2024, GPT-4.1 is well-suited for more complex tasks.²⁸

This model has also declined into 3 major versions that serve different purposes. First the Full version GPT-4.1 serves the most complex tasks or queries that need to have a powerful model. Next, GPT-4.1-mini is a balanced version between speed and accuracy that allows the model to respond more quickly and for a fraction of the base price. Finally, GPT-4.1-nano prioritizes response speed while maintaining a performance level close to the higher versions. It is also the most cost-effective option. For instance, the nano version costs twenty times less than the base version.²⁹

Pricing

Here below is the detailed pricing table for all the models discussed in the previous part. Here you can clearly see the differences between the various declinations of the model.

Model (Prices are per 1M tokens)	Input	Cached input	Output	Blended Pricing*
gpt-4.1	\$2.00	\$0.50	\$8.00	\$1.84
gpt-4.1-mini	\$0.40	\$0.10	\$1.60	\$0.42
gpt-4.1-nano	\$0.10	\$0.025	\$0.40	\$0.12

*Based on typical input/output and cache ratios.

These models are available for use in our [Batch API](#) at an additional 50% pricing discount.

Figure 13: Pricing of OpenAI GPT-4.1, GPT-4.1-mini, GPT-4.1-nano
Captured on May 2025 from the OpenAI website, <https://openai.com/index/gpt-4-1/#pricing>

Capabilities

- **Multimodal Input:** This model allows multimodal input such as plain text, PDFs, and various format documents, images, and more. This can really be an important point to transmit our data to the model easily with a file.
- **Extended context windows:** With its new enhanced context windows, this new model can now take better advantage of the RAG approach and has also received improvement in its context usages.
- **Better coding performance:** This new model has received a significant upgrade in coding performance that has raised its score in SWE-bench from 33.2% to 54.6%.²⁹

²⁶ OpenAI. (n.d.). About. Retrieved May 20, 2025, from OpenAI. <https://openai.com/about/>

²⁷ OpenAI. (n.d.). Evolving OpenAI's structure. Retrieved May 2025, 20, from <https://openai.com/index/evolving-our-structure/>

²⁸ OpenAI. (14, April 2025). Introducing GPT-4.1 in the API. Retrieved May 2025, 19, from OpenAI. <https://openai.com/index/gpt-4-1/>

²⁹ OpenAI. (n.d.). GPT-4.1. Retrieved May 19, 2025, from OpenAI Documentation. <https://platform.openai.com/docs/models/gpt-4.1>

- **Improved Instruction Following:** This newer model has been trained to better follow the instruction set in its context windows.
- **Tools Integration:** These newer models have been built in parallel with the new OpenAI SDK agent that allows agents to use custom tools. The following section talks about the [OpenAI SDK agent](#).

If you want to learn more in detail about the different scoring of this IA compared to the last-generation model, please look at <https://openai.com/index/gpt-4-1/> in particular the Appendix section.

Limitations

- **Slowness of the answer:** depending on the complexity of the request and the choice of the full model the answer can take up a few seconds to be generated.
- **Platform dependent:** This model is closed source and only accessible with the OpenAI API, and OpenAI has already completely altered their API or rapidly deprecated a model.³⁰
- **Knowledge Cutoff:** While not a critical issue for our current application, it is important to note that this model cannot handle responding with actual up-to-date information, which may limit its use in dynamic or current-event-related queries.

2.5.2.2. o3 and o4-mini

Description

OpenAI o3 and o4-mini on April 16, 2025, represent a significant advancement in AI reasoning capabilities. These models are designed to “think for longer before responding”, and allow them to better understand complex, multi-step problems across various domains³¹.

The o3 model excels in areas such as coding, mathematics, science, and visual perception. The most important improvements of this model are the different scoring on platforms like Codeforces, SWE-bench, and MMMU. It has strong capabilities at handling complex queries that require multi-step analysis where the answers are not obvious at all. In evaluations, o3 makes 20% fewer major errors than its previous model, o1, especially in real-world tasks related to programming, business consulting, and creative ideation³¹.

On the other hand, the o4-mini model is a smaller, more cost-efficient model optimized for speed. Despite its size, it achieves remarkable performance in mathematics and coding. Given a Python interpreter, this model achieves a 99.5% pass rate on AIME 2025, highlighting its ease in mathematical problem-solving³¹.

Pricing

The pricing of o3 and o4-mini is higher than the non-reasoning model, such as gpt-4.1. Here below is the detailed pricing for these models.

OpenAI o3 Our most powerful reasoning model with leading performance on coding, math, science, and vision Price Input: \$2.00 / 1M tokens Cached input: \$0.50 / 1M tokens Output: \$8.00 / 1M tokens	OpenAI o4-mini Our faster, cost-efficient reasoning model delivering strong performance on math, coding and vision Price Input: \$1.100 / 1M tokens Cached input: \$0.275 / 1M tokens Output: \$4.400 / 1M tokens
---	---

Figure 14: Pricing of o3 and o4-mini

Captured on June 2025 from OpenAI pricing page, <https://openai.com/api/pricing/>

³⁰ OpenAI. (n.d.). OpenAI Platform Deprecations. Retrieved May 21, 2025, from <https://platform.openai.com/docs/deprecations>

³¹ OpenAI. (2025, April 15). *OpenAI o3 and o4-mini System Card*. <https://openai.com/index/o3-o4-mini-system-card/>

Capabilities

- **Strong capabilities in coding and mathematics:** These models excel in mathematics and coding tasks and are able to correct their thinking before giving a response to the user.
- **Advanced reasoning capabilities:** These models take time to answer the given query, resulting in an advanced reasoning process where artificial intelligence will reflect on itself to produce a more advanced response.
- **Multimodal Input:** This model allows multimodal input such as plain text, PDFs, and various format documents, images, and more.

Limitations

- **Slowness:** Due to their advanced reasoning process, these models are significantly slower in their response time. As our project needs a real-time response, this will be a big concern.
- **Knowledge Cutoff:** While not a critical issue for our current application, it is important to note that this model cannot handle responding with actual up-to-date information, which may limit its use in dynamic or current-event-related queries.
- **Platform dependent:** This model is closed source and only accessible with the OpenAI API.

2.5.3. Anthropic

Anthropic is an AI safety and research company dedicated to the development of AI systems and models. Structured as a Public Beneficial Corporation, Anthropic's mission is the development and maintenance of advanced AI for the long-term benefit of humanity.³²

2.5.3.1. Claude 4

Description

Claude 4 models are the newest flagship of Anthropic introduced on the 22 May 2025. It offers better performance at coding tasks than Claude 3.7. These models also received major improvements in running long tasks and agent workflows.³³

This new model has declined into 2 versions. Claude 3.7 Opus is the most powerful model that is built to respond to the most complex tasks. This model has the best performance of coding with a 72.5% score on SWE-bench. On the other hand, Claude 4 Sonnet is a smaller more efficient model that achieves practically the same scoring as the Opus versions³³.

Pricing

The pricing of the Claude 4 model is detailed in the screen below. Input tokens cost significantly less than output tokens for these models. (These prices are exactly the same as Claude 3.7.)

Claude Opus 4 • Most intelligent model for complex tasks • 200K context window • 50% discount with batch processing*			
Input	\$15 / MTok	Prompt caching write	\$18.75 / MTok
		Prompt caching read	\$1.50 / MTok
Output	\$75 / MTok		
Claude Sonnet 4 • Optimal balance of intelligence, cost, and speed • 200K context window • 50% discount with batch processing*			
Input	\$3 / MTok	Prompt caching write	\$3.75 / MTok
		Prompt caching read	\$0.30 / MTok
Output	\$15 / MTok		

Figure 15: Pricing of Claude Opus 4 and Claude Sonnet 4

Captured on May 2025 from Anthropic pricing page, <https://www.anthropic.com/pricing#api>

³² Anthropic. (n.d.). *Company*. Retrieved May 21, 2025, from <https://www.anthropic.com/company>

³³ Anthropic. (2025, May 22). *Introducing Claude 4*. <https://www.anthropic.com/news/claude-4>

Capabilities

- **Extended thinking with tools:** The thinking mode of Claude 4 can now use tools to generate more accurate responses.
- **Parallel usage of tools:** Tools on both Claude 4 models can be run in parallel, which increases the speed of the answers.
- **Coding performance:** Claude 4 models have amazing coding performances.

Limitations

- **Cost:** The cost of the model is a bit high, especially for the output tokens. This can become an issue if we need to generate longer answers. But with the new improvement of capabilities this pricing becomes not so high.
- **Knowledge Cutoff:** While not a critical issue for our current application, it is important to note that this model cannot handle responding with actual up-to-date information, which may limit its use in dynamic or current-event-related queries.
- **Platform dependent:** This model is closed source and only accessible with the Anthropic API

2.5.3.2. Claude 3.7 Sonnet

Description

Claude 3.7 Sonnet LLM from Anthropic introduced on February 24, 2025. It offers a significant improvement over the previous Claude 3.5 model. This new model enhances capabilities in coding, following instructions, and long-context comprehension. With a context of 200,000 tokens (about 500 pages) and a knowledge cutoff of June 2024, Claude 3.7 is a strong-performing model for complex tasks. ^{34,35}

This model is available in 2 different versions that serve different purposes. The full version, Claude 3.7 Sonnet, is designed for the most complex tasks that require a strong and powerful model. Additionally, Claude 3.7 offers a “thinking” mode, that allows more elaborate, step by step reasoning when this is needed. ^{34,35}

Pricing

The screen below details the pricing of Claude 3.7 Sonnet. Input tokens cost significantly less than output tokens for this model.

Claude 3.7 Sonnet			
- Most intelligent model, with visible step-by-step reasoning 200K context window 50% discount with batch processing*	Input	Prompt caching write	Output
	\$3 / MTok	\$3.75 / MTok	\$15 / MTok
		Prompt caching read	
		\$0.30 / MTok	

Figure 16: Pricing of Claude 3.7 Sonnet

Captured on May 2025 from Anthropic pricing page,
<https://web.archive.org/web/20250518100530/https://www.anthropic.com/pricing>

Capabilities

- **Complex reasoning:** Claude excels in complex reasoning tasks that require multi-step problems and follow instruction reliability.
- **Multimodal Understanding:** Claude can interpret and extract information from various data types, including CSV documents, images, graphs, and complex diagrams, making it a powerful tool to interpret specific data.
- **Thinking mode:** Claude provides a thinking mode that will take longer to create a response that will be a more complete response.

³⁴ Anthropic. (n.d.). *Claude 3.7 Sonnet*. Retrieved May 20, 2025, from <https://web.archive.org/web/20250518164908/https://www.anthropic.com/claude/sonnet>

³⁵ Anthropic. (2025, February 24). *Claude 3.7 Sonnet and Claude Code*. <https://www.anthropic.com/news/claude-3-7-sonnet>

Limitations

- **Cost:** The cost of the model is a bit high, especially for the output tokens. This can become an issue if we need to generate longer answers.
- **Knowledge Cutoff:** While not a critical issue for our current application, it is important to note that this model cannot handle responding with actual up-to-date information, which may limit its use in dynamic or current-event-related queries.
- **Platform dependent:** This model is closed source and only accessible with the Anthropic API

2.5.4. xAI

xAI is an artificial intelligence company founded by Elon Musk in 2023. It has a goal to accelerate human scientific discoveries discovery and our understanding of the universe. It operates as a public benefit corporation, committed to prioritizing social impact rather than profit.³⁶

2.5.4.1. grok-3

Description

Grok-3 is the latest model developed by xAI released with a beta version on the 19 February 2025. Built from scratch by xAI, Grok-3 represents a significant step in reasoning capabilities. It has been trained on the Colossus supercluster that is 10 times more powerful than the previous model. Grok 3 offers significant improvements in reasoning, mathematics, coding, world knowledge, and instruction-following tasks.³⁷

This new version of Grok has declined in 2 versions. The normal version, Grok-3 has strong capabilities in complex tasks, and the lighter Grok 3 Mini has faster and more cost-effective responses with slightly reduced accuracy.³⁷ xAI also provides a fast mode of these 2 models that uses a faster infrastructure to provide significantly faster response time.³⁸

Pricing

Grok 3 pricing matches the pricing of other LLMs on the market. The normal version of Grok-3 costs 10 times more than the mini version. xAI also offers a fast mode that costs about 2 times more than the normal mode to be prioritized on the API. For a detailed vision, please see the screen below.

GROK 3		GROK 3 MINI	
Grok 3 Excels at enterprise use cases like data extraction, coding, and text summarization. Possesses deep domain knowledge in finance, healthcare, law, and science.		Grok 3 Mini (with Thinking) A lightweight model that thinks before responding. Fast, smart, and great for logic-based tasks that do not require deep domain knowledge. The raw thinking traces are accessible.	
Example Usage		Example Usage	
grok-3		grok-3-mini	
Fast mode ☐		Fast mode ☐	
↑ Input	\$3.00/M	↑ Input	\$0.30/M
↓ Output	\$15.00/M	↓ Output	\$0.50/M

Figure 17: Pricing of Grok 3 and Grok 3 mini

Captured on May 2025 from xAI model documentation, <https://docs.x.ai/docs/models>

³⁶ xAI. (n.d.). *Company*. Retrieved May 21, 2025, from <https://x.ai/company>

³⁷ xAI. (n.d.). *Grok 3*. Retrieved May 21, 2025, from <https://x.ai/news/grok-3>

³⁸ xAI. (n.d.). *Models*. Retrieved May 21, 2025, from <https://docs.x.ai/docs/models>

Capabilities

- **Context Windows:** With a context window of 1 million tokens (about 2500 pages), Grok has the possibility to process long-form content effectively.
- **Big brain mode:** Grok offers a “Big brain mode” that allows the model to reflect much longer on a query to produce better results.

Limitations

- **Accuracy:** The mini version of Grok 3 Mini has lower accuracy, especially on tasks that require extensive knowledge.
- **Context Windows:** Grok 3 Mini struggles to use the full context in particular on complex tasks which leads to inconsistencies.
- **Knowledge Cutoff:** This model has been trained on data up to October 2024, which led to inaccuracy in recent events and actuality. For our project, this point will not be critical.
- **Platform dependent:** This model is closed source and only accessible with the xAI API.

2.5.5. Google DeepMind

Google DeepMind is an artificial intelligence research lab focused on building AI responsible for the benefit of humanity. It is a subsidiary of Alphabet Inc. focused on developing a new AI system. The team is composed of scientists, engineers, and ethicists working together to discover new advances in the AI field.³⁹

2.5.5.1. Gemini 2.5

Description

Gemini 2.5 Pro is Google DeepMind’s most advanced AI model, released on April 9, 2025. This model is designed to handle complex tasks with enhanced reasoning, coding understanding, and multimodal management.⁴⁰

Gemini 2.5 Pro excels in processing complex prompts and coding challenges. It is supported by default multimodality, which enables us to process information from text, audio, images, videos, and files. This model also has a substantial window context of 1 million tokens (about 2500 pages), with some plans to upgrade it to 2 million tokens.⁴⁰

Gemini 2.5 has been released in 2 major versions. The full version of Gemini 2.5 Pro is designed for complex prompting and excels at coding tasks. Then Gemini 2.5 Flash is a lightweight version that delivers nearly the same performance, achieving practically the same score as Gemini 2.5 Pro on LMArena for a fraction of the price of the Pro model ⁴¹. As of May 22, 2025, Gemini 2.5 Pro and Gemini 2.5 Flash hold the 2 top spots on LMArena with overall score of 1446 and 1424.⁴²

³⁹ Google DeepMind. (n.d.). *About Google DeepMind*. Google DeepMind. Retrieved May 21, 2025, from <https://deepmind.google/about/>

⁴⁰ Kavukcuoglu, K. (2025, March 28). *Gemini 2.5: Our most intelligent AI model*. Google. <https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/>

⁴¹ Google DeepMind. (n.d.). *Gemini*. Google DeepMind. Retrieved May 22, 2025, from <https://deepmind.google/models/gemini/>

⁴² LMArena. (n.d.). *Leaderboard*. Retrieved June 06, 2025, from <https://lmarena.ai/leaderboard>

Pricing

Gemini 2.5 Pro and Flash are relatively cheap and cost-effective models for their relatives' capabilities. To have a detailed vision of the pricing, see the screenshot below.

Gemini 2.5

Model	Type	Price (/1M tokens) <= 200K input tokens	Price (/1M tokens) > 200K input tokens
Gemini 2.5 Pro	Input (text, image, video, audio)	\$1.25	\$2.5
	Text output (response and reasoning)	\$10	\$15
Gemini 2.5 Flash	Input (text, image, video)	\$0.15	\$0.15
	Audio Input	\$1	\$1
	Text output (no thinking)	\$0.60	\$0.60
	Text output (thinking- response and reasoning)	\$3.50	\$3.50

* If a query input context is longer than 200K tokens, all tokens (input and output) are charged at long context rates.

Figure 18: Pricing of Gemini 2.5 Pro and Gemini 2.5 Flash

Captured on May 2025, Vertex AI generative AI Pricing, <https://cloud.google.com/vertex-ai/generative-ai/pricing>

Capabilities

- **Context Windows:** With a context window of 1 million tokens (about 2500 pages) and up to 2 million tokens in the coming months, Gemini 2.5 can handle an entire codebase to analyze it more easily.
- **Multimodal Understanding:** These models can handle various types of information such as images, documents, videos, audio and plain text. This makes an interesting point, as it allows us more flexibility and opens new gates to build a more powerful chatbot.
- **Reasoning capabilities:** Gemini 2.5 Pro has the ability to think step by step in order to process more complex prompts.
- **Cost-effectiveness:** Compared to the other model for their capabilities, these models are relatively cheap to operate

43

Limitations

- **Knowledge Cutoff:** This model has been trained on data up to January 2025, which led to inaccuracy in recent events and actuality. For our project, this point will not be critical.
- **Platform dependent:** This model is closed source and only accessible with Google Cloud API and sometimes Google Cloud Interface can make it hard to find information easily. But that is clearly a small weak point.

2.5.6. Meta AI

Meta AI is a research division of Meta Platforms that develops AI products and augmented technologies. It is engaged in several open-source activities, initiatives, and contributes to academic research, but it remains a for-profit division. Its activities are aimed at enhancing the Meta's product and services.⁴⁴

2.5.6.1. Llama 4 and Llama 3

Description

Llama 4 is the latest model released by Meta AI on April 5, 2025. This new generation of models has received new improvements with native multimodal capabilities and a significant upgrade to their context windows. Llama 4 models are available in 3 versions. Llama 4 Scout is the light way version with 17 billion parameters and 10 million tokens (25000 pages). The context allows the model to easily process visual

⁴³ Google DeepMind. (n.d.). *Gemini 2.5 Pro | Generative AI on Vertex AI*. Google Cloud. Retrieved May 21, 2025, from <https://cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-5-pro>

⁴⁴ Meta AI. (n.d.). About. Retrieved May 22, 2025, from <https://ai.meta.com/about/>

information or even an entire codebase. Then Llama 4 Maverick is a smaller model that generate a faster response while keeping a score close to Llama 4 Scout.⁴⁵

The last model that Meta will propose soon will be Llama 4 Behemoth, which is the teacher model that was used to distill⁴⁶ Llama 4 Scout and Llama 4 Maverick. This model will have 2 trillion parameters, which will be the highest model of Meta AI, slightly better than GPT-4.5 or Gemini 2.0 Pro that were already last generation model.⁴⁷

Llama 3 is the last generation models released by Meta AI on April 18, 2024, and declined later into multiple models with different purposes have still performance that are equivalent to all free commonly available models such as Gpt-4o which make them great for project that does not require heavy models. These models range from a 2 and 3 billion parameter model that has been built to run on a mobile device to a 405 billion parameter model that allows heavier tasks.⁴⁸

Pricing

These models are built to be self-hosted by users or organizations so the pricing of the model will highly depend on whether it is hosted on a local machine, a local server or by a company that offers cloud services to host IA models. If you want more information about Infomaniak, which has an IA tool service, please see the following section, [Infomaniak IA Tools](#).

Capabilities

- **Multimodal understanding:** Llama 4 models are by default capable of understand various data types such as file images, text, and more.
- **Context Windows:** Llama 4 models have a large context window of up to 10 million that can easily fit multiple code bases or multiple books at once.
- **Open Source:** These models are open source, which means that we can use the model as we want for a project and can be install locally.
- **Cost:** As the model can be installed locally on a PC, the price can be as low as the electricity price to run the model. (This will highly depend on the chosen model, the more powerful and heavier the model will be the host machine will need to be more powerful which can increase costs)

Limitations

- **High computational cost for the higher model:** If we want to use Llama 4 Behemoth, it will require a powerful server to host it, which can cause some issues depending on the use case.
- **Slowness of answers:** A local model can be slower, as it depends on the hardware of the machine to work. This slowness can lead to problems when a fast-responding AI is required.
- **Knowledge Cutoff:** While not a critical issue for our current application, it is important to note that this model cannot handle responding with actual up-to-date information, which may limit its use in dynamic or current-event-related queries.

⁴⁵ Meta AI. (n.d.). *Llama 4*. Llama. Retrieved May 22, 2025, from <https://www.llama.com/models/llama-4/>

⁴⁶ Knowledge distillation refers, in the context of AI model a technique where knowledge from a large, complex model (teacher model) is transferred to a smaller, more efficient model (student model). The main goal of distillation is to create a more compact model that mimics the larger one while keeping as much knowledge as possible. This results in a faster, slightly less powerful student model. Bergmann, D. (2024, April 16). *What is Knowledge distillation?* | IBM. <https://www.ibm.com/think/topics/knowledge-distillation>

⁴⁷ Meta AI. (2025, April 5). *The Llama 4 herd*. Llama. <https://ai.meta.com/blog/llama-4-multimodal-intelligence/>

⁴⁸ Meta AI. (n.d.). *Llama 3*. Llama. Retrieved May 22, 2025, from <https://www.llama.com/models/llama-3/>

2.5.7. DeepSeek

Hangzhou DeepSeek Artificial Intelligence Basic Technology Research Co., Ltd., called DeepSeek, is a Chinese AI company that creates LLMs. It was founded on July 2023 and has created multiple LLMs with eponymous names.⁴⁹

2.5.7.1. DeepSeek-V3 and DeepSeek-R1

Description

DeepSeek-V3 is an advanced open-source Mixture-of-Experts (MoE)⁵⁰ language model. Released in 2025, this model comprises 671 billion parameters with 37 billion activated per token enabling efficient processing of complex reasoning tasks, natural language understanding, and code generation. DeepSeek-V3 also demonstrates strong competitive performance compared to other open-source and closed-sources models.⁵¹

DeepSeek-R1 on the other hand, is the first reasoning model in the DeepSeek series. Also released in 2025, this model demonstrates strong reasoning capabilities. It was trained with large-scale reinforcement learning⁵² (RL) without supervised fine-tuning methods.⁵¹

Pricing

As these models are open source, they can be hosted locally at no cost if the machine provided the machine has sufficient resources to run the model, e.g. some of these models can take a lot of RAM and storage. This, we think, is an important point to show, as it allows anyone to use advanced AI models without needing to be connected to the internet or pay something.⁵³

DeepSeek also provides an API endpoint to access its model. Here below is detailed pricing information:

⁴⁹ 全国互联网安全管理平台. (n.d.). Retrieved May 22, 2025, from

<https://beian.mps.gov.cn/#/query/webSearch?code=33010502011812>

⁵⁰ Mixture-of-Experts is a machine learning approach that breaks down large language models into smaller, specialized sub-networks. It allows you to have billions of parameters for the model while greatly reducing the computation cost to train the model. Bergmann, D. (2024, April 5). *What is mixture of experts?* | IBM. <https://www.ibm.com/think/topics/mixture-of-experts>

Bhavishya, P. (2024, July 28). *What Is Mixture of Experts (MoE)?* <https://www.datacamp.com/blog/mixture-of-experts-moe>

⁵¹ *deepseek-ai/DeepSeek-V3*. (2025). DeepSeek. <https://github.com/deepseek-ai/DeepSeek-V3> (Original work published 2024)

⁵² Large-scale RL without supervised fine-tuning trains a base model by rewarding correct and well formatted reasoning, letting the model learn complex problem solving on its own. This enables strong reasoning skills without using labeled data first. (DeepSeek-AI et al., 2025)

⁵³ Ollama. (n.d.). *deepseek-r1*. Ollama. Retrieved May 22, 2025, from <https://ollama.com/library/deepseek-r1>

MODEL ⁽¹⁾		deepseek-chat	deepseek-reasoner
CONTEXT LENGTH		64K	64K
MAX COT TOKENS ⁽²⁾		-	32K
MAX OUTPUT TOKENS ⁽³⁾		8K	8K
STANDARD PRICE (UTC 00:30-16:30)	1M TOKENS INPUT (CACHE HIT) ⁽⁴⁾	\$0.07	\$0.14
	1M TOKENS INPUT (CACHE MISS)	\$0.27	\$0.55
	1M TOKENS OUTPUT ⁽⁵⁾	\$1.10	\$2.19
DISCOUNT PRICE ⁽⁶⁾ (UTC 16:30-00:30)	1M TOKENS INPUT (CACHE HIT)	\$0.035 (50% OFF)	\$0.035 (75% OFF)
	1M TOKENS INPUT (CACHE MISS)	\$0.135 (50% OFF)	\$0.135 (75% OFF)
	1M TOKENS OUTPUT	\$0.550 (50% OFF)	\$0.550 (75% OFF)

- (1) The `deepseek-chat` model points to **DeepSeek-V3**. The `deepseek-reasoner` model points to **DeepSeek-R1**.
- (2) **CoT (Chain of Thought)** is the reasoning content `deepseek-reasoner` gives before output the final answer. For details, please refer to [Reasoning Model](#).
- (3) If `max_tokens` is not specified, the default maximum output length is 4K. Please adjust `max_tokens` to support longer outputs.
- (4) Please check [DeepSeek Context Caching](#) for the details of Context Caching.
- (5) The output token count of `deepseek-reasoner` includes all tokens from CoT and the final answer, and they are priced equally.
- (6) DeepSeek API provides **off-peak pricing discounts** during 16:30-00:30 UTC each day. The completion timestamp of each request determines its pricing tier.

Figure 19: Detailed pricing of DeepSeek-V3 and DeepSeek-R1

Captured in May 2025, DeepSeek documentation pricing, https://api-docs.deepseek.com/quick_start/pricing/

It is also possible to host it on other platforms. For more information about this, see the section titled, 2.9.3 Infomaniak IA Tools.

Capabilities

- **High number of parameters:** DeepSeek R1 has a high number of parameters, specifically 671 billion parameters, is able to understand natural languages well and has more power to process requests.
- **Reasoning capabilities:** DeepSeek R1 has the possibility to reflect before answering a query, which makes him more accurate in most cases.
- **Open Source:** These models are open source, which means that we can use the model as we want for a project and can be installed it locally.
- **Cost:** As the model can be installed locally on a PC, the price can be as low as the electricity price to run the model. (This will highly depend on the chosen model, the more powerful and heavier the model will be the host machine will need to be more powerful which can increase costs)

Limitations

- **High computational cost for the higher model:** Higher models such as DeepSeek R1 require a lot of RAM and power to be run locally. That can reduce the facility's use of this model depending on the project.
- **Slowness of answers:** A local model can be slower as it depends on the hardware of the machine to work. This can lead to some problems if we need a fast-responding AI.
- **Knowledge Cutoff:** While not a critical issue for our current application, it is important to note that this model cannot handle responding with actual up-to-date information, which may limit its use in dynamic or current-event-related queries.

2.5.8. Mistral AI

Mistral AI is a French AI founded in April 2023 by researchers Mensch, Guillaume Lample, and Timothée Lacroix. Mistral AI as a goal to make AI accessible for everyone and they focus on developing open-source, efficient and innovative models. The company model of Mistral AI is a profitable company. ⁵⁴

⁵⁴ Mistral. (2025, May 23). About. <https://mistral.ai/about>

2.5.8.1. Devstral

Description

Devstral is the latest model from Mistral AI built for agentic applications. This model was released on May 21, 2025. This model is a light model with only 26 billion parameters that has been specially created for agentic apps.

This model is open source under the Apache 2 license. The important breakthrough with this model is performance compared to much larger models like DeepSeek-V3 or Qwen3. It makes this model a good choice of a lightweight but performant model.

Pricing

Devstral can be either self-hosted on a machine that has an RTX 4090 from Nvidia and 32 GB of RAM. This makes the self-hosting of this model not too difficult.

Mistral AI also offers a way to interact with this open-source model via their API. Here below are the detailed pricing.⁵⁵

	Devstral	The best open-source model for coding agents.	Input (/M tokens) \$0.1	Output (/M tokens) \$0.3
---	----------	---	----------------------------	-----------------------------

Figure 20: Pricing of Devstral

Captured on May 2025, Mistral AI pricing, <https://mistral.ai/pricing#api-pricing>

Capabilities

- **Performance relative to the size:** Devstral achieves great performance relative to the size of the model and is as performant as OpenAI GPT-4.1 mini model.
- **Build for agentic app:** Devstral has been trained to follow order and prompt strictly with is a huge plus for building agentic app with multi-step agent and big system prompts.
- **Open Source:** These models are open source, which means that we can use the model as we want for a project and can be installed locally.
- **Cost:** As the model can be installed locally on a PC, the price can be as low as the electricity price to run the model. And this model is lightweight so it will not consume much power compared to

Limitations

- **Not so powerful:** This model is enough for simple tasks or relatively complex prompts but will struggle with really complex prompt and multistep thinking
- **Knowledge Cutoff:** While not a critical issue for our current application, it is important to note that this model cannot handle responding with actual up-to-date information, which may limit its use in dynamic or current-event-related queries.

⁵⁵ <https://huggingface.co/mistralai/Devstral-Small-2505> · Hugging Face. (2025, May 21).

2.5.9. Model Comparisons

In order to have a global vision of all the models explained earlier available we have relied on an Artificial Analysis company this is a well-known analysis company to compare IA models.⁵⁶ The following section will present to you their comparative results on some key points to allow you to have a more global vision of all the models available. In short, this section is like a resume of the detailed model section but more on a technical and measurable manner.

2.5.9.1. Artificial Analysis Intelligence Index

The Artificial Analysis Intelligence Index is a metric developed by Artificial Analysis to rank the intelligence of large language model across various dimensions, like reasoning, knowledge, math, and programming Artificial Analysis.⁵⁷

Artificial Analysis Intelligence Index

Artificial Analysis Intelligence Index incorporates 7 evaluations: MMLU-Pro, GPQA Diamond, Humanity's Last Exam, LiveCodeBench, SciCode, AIME, MATH-500

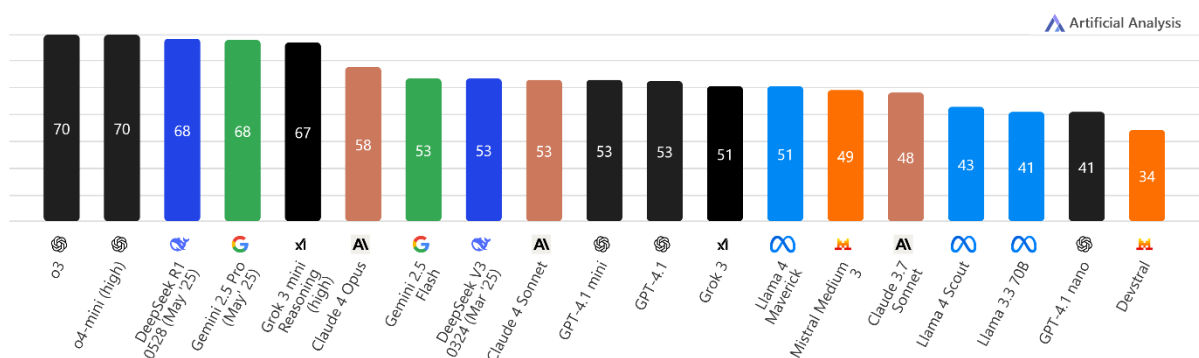


Figure 21: Artificial Analysis - Artificial Analysis Intelligence Index

Retrieved from Artificial Analysis, on the 10th of July 2025, <https://artificialanalysis.ai/>

The results presented above show an important point all reasoning models like o3 or Grok 3 reasoning models show a gap in their intelligence score. After that, non-reasoning models are performing quite similarly followed at the end by smaller models. This graph will be interesting to compare to our future benchmarking to see if the result is similar or totally different.

⁵⁶ AI Model & API Providers Analysis | Artificial Analysis. (n.d.). Retrieved June 18, 2025, from <https://artificialanalysis.ai>

⁵⁷ Intelligence Benchmarking | Artificial Analysis. (n.d.). Retrieved July 10, 2025, from <https://artificialanalysis.ai/methodology/intelligence-benchmarking>

2.5.9.2. Pricing

One of the most important metrics that must be analyzed is the pricing of each different model. The Artificial Analysis company has also created graphs to show the different models. Here below is the comparison of the input and output tokens for each model.

Pricing: Input and Output Prices

Price: USD per 1M Tokens

■ Input price ■ Output price

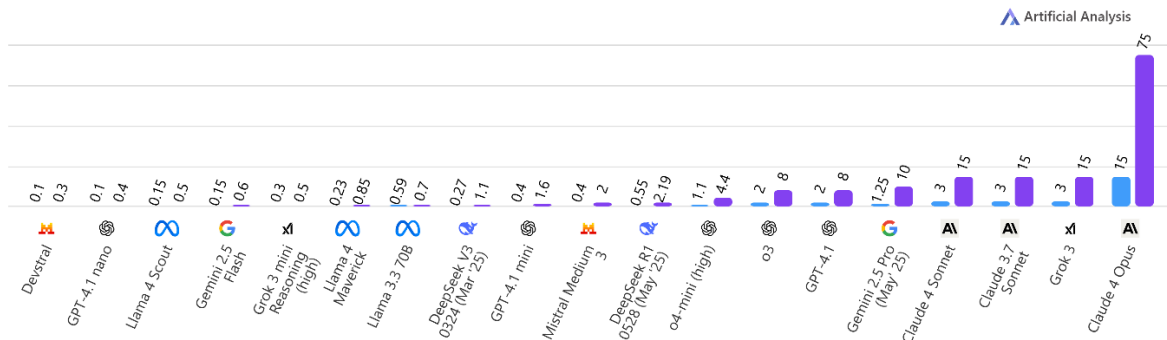


Figure 22: Artificial Analysis - Pricing: Input and Output Prices

Retrieved from Artificial Analysis, on the 10th of July 2025, <https://artificialanalysis.ai/>

Here we can see that there are 3 principal groups of pricing for each model:

- **Small models:**
- This group include smaller model form company like “GPT-4.1 nano” as well as most open-source models. These models are lightweight and typically optimized for efficiency over their accuracy and complexity.
 - The pricing of this category ranges from 0.1 USD/millions of tokens to around 1.1 USD/ millions of tokens.
- **Standard model and Reasoning model:**
- This category includes most of the general-purpose models available as well as some of the reasoning. These models are more capable and can resolve more complex queries
 - The pricing of this category ranges from 0.1 USD/millions of tokens to around 1.1 USD/ millions of tokens.
- **High-End Reasoning and Anthropic Models:**
- These are usually the most advanced modes with strong reasoning capabilities. Anthropic models are practically all located inside this category which means their model costs clearly more than comparable models from other companies.
 - The pricing of this category ranges from 8 USD/millions of tokens to around 75 USD/ millions of tokens.

To better visualize these categories, it is possible to show a graph that compares the pricing of the different models relative to their intelligence index. Here we can clearly see 2 clusters on the left with Claude 4 opus on the right that is completely alone

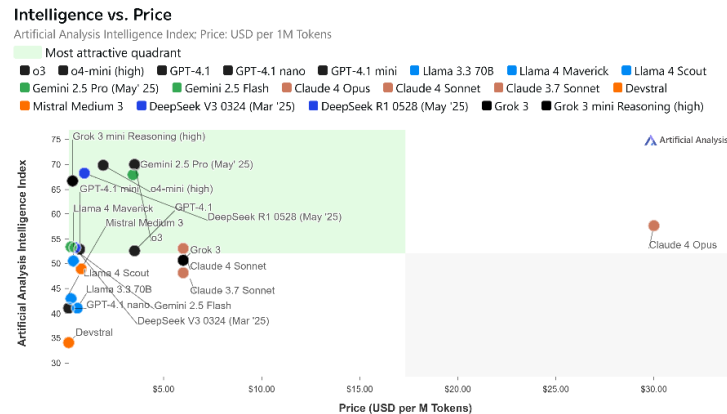


Figure 23: Artificial Analysis - Intelligence vs. Price

Retrieved from Artificial Analysis, on the 10th of July 2025, <https://artificialanalysis.ai/>

2.5.9.3. Output speed

One of the last interesting metrics to analyze is the speed of the generation of tokens for each model. This will have a correlation with the real speed of the model but not represent its exact speed because some models may generate more tokens to answer a given question and thus will take more time.

Output Speed

Output Tokens per Second; Higher is better

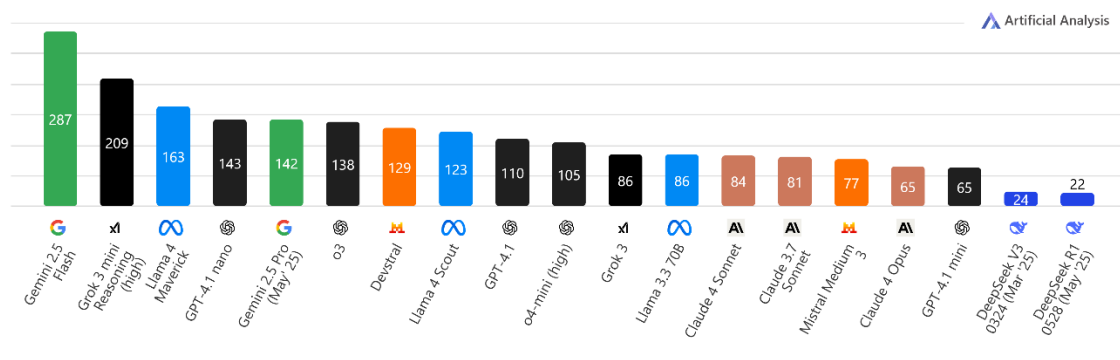


Figure 24: Artificial Analysis - Output pricing

Retrieved from Artificial Analysis, on the 10th of July 2025, <https://artificialanalysis.ai/>

On this graph we can clearly see that the smaller models are generally faster to generate more tokens in a given time.

For example, Gemini 2.5 Flash, which is a lightweight model from Google leads the competition with nearly 300 tokens generated per second. This model is followed by the mini version of the Grok 3 model with just over 200 tokens per second. The third place is held by Llama 4 Maverick, also a lightweight model.

As we move to the right on the graph, we start to see more capable models and reasoning models with o3 from OpenAI or Devstral from Mistral AI. At the bottom of the chart is located the DeepSeek model, with a significant lack of speed compared to all the other models.

This tradeoff between the speed of the model and its general intelligence will be an important point for our chatbot to consider as it will be crucial to create a chatbot where users will not have to wait too much to get their information.

2.6. List of Embedding models available

Embeddings are a way to transform natural language into processable data. To learn more about embeddings see the following section: 2.4.2.5 Transformers, Tokens and embeddings.

The following section will present you with some of the embedding models available actually. This will be grouped into 2 main sections: the local embeddings model and the proprietary embeddings model.

2.6.1. Generalities

The current landscape of embedding models available locally is really diverse and there are a lot of models optimized for different languages, tasks, or computational constraints. To explain a bit more in detail some models that fit the best needs for this project we rely on the Massive Text Embedding Benchmark (MTEB) which is a standardized benchmark that evaluates models across a broad range of tasks, including classification, clustering, semantic search and more. (Muennighoff et al., 2023)

To identify the most suitable models for a local deployment we have used MTEB with the following criteria:

- **Filter by the language:** Models have been filtered by European languages as our chatbot needs to be performant both in French and German.
- **Select the top performers:** We have chosen one of the best-performing models available with the global MTEB score.
- **Ensure Local Suitability:** Only smaller models that can run practically everywhere have been selected to avoid consuming like 32GB of RAM just for the embedding model. For that we have decided to choose a model under 1 billion parameters.

2.6.2. Proprietary models

2.6.2.1. OpenAI text-embedding and text-ada-002

OpenAI offers several performance-embedding models designed for tasks such as semantic search, clustering, classification, and recommendation systems. Of these, text-embedding-ada-002 and the more recent text-embedding-small/large versions are widely used in practice.

text-embedding-ada-002

text-embedding-ada-002 model was introduced by OpenAI in December 2022 and rapidly became the default embedding model in their API replaced later on by text-embedding small. It generates 1,536-dimensional embeddings and is based on a transformer architecture explain in section 2.4.2.5 capable of capturing both semantic and syntactic relationships between words and sentences⁵⁸. This model has demonstrated robust performance in various use cases, including question answering, document similarity, and information retrieval.

According to evaluations on the Massive Text Embedding Benchmark (MTEB), text-embedding-ada-002 achieves a relatively correct score on multiple tasks. His score is now low compared to the newer models, but it is still used as a default embedding at several places. (Muennighoff et al., 2023)

⁵⁸ New and improved embedding model. (2022, January 25). <https://openai.com/index/new-and-improved-embedding-model/>

text-embedding-3-small and text-embedding-3-large

The text-embedding-3, released by OpenAI on 25, January 2024, introduces two new models: text-embedding-3-small and text-embedding-3-large. These 2 models are designed to improve earlier models like text-embedding-ada-002 by providing better semantic representations, and a less costly model that achieves better performance.⁵⁹

The 3-small model has been fine-tuned for speed and efficiency, supporting 1536-dimensional embeddings. It is ideal for real-time and small applications, including mobile or embedded systems. On the other hand 3-large offers high-precision, 3,072-dimensional embeddings optimized for more complex tasks such as semantic search, multilingual understanding, and RAG (retrieval-augmented generation) pipelines. Both models support the same input length, that is 8,192 tokens.⁵⁹

2.6.3. Open-source models

2.6.3.1. all-MiniLM-L6-v2

all-MiniLM-L6-v2 is a small and fast model for generating embeddings that was primarily trained on English data. It's widely used for tasks like semantic similarity, clustering, and information retrieval (RAG). Despite its compact size, it offer strong performance and in real-time or resource-limited environments.⁶⁰ Also despite that it was not designed for multilingual tasks but it show shows acceptable performance on non-English texts, making it a versatile choice for lightweight multilingual applications (Muennighoff et al., 2023)

2.6.3.2. multilingual-e5-large-instruct

multilingual-e5-large-instruct is a powerful multilingual embedding model created by Microsoft. It has one billion parameters. It has been trained on datasets of over 100 different languages making it suited for complex tasks like multilingual semantic search retrieval. Its strong performance on multilingual benchmarks makes it a good choice for applications that operate across diverse languages and on a user base. It also has a remarkable performance on the MTEB benchmark. (Wang et al., 2024)

⁵⁹ New embedding models and API updates. (2024, January 25). <https://openai.com/index/new-embedding-models-and-api-updates/>

⁶⁰ sentence-transformers/all-MiniLM-L6-v2 · Hugging Face. (2024, January 5). <https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>

2.7. Available dataset and information's

This project relies on two main categories of data sources: publicly accessible resources and a confidential internal dataset used for the geographic position of the collection points.

2.7.1. Publicly available online resources

The first category includes multiple official websites that provide some information on waste management practices:

- **Canton of Valais Website** (<https://www.vs.ch/web/sen/dechets-introduction>) : Inside the "Waste" section, this portal hosts multiple pages and PDFs that detail local sorting guidelines and municipal regulations and more like podcast about waste management.
- **Swiss Confederation Website** (<https://www.bafu.admin.ch/bafu/en/home/topics/waste/guide-to-waste-a-z.html>): This federal platform centralizes information about waste classification and disposal procedures across the whole country, serving as a base of work for the canton and an information website for the citizen.
- **Swiss Recycling** (<https://swissrecycling.ch>): As discussed in section 2.2, this website offers detailed explanations for each type of recyclable material, including best practices, how waste is processed, and more

These publicly accessible sources will create the base knowledge for the large language model, enabling accurate answers to citizen questions.

It is also important to note that many of these documents and web pages are available in multiple languages that include French and German, depending on the source. This multilingual aspect introduces additional complexity for the information retrieval aspect and coherence of answers in multiple languages.

2.7.2. Confidential Dataset of Recycling Points

The second category is a confidential dataset that contains a list of recycling locations in the Canton of Valais. Each entry includes metadata such as:

- The exact coordinates of the point.
- A complete list of all accepted types of waste
- Basic information about the place such as its name which communal state it belongs to or its complete address

This dataset is not publicly available and must be used with strict confidentiality and data protection measures. This plays an important role in the dynamic Geographic Information System (GIS) component of the project as LLMs must not have access to the entire dataset, enabling spatial visualization of relevant recycling points.

Additionally, the GeoJSON file used to represent these locations is fully labelled in German, particularly for category names and feature descriptions. This will require careful preprocessing to ensure consistency in multilingual responses.

Together, these datasets support both the retrieval-augmented generation (RAG) system and the dynamic mapping functions integrated into the agent architecture.

2.8. Available frameworks

LLM development has converged in the recent year on several libraries that are used to create magnetic applications and RAG workflows. Nowadays big AI companies try to shift this by creating their own ecosystems and libraries like OpenAI with OpenAI Agent SDK or recently Google with Agent Development Kit. For this section we will not talk about all the frameworks available, but we have decided to choose the 2 most important ones, LlamaIndex and LangChain, and also explain one proprietary library, OpenAI SDK agent.

2.8.1. LlamaIndex

First released under the name GPT-Index in November 2022⁶¹, Llama index is the framework designed for building context-augmented generative AI applications with LLMs including agents and workflows. LlamaIndex continues to be heavily updated and acquire new functionalities.⁶²

2.8.1.1. Key Capabilities

- **Data Loaders and Connectors:** LlamaIndex has the ability to load data from a wide range of sources, including:
 - o Various file types including PDF, DOCX, Markdown, CSV, JSON
 - o APIs and database connectors: SQL, Notion, Google Drive, etc.
 - o Web resources: HTML pages, websites, sitemaps⁶³
- **Indexing and Chunking:** Intelligent indexes and chunking capabilities with various strategies (list, vector stores, Tree, metadata). Enables optimal storage and retrieval.⁶⁴
- **Strong debug capabilities:** LlamaIndex offers extensive ways to debug our workflow in order to understand which step has been called with which tool.
- **Support of MCP tools:** LlamaIndex supports the creation of MCP tools

2.8.1.2. Advantages

- **Modular and Flexible:** Each component of LlamaIndex like indexing, retrieval, agent, and tool, is independent and can be exchanged which allows you to build your workflow for your specific need. integration into both simple and complex systems. You can also quickly create custom elements like a new indexing system or new models.
- **RAG Capabilities:** It is strong for applications when an LLM needs to retrieve and find specific answers in long or structured documents (e.g., PDFs, reports, public datasets).
- **Strong Agent Integration:** You can easily integrate agents with external tools such as search engines and APIs, which are ideal for building multi-modal or tool-augmented assistants. It also fully supports MCP tools.
- **Good Ecosystem and large Compatibility:** Native integrations with LangChain, OpenAI, Chroma, All popular model available, and other popular.⁶⁵

2.8.1.3. Downside

- **Learning Curve:** LlamaIndex is easy to start with but can rapidly become hard to build complex systems directly due to its high flexibility and enormous documentation.
- **Relatively new:** LlamaIndex is a new framework that evolves quickly which can introduce

⁶¹ *GitHub - run-llama/llama_index at b2edd07c60b16ade73f038e7b6356b62046ac87d.* (n.d.). GitHub. Retrieved July 10, 2025, from https://github.com/run-llama/llama_index

⁶² LlamaIndex - LlamaIndex. (n.d.). Retrieved July 10, 2025, from <https://docs.llamaindex.ai/en/stable/>

⁶³ *SimpleDirectoryReader - LlamaIndex.* (n.d.). Retrieved July 10, 2025, from https://docs.llamaindex.ai/en/stable/module_guides/loading/simplydirectoryreader/

⁶⁴ *Indexing & Embedding - LlamaIndex.* (n.d.). Retrieved July 10, 2025, from <https://docs.llamaindex.ai/en/stable/understanding/indexing/indexing/>

⁶⁵ *Examples - LlamaIndex.* (n.d.). Retrieved July 10, 2025, from <https://docs.llamaindex.ai/en/stable/examples/>

2.8.2. Langchain

LangChain is a framework to build context-aware and logic-driven applications using LLMs. It offers an extensive set of building blocks that can be composed into chains, agents, or memory-based systems. While LlamaIndex is more focused on data LangChain is more focused on the orchestration part.⁶⁶

2.8.2.1. Key Capabilities

- **Chains & Templates:** With LangChain you can easily create prompts or embedding system. LangChain use Jinja-like templates support partials and few-shot examples.⁶⁷
- **Agents & Tools:** LangChain has dozens of pre-made tool agents for SQL, Bash, and web research and you can also create your custom agent.⁶⁷
- **Memory Modules:** LangChain offers persistent conversation using the context of the discussion with, for example, ConversationBufferMemory.⁶⁷
- **Tracing & Evaluation:** There Native integration with LangSmith, LangChain's SaaS for observability, debugging, and dataset evaluation.⁶⁷
- **Integration with External Tools:** With LangChain you can have a lot of integration with vector databases (e.g., Chroma, FAISS), cloud storage, APIs, and other frameworks with LangGraph.⁶⁷

2.8.2.2. Advantages

- **Granularly control:** LangChain offers a basic set of tools that enable you to build your system with the granularity you want.
- **Massive community:** LangChain has an enormous community and a lot of existing projects with LangChain already integrated which is a plus when you need to create a new application.
- **Good ecosystem:** LangChain is really well integrated with LangGraph, and LangSmith allowing them to work seamlessly.

2.8.2.3. Downsides

- **Not as flexible:** The flexibility in LangChain can lead to architectural problems if the best practices are not followed
- **Verbose Syntax:** LangChain can become tedious to maintain with deeply nested chains

⁶⁶ *LangChain*. (n.d.). Retrieved July 13, 2025, from <https://www.langchain.com>

⁶⁷ Introduction |  LangChain. (n.d.). Retrieved July 11, 2025, from <https://python.langchain.com/docs/introduction/>

2.8.3. OpenAI SDK agent

OpenAI was released on 11 March 2025 a new set of tools and libraries to build agents, which enable developers to create agentic application system with an official SDK. This new framework aims to simplify the development of applications that use LLMs with tools.⁶⁸ This new framework is available in both Python and JavaScript.

2.8.3.1. Key Capabilities

- **OpenAI Tool Calling:** Native support for function/tool calling for all the OpenAI models. There are also built-in tools for web searching.⁶⁹
- **Multi-step Agent:** The agent can have multi-step reasoning and between each step the general context is preserved. It is also possible to run tools in a strict particular sequence.⁶⁹
- **OpenAI-native integration:** OpenAI agent is developed by OpenAI for OpenAI-hosted tools which creates nearly perfect integration.⁶⁹
- **Simplified Agent API:** Developers can register tools and define agent behaviors using simple decorators and configuration without managing prompts or chains manually.⁶⁹
- **Observability and debug:** OpenAI has some development that allows you to see the details of each step and also have the possibility to display a graph of the chain of thought.⁶⁹

2.8.3.2. Advantages

- **Good documentation:** The documentation provided with the SDK is clear and not too long with concrete examples.
- **Few lines of code:** With OpenAI SDK agent it is possible to create a completely new tool in just a few lines of code.
- **Easy to use and learn:** work directly with OpenAI-hosted tools and models and it is possible to develop easily a new agentic app.⁶⁹

2.8.3.3. Downsides

- **Limited compatibility:** OpenAI SDK agent currently only works with OpenAI model and has no possibility to implement other models or open-source models.
- **New SDK:** OpenAI is a new SDK that was only released a few years ago and has a smaller and less integration and plugins than other frameworks.

⁶⁸ New tools for building agents. (2025, November 3). <https://openai.com/index/new-tools-for-building-agents/>

⁶⁹ OpenAI Agents SDK. (n.d.). Retrieved July 11, 2025, from <https://openai.github.io/openai-agents-python/>

2.9. Available technologies

2.9.1. Ollama

Ollama is an open-source runtime model registry that lets you pull, run, and use powerful open sources large-language-model builds like Llama-4, Mistral, or custom fine-tunes. This enables you to run powerful models on your own machine with a single command. It packages everything, like the model weights, tokenizer, and settings for you so you can use a model by running one command. Ollama automatically also chooses whether it is necessary to run on your CPU or GPU, making it quick and efficient. This means you can access the latest AI models privately, without needing complex setup or relying on cloud services.⁷⁰ This is a clear plus in context of high confidentiality app or project.

2.9.2. Hugging face

Hugging Face is the main open-source platform for sharing operational open-source AI models. Through their Model Hub you can pull in one click over 600'000 different AI models. This model can range from small sentence-embedders to frontier-scale LLMs. This model can then directly be used using the hugging face CLI. What is important with Hugging Face is you can see all the parameters and details used by the model to be able to create new exact copy of this model. It also Host some of the major comparative analysis of Embedding model call MTEB.⁷¹

⁷⁰ ollama/ollama. (2025). Ollama. <https://github.com/ollama/ollama> (Original work published 2023)

⁷¹ Hugging Face – The AI community building the future. (2025, July 8). <https://huggingface.co/>

2.9.3. Infomaniak IA Tools

2.9.3.1. Company Overview

Infomaniak Network SA is a Swiss cloud service provider company-based Geneva. Founded in 1994, it has grown to become one of the largest independent cloud and web hosting companies in Switzerland. The main difference with their competitors is that Infomaniak is entirely employee-owned and does not rely on foreign investors. It allows Infomaniak to retain full independence and align closely with customer privacy interests.⁷²

Infomaniak's infrastructure is entirely located in Switzerland, and all its services are designed to comply with the Swiss Federal Act on Data Protection (LPD) as well as the European General Data Protection Regulation (GDPR). Its privacy policy explicitly excludes any form of data exploitation for commercial purposes, particularly regarding AI data processing.⁷³ Moreover, the company is a leader in sustainable digital services, operating data centers that use 100% renewable energy, and actively recover heat for municipal heating networks.⁷²

2.9.3.2. Overview of Infomaniak AI Tools

In response to the growing demand for sovereign and ethical AI infrastructure in Europe, Infomaniak has developed a suite of "AI tools" that are fully hosted and operated within Switzerland. These tools are based on open-source LLMs and open-source models for speech-to-text and image generation.⁷³

The core features of Infomaniak's AI offering include:

- **LLMs and Chat Assistants:** Models such as Mixtral 8x7B, LLaMA 3.3, Mistral Small, and Granite are made available through Infomaniak's infrastructure. Function-calling support is also in development.⁷³
- **Speech-to-Text:** Infomaniak offers a possibility to transform speech to text powered by Whisper v2 and v3.⁷³
- **Image Generation:** By using models such as **Photomaker v2** and **Flux Schnell**, users can create high-resolution images from textual prompts.⁷³
- **API Access & Web Integration:** Infomaniak offers an API that enables developers to integrate AI capabilities directly into their own applications. These tools are also embedded in Infomaniak's own platforms, such as kSuite, kMail, and kChat, where AI assists users in correcting, summarizing, and translating content.⁷³

⁷² About Infomaniak: our history and our vision of an eco-responsible cloud. (n.d.). Retrieved June 18, 2025, from <https://www.infomaniak.com/en/about>

⁷³ AI Tools - Integrate sovereign AI services for your applications. (n.d.). Retrieved June 18, 2025, from <https://www.infomaniak.com/en/hosting/ai-tools>

3. Methodology

3.1. Selection process

3.1.1. Language Model Selection

After reviewing the state of the art, we made the decision to test all the different models, explaining that it would be more informative than a more focused version. So, we have taken all the model explanations inside this bachelor thesis to incorporate into our benchmarking suite. This approach really allows us to have a global vision of the current market of AI models and ensure an accurate result.

To keep this large scope manageable, we have adopted a more automated and modular approach that will be explained inside section 3.2.12. In summary each part of the benchmark can be selected independently from the other one and it is possible to choose everything.

One of our original plans was to run a deeper benchmarking on some representative model. However, with some preliminary manual testing we rapidly see that a model that we considered performant with its general capabilities and shown result form will be performant here too, but we have seen that some better models can perform worse for our type of task. In short, testing all the models assures us that we will have an accurate and usable result for production usage later on.

3.1.2. AI Framework Analysis

3.1.2.1. Selection Criteria

To be able to identify the most suitable framework we have used the following evaluation criteria based on the project requirements. Also, not all the criteria above have the same relevance for our project; we have decided on the following ponderation that privileges the functionalities of the frameworks.

CRITERION	DESCRIPTION	WEIGHT
Data Integration	Ability of the framework to index various formats (PDFs, website, structured files)	3
External Tool Compatibility	Simplicity to connect external tools such as database and APIs.	2
Agentic Workflow Support	Capability to define agent behaviors, including creation of tools or a complete chain-of-thought	2
Documentation & Community	Quality of resources, examples, Usage on other projects, community overview	1
Extensibility & Modularity	Framework's architecture allows future extension such new tools, capabilities, multi-modal inputs	1
Ease of Use	Learning curve and simplicity to use read the code.	2

Table 1: Framework Selection criteria

3.1.2.2. Detail Analysis

LlamaIndex

Data Integration

LlamaIndex is particularly efficient for indexing diverse data sources such as PDFs, HTML pages, Notion documents, SQL databases, and APIs with it numerous connectors and node parsers⁷⁴. Its ingestion pipeline is flexible and supports real time updates and chunking strategies making it ideal for RAG applications.

LlamaIndex offers also a cloud environment llama cloud that allows you to store ingest and manage documents directly on a web interface. This cloud version uses LlamaParse which is one of the best parsers available currently. Unfortunately, this cloud version is paid for and LlamaParse works with credits that can rapidly become pricy for advanced parsing methods.⁷⁵

External tool callings

With its tool abstraction and Agent classes, LlamaIndex offers a simple and structured way and easy way to integrate external APIs and functions. It also supports OpenAI function calling and LangChain agents as plugins⁷⁶.

Agentic Workflow Support

LlamaIndex supports a large variety of workflows including multi-step workflow with memory and tool use. The AgentRunner and OpenAI Agent classes facilitate structured thought processes, planning, and tool execution⁷⁷. It also supports a large variety of models, making it a good choice in the context of rapid changes of capabilities and performance of models.

⁷⁴ LlamaIndex. (n.d.). Documentation. LlamaIndex. Retrieved June 22, 2025, from <https://llamaindex.ai/docs>

⁷⁵ LlamaParse: Transform unstructured data into LLM optimized formats — LlamaIndex - Build Knowledge Assistants over your Enterprise Data. (n.d.). Retrieved June 22, 2025, from <https://www.llamaindex.ai/llamaparse>

⁷⁶ Readthedocs - LlamaIndex. (n.d.). Readthedocs - LlamaIndex. Retrieved June 22, 2025, from <https://llama-index.readthedocs.io/zh/stable/>

⁷⁷ Agents - LlamaIndex. (n.d.). Retrieved June 22, 2025, from https://docs.llamaindex.ai/en/stable/module_guides/deploying/agents/

Documentation & Community

LlamaIndex offers full documentation with example notebooks, recipes, and tutorials. Generally, this documentation is full and complete but sometimes it can be a bit hard to find the solution for a simple problem rapidly rather than read a whole notebook. Also, LlamaIndex has a strong community with an active Discord and a GitHub. It is also used by multiple big companies that also contribute to the development of LlamaIndex.⁷⁴

Extensibility & Modularity

LlamaIndex is a very modular framework where developers can build custom receivers, embeddings, memory, agents, and more. This allows us to be really flexible to add extensions and multimodal support. For instance, one of the prototypes given to us as a PoC for the project have used a custom agent system completely to be able to integrate and connect new Infomaniak IA tools that hadn't any official support by LlamaIndex⁷³.

LangChain

Data integration

LangChain is built in with a lot of documents loader that can ingest various formats such as PDFs, HTML, Markdown, Notion and more. LangChain will automatically convert them into a standard document format so they can be indexed later on. The loaders are available both in python and JavaScript and are really easy to invoke with only simple load() method.⁷⁸

External Tool Compatibility

With LangChain you can build and create custom tools for services that didn't have direct integration. LangChain contain also a lot of third-party tools such as search, calculator, vector DBs and more that can embed LangChain tools inside other frameworks such as the OpenAI Agents SDK or LlamaIndex.⁷⁹

Agentic Workflow Support

LangChain offers multiple types of agents including simple ReAct-based model⁸⁰ to multi-step agents. It also supports built-in memory persistence. This allows developers to rapidly configure agents that can react to the user input and execute the tool you want. With LangGraph it is possible to create entire workflow that is ideal for complex use cases like dynamic data pipeline.⁸¹

Documentation & Community

LangChain provides a lot of documentation with clear examples, guides and tutorials. Their documentation is easily accessible, covering basic as well as advanced use cases with simple examples and detailed explanations. LangChain as a large and active communities in the LLM tooling, with active forums, including discord and GitHub, and a lot of real-world example of app build with LlamaIndex.⁸²

Extensibility & Modularity

LangChain has a highly extensible and modular design, allowing us to easily integrate different language models, embedding methods, vector databases, or external services services. It is easy to add new tools or services. LangChain is really modular so you can replace a component seamlessly based on your need.⁸³

⁷⁸ Document loaders |  LangChain. (n.d.). Retrieved July 2, 2025, from https://python.langchain.com/docs/integrations/document_loaders/

⁷⁹ How to do tool/function calling |  LangChain. (n.d.). Retrieved July 2, 2025, from https://python.langchain.com/docs/how_to/function_calling/

⁸⁰ ReAct agent is a AI agent that use the concept of reasoning and acting framework to combine a chain-of-thought with external tool uses. This enable the model to resolve more complex task in agentic workflow. What is a ReAct Agent? | IBM. (2025, March 18). <https://www.ibm.com/think/topics/react-agent>

⁸¹ Overview. (n.d.). Retrieved July 11, 2025, from <https://langchain-ai.github.io/langgraph/concepts/why-langgraph/>

⁸² The LangChain Community. (n.d.). Retrieved July 14, 2025, from <https://www.langchain.com/community>

⁸³ Providers |  LangChain. (n.d.). Retrieved July 14, 2025, from <https://python.langchain.com/docs/integrations/providers/>

Ease of Use

LangChain has an intuitive API and helpful built-in components that make quick prototyping easy, even for beginners. However, when more complex or customized workflows

are built, their structure can be complex and restrictive. This led to difficulties in debugging augmenting the time of development.

OpenAI SDK Agent

Data integration

OpenAI SDK agent does not provide built-in document loaders or connectors for various file formats. So in order to use the file we need to have a preprocessing step that can require to use LlamaIndex or LangChain. This is really a downside for our project as we need to have an easy way to process multiple types of files.⁸⁴

External Tool Compatibility

OpenAI SDK offers easy support for integrating external tools via the function-calling API included within OpenAI's models. This allows developers to define the functions, parameters, the models' capabilities with external APIs. However, the SDK itself doesn't provide detailed implementations or extensive libraries such as LangChain or LlamaIndex.

Agentic Workflow Support

OpenAI SDK agent supports Function calling capability provides. While suitable for simple workflows and straightforward external integrations, the OpenAI SDK agent approach is generally less flexible for multi-step or highly complex agentic workflows than explicitly agent-oriented frameworks. If the required functionality involves advanced execution chains, planning or complex decision-making, developers often supplement it with more specialized tools or frameworks.⁸⁵

Documentation & Community

OpenAI SDK agent provides comprehensive documentation with some clear examples that help the developers build their agent. However, as this SDK is new, there is not a strong community around this new SDK which can pose some problems.

Extensibility & Modularity

Open AI SDK agent features are limited. First this new SDK is relatively new, which means less integration or community to support the tool and secondly it is designed explicitly for interaction with OpenAI API. This clearly limits the possibilities of external integration.

Ease of Use

With an Open AI SDK agent, you can build a complete chatbot with tools easily with minimal complexity and code. This SDK is really easy to use and develop with prefabricated tools and agents. Starting with OpenAI SDK agent is clearly faster and easier than with LlamaIndex or LangChain.

⁸⁴ *OpenAI Agents SDK*. (n.d.). Retrieved July 15, 2025, from <https://openai.github.io/openai-agents-python/>

⁸⁵ *Agents - OpenAI Agents SDK*. (n.d.). Retrieved July 15, 2025, from <https://openai.github.io/openai-agents-python/agents/>

3.1.2.3. Comparative analysis

CRITERION	LLAMA INDEX	LANGCHAIN	OPENAI SDK AGENT
Data Integration	$5 \times 3 = 15$	$4 \times 3 = 12$	$1 \times 3 = 3$
External Tool Compatibility	$5 \times 2 = 10$	$5 \times 2 = 10$	$3 \times 2 = 6$
Agentic Workflow Support	$4 \times 2 = 8$	$5 \times 2 = 10$	$2 \times 2 = 4$
Documentation & Community	$4 \times 1 = 4$	$5 \times 1 = 5$	$4 \times 1 = 4$
Extensibility & Modularity	$5 \times 1 = 5$	$4 \times 1 = 4$	$2 \times 1 = 2$
Ease of Use	$4 \times 2 = 8$	$3 \times 2 = 6$	$5 \times 2 = 10$
TOTAL	50	47	29

Table 2: Table 2: Comparative analysis LlamaIndex LangChain OpenAI SDK Agent
 Autor's source

In detail we can say that LlamaIndex is more suitable for applications that need to get information from documents and LangChain is more suitable for applications with a big step-by-step workflow.

The key difference that makes LlamaIndex the best AI framework for our purpose is that it is easy to learn and use. LlamaIndex offers enormous documentation that can be too much but offers a lot of examples that help us to use it effectively.

For OpenAI SDK agent the 2 biggest weaknesses he has are, first, only OpenAI models are compatible with this framework which is an enormous downside in a field where AI models are constantly evolving, and new models from different companies are breaking records and are more powerful every week. The biggest advantage of OpenAI SDK agent is its simplicity to use you can create a working multi-step agent rapidly.

To sum up, as we need for our project an LLM that can perform RAG operations and be able to understand GIS queries, we need to have a framework that can do both. As LlamaIndex has a superior connector and a better parsing feature it makes it a great choice for our project.

3.2. Prototype Implementation

This section will present to you in detail how we implemented the prototype for this bachelor thesis. The first 2 subsections will be here to give you a general idea of the project and then the following section will be more focused on all the different aspects of the project.

3.2.1. Technologies

This section will enumerate the key technologies and building blocks used to create the prototype separated by their respective groups. Each subsection will describe roughly what the key implementations of the different technologies used are.

3.2.1.1. LlamaIndex

3.2.1.2. Embeddings

On this prototype multiple embedding models from different providers have been used and configured. To find more information about this model please refer to the section 2.6. Those embedding models have been used as a base for our RAG queries and user prompt understanding.

3.2.1.3. Vector Database

For this project the Chroma DB vector database has been implemented to allow us to retrieve with a high accuracy the information needed to answer the citizen questions. This approach is more flexible and scalable than only using basic index files. It really addresses the problem of getting information more easily from multiple sources.

3.2.1.4. Fronted

The front end of the prototype was built using flask template, with HTML JavaScript and Tailwind CSS for styling. This approach enables the creation of a user-friendly interface that can accommodate both citizen skinning information's and also developers managing the system with an administration panel.

3.2.1.5. Backend

The backend runs on Flask 2.3, deployed behind Gunicorn. The prototype routes are organized by function to help to support all the required features and endpoints of the project. The backend has been developed using Python and integrates the LlamaIndex framework for the AI management part.

3.2.2. General architecture

To be able to better understand the project the best way is to present to you graphically in a stylish architecture diagram the main interaction of the project. Here below will be displayed the full architecture. The next sub section will describe more in detail this architecture.

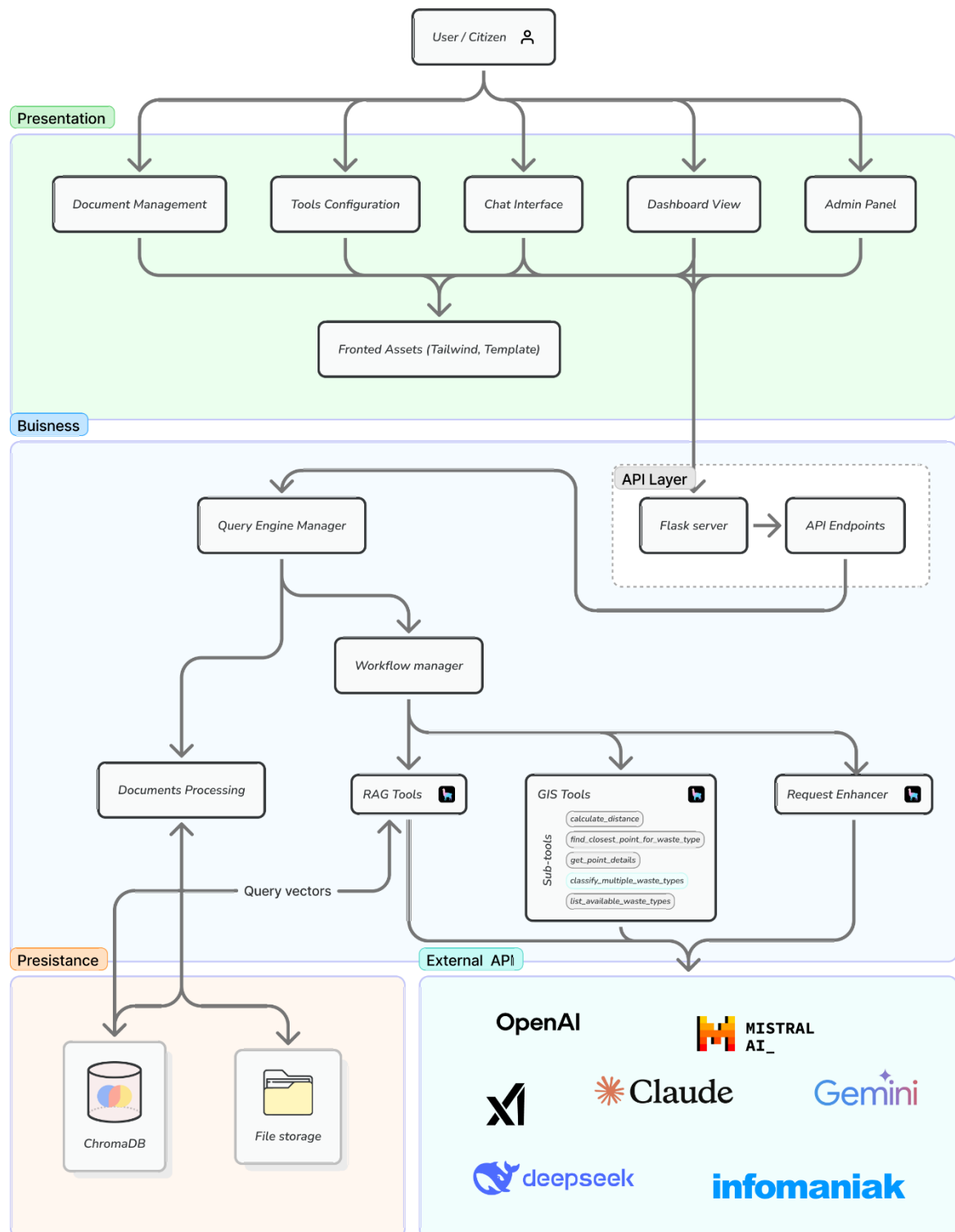


Figure 25: Prototype Architecture Diagram
 Author's source

3.2.2.1. Presentation layer

The presentation layer is responsible for all the interaction and visual elements of the prototype. It consists of web-based interfaces built with HTML, Tailwind CSS, and JavaScript. It is composed of a dashboard and interfaces where we can manage our index document discussion with AI agents or configure the embeddings model. This layer ensures great experience for the final users.

3.2.2.2. Business layer

The business layer contains the core logic and processing component of the application. It is implemented in Python and has an API endpoint, workflow orchestration, and integration with external services such as the LLMs or GIS tools. This layer also handles the management of document processing and all the index management.

3.2.2.3. Persistence layer

The persistence layer is responsible for storing and retrieving data used inside our application. It manages our files linked to ChromaDB and all the uploaded files. All the data is stored locally inside the Docker instance.

3.2.3. File structures

The following section will explain in detail the file structure of the project and have a deeper look into some of them like the configuration file.

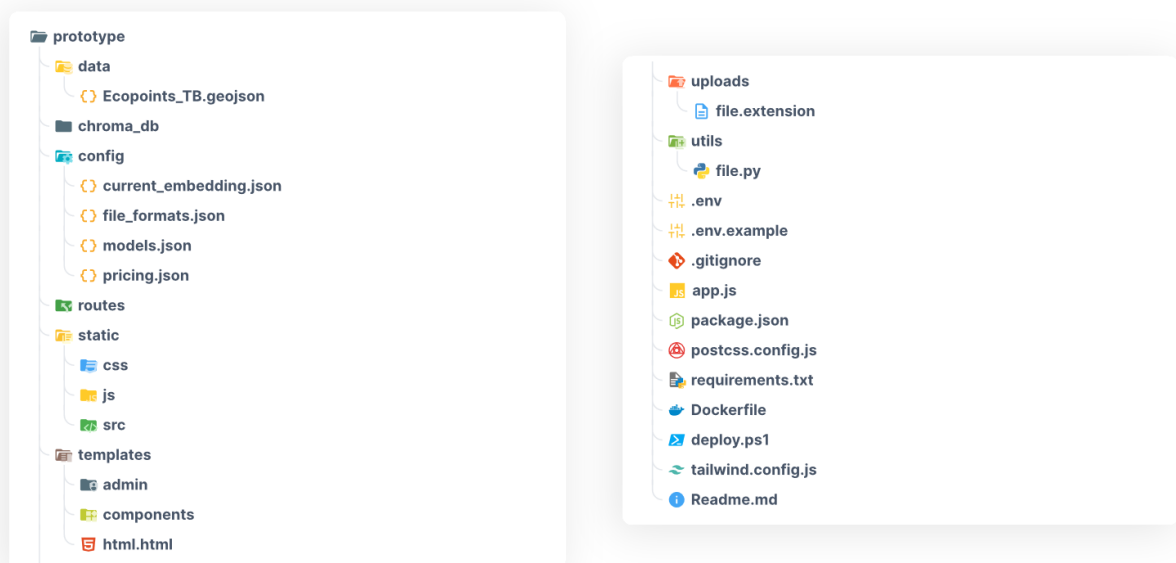


Figure 26: File structures

Author's source

The file structure of the project is quite easy to understand. First there are the persistence folders that contain:

- **The data folder:** this folder contains the GeoJSON file with all the collection points
- **Chroma_db folder:** This folder contains the actual ChromaDb.
- **Config folder:** contains all the configuration files of the project like the file format or models
- **The upload folder** contains all the files uploaded on the prototype.

Then there are all the files related to the general project and deployment with:

- **.env file**: that contains all the API key
- **Package.json and requirements.txt** for both the NodeJS and python dependencies
- **Post.css.config and tailwind.config.js** for the Tailwind CSS configuration
- **Dockerfile and deploy.ps1**: used for the project deployment
- **Readme.md**: contain information about the project and how to deploy it

Lastly there are the actual files of the project with first the front that is located inside the template folder and inside the "static/js" folder for the JavaScript logics. The business layer is held first with the route folder and then inside the utils folder.

3.2.4. Configuration files

3.2.4.1. Current embeddings

This file is automatically generated and stores the current embedding model used by the application

3.2.4.2. Files format

This file is dedicated to listing all the extensions supported by the prototype to be used both in front as a display and on the backend for verification. Please note, however that if you add a new file format you might need to change code. This file is more there to avoid having a consistency between the front and the backend.

3.2.4.3. Models

This is the main configuration file of the project. This file allows you to configure a new provider and their related model directly inside the JSON file without the need to modify the code, except if you know the new provider you add doesn't have direct integration like the Infomaniak models.

In this file you can also configure the list of all the embedding models available. Now if we take a closer look at the Json file:

```
"mistral": {
  "name": "Mistral AI",
  "library_import": "llama_index.llms.mistralai.MistralAI",
  "base_url": null,
  "default_params": {
    "temperature": 0.1,
    "max_tokens": 10000
  },
  "models": [
    {
      "id": "mistral-medium",
      "name": "Mistral Medium",
      "category": "chat",
      "show_in_normal_chat": false
    },
    {
      "id": "devstral-small-2505",
      "name": "Devstral Small",
      "category": "chat",
      "show_in_normal_chat": false
    }
  ]
},
```

Here for each new provider, you want to add you need first to set it a display name for the provider and then add the exact library user for this provider. This will allow the prototype to test if this library is installed and inform the user if it is not installed.

Then there is the default parameter section where you can define all the normal parameters of an IA like the number of tokens or its temperature.

The last section of this Json is the model itself and here for each model you have to define the ID of the model it's display name, category, and if this model is usable for the citizen or for the administrator.

With this configuration we can easily add a new model of an existing provider or add a new provider easily. This is a major point considering the current rapidly evolving subject with new models like Grok 4 or GPT 5 that will be soon announced and available.

For the model and provider part this JSON also has 2 more functionalities. First we can determine which model use as a fallback if there is any error with the call of the model during an answer with the following code

```
"fallback": {
  "provider": "openai",
  "model": "gpt-4.1-nano"
},
```

And secondly it is possible to set the default model at the start of the application.

```
"default_model": "gpt-4.1-nano",
```

The next part of this config file is the embedding model configuration that is exactly the same as above but under the “embedding_providers” sections. As well you also have a fallback feature and a default embedding model configuration. Here below are the configuration examples of Hugging Face embedding models.

```
"huggingface": {
  "name": "HuggingFace",
  "library_import": "llama_index.embeddings.huggingface.HuggingFaceEmbedding",
  "default_params": {
    "max_length": 512,
    "normalize": true
  },
  "models": [
    {
      "id": "intfloat/multilingual-e5-large-instruct",
      "name": "Multilingual E5 Large Instruct",
      "dimensions": 1024
    },
    {
      "id": "sentence-transformers/all-MiniLM-L6-v2",
      "name": "All-MiniLM-L6-v2",
      "dimensions": 384
    }
  ]
},
```

Here the only real difference is the information about the model where we only set the dimensions.

3.2.5. RAG System Implementation

The RAG (Retrieval-Augmented Generation) agent is designed to enhance the chatbot's capabilities to answer complex queries. This system uses the documents ingested by the prototype to enhance the quality of the responses.

3.2.5.1. Overview

First, when the query is sent by the user the workflow will determine the type of the query using an LLM call to determine the user's language of the query. Then we process like that:

1. First, we determine which tool is enabled to route the user prompt to the right process
2. One inside the RAG method we start by getting the actual query engine available with all the documents indexed. Here below is the code to get the query engine:

```
llm, actual_model_info = self._get_llm_for_model(model)
query_engine = index.as_query_engine(similarity_top_k=3, response_mode="compact",
llm=llm)
```

3. Once we have got the query engine, we only need to pass the questions of the engine to be able to generate an answer to the questions.
4. Then once the response is generated, we format it to be able to extract the relevant information like the actual answer and how many files are used to generate the answer.
5. Finally, we check if the answer generated is valid. For instance. we do simple test like if the response generated as at least 50 word and doesn't contain "cannot find". These 2 words are a kind of default response when a model does not find any relevant information.
6. At the end we translate the answer into the user's language and return it to the front end.

Note during this entire process we keep track of all the tokens used and the total time passed. This implementation of the RAG is really basic thanks to the ChromaDB implementation that allows the LLM to find relevant information about the question without the need to create a complex system of metadata for the documents indexed.

3.2.6. GIS Integration

3.2.6.1. Overview

The GIS agent is developed to assist citizens in efficiently locating waste collection points using an intelligent tools-based LLM agent. This integration integrates multiple tools and steps to ensure accurate responses.

3.2.6.2. Process Step breakdown

When a user asks a question on the chatbot and the GIS tool is activated, the system will start with the following steps:

7. Verification and validation step

When the query is received, we have to test multiple points to determine if the query is correct and can be processed. Here are the following points:

- **GIS query detection:** This identifies if the query is related to geographical interaction about waste-related topics using an LLM query with a system prompt. Here below is an example of a part of this system prompt:

```
Context: [ChatBot Context]
```

```
GIS-related queries include:
```

```
Examples of GIS queries: [example of type of GIS question]
```

```
NON-GIS queries (GENERAL information): [Example questions about non-gis queries]
```

```
IMPORTANT: [More information about what the llm have to do]
```

```
Respond with ONLY "YES" if this is about waste disposal, recycling specific items, or asks for location information, or "NO" if it's general/abstract information.
```

- **Language detection:** Then to be able to answer in the user's language we need to determine it using once again an LLM call. Here below is the system prompt used:

```
Detect the language of this text and respond with ONLY the language code.
```

```
Text: "{query}"
```

```
Respond with ONLY one of these codes:
```

- "en" for English
- "fr" for French
- "de" for German

```
Examples: [3 examples to guide the AI]
```

- **Coordinate validation:** The last step we perform is the verification of whether the GIS coordinates are present and valid. This is done with a simple method "validate_coordinates" that verifies that.

Once all the verification parts have been processed, we now start the real workflow step that will allow us to generate a precise answer for the citizen.

8. Waste Type classification

First to be able to get the right collection point we need to determine the waste type of the user's questions. To be able to do this we need to use an LLM that will classify the list of waste of the query based on the list of waste types.

```
Analyze this user query and determine if it's about waste disposal or recycling:
"{item_description}"
```

```
Available waste types in our system: {'', '.join(self.available_waste_types[:50])}
```

```
Task:
```

1. First determine if this query is about waste disposal, recycling, or environmental services

2. If YES, identify which specific waste types from the available list match the query
3. If NO, respond with "NOT_WASTE"

Examples: [some example]

IMPORTANT: Use this exact response format:

- If NOT about waste: Return exactly "NOT_WASTE"
- If about waste: Return "WASTE_TYPES:" followed by the exact waste type names from the available list, separated by semicolons

Format examples: [some examples]

Here we had to add more steps to see the system to be able to have a well-formatted answer. Once we have the list of all the waste types of the query, we now need to find collection points that match these criteria.

9. Collection Point Finding

The next step is to find the closest point for each type of waste listed and to be able to do that we use a method that iterates through the whole list of collection points and then for each point, gets it filtered by the type of waste and then finds the closest point with a method dedicated to calculating the real distance between 2 points.

Here we avoid any use of AI to keep the whole list of points which are confidential safe. Here, even if the LLM calls this function he will only be able to see the output.

10. Response Generation

To be able to generate a correct answer for the end user we need to perform these following steps in order:

- Get detailed information on all the collection points found.
- Calculate the distance between the point and the position of the user.
- Create the structured output
- Add tips or error management
- Create the answers

11. Map Visualization

Now that the data has been generated, we need to display the actual points on the map. To do that we will take these steps in order:

- **Map Initialization:** Creates a Leaflet map centered on Switzerland
 - **Marker Creation:** Uses the custom marker from the GeoJSON to display it on the map
- Display of the point: Shows collection points found by the queries and nearby alternatives but only for the administration page. The point will also contain all the other waste they accept and their relative information such as their name or postal code. Here below is an example of the map pin popup:

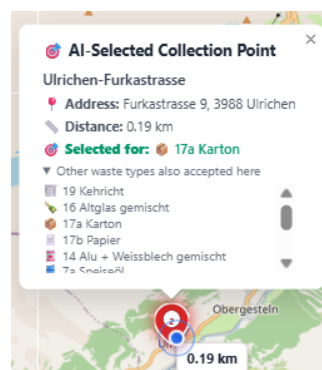


Figure 27: Map integration pin popup example

Author's source captured on the 15 July 2025 on the chat admin interface

- **Interactive Features:** Then we add the last missing information such as the line between the points and the position of the user or the distance between those points

3.2.6.3. Confidentiality

As the entire list of points is confidential and should not be exposed inside the LLM's call we would like to describe this part of the implementation in even more detail. Inside our code we only perform 3 LLMs calls, first with the type and language of the query and then the categorization of the type of waste. Everything else is only managed by proper code and no other LLM calls are performed.

During our first test we tested another way to perform this with LLM workflow and tool registrations, and we have also done research to understand the limits of vision of LLM with tools. Our research confirms that if an LLMs uses a tool it can only see the input of the tools and it provides results but can never see its internal implementation. Here we assume that the tool use does not contain any other call on LLMs. (Unlu, 2023)

As we need to calculate the distance between points of similar waste type and the position of the user, we can define tools as follows:

- A tool that lists the ID of places that have the possibility to recycle this type of waste
- A tool that calculates the distance between a list of recycling points and the user and finds the closest one.
- A tool to get the details of a point based on its ID.

Like that the full list of geographical data points is safeguarded behind functions.

For the prototype we hadn't set up this solution as we have found that it is unnecessary to have an IA that manages a static workflow as the detection of localization is step that always has the same order. This new approach first saves a bit of cost and time and secondly assures that no AI is used at all for this part of the project except for the necessary things like the detection of waste or language.

3.2.7. Combined tools

When both GIS and RAG mods are activated on the chatbot the chatbot will just perform the 2 different workflows independently and then at the end we just append the GIS result at the end of the RAG answer.

We have really kept these combined modes as simple as possible to have consistent results with our benchmarking. If we have added a new LLM step to wrap up the answer it may create inconsistencies and will take even more time to generate the answer for the user. As the time of generation is an important concern, we have just aggregated the 2 answers.

3.2.8. Embedding management page

This page is relatively straightforward it is used to manage the embedding model for our chatbot. On this page we can only do one thing select an embedding model and ask the system to recreate the indexes with these new embeddings.

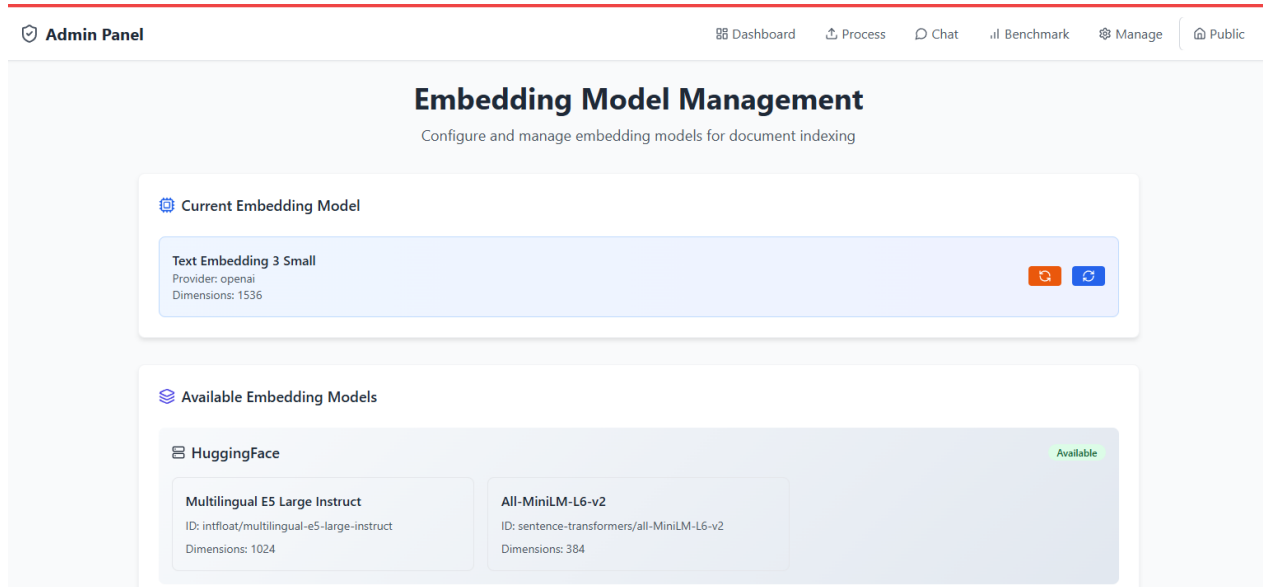


Figure 28: Embedding model page
 Author's source captured on the 08 July 2025

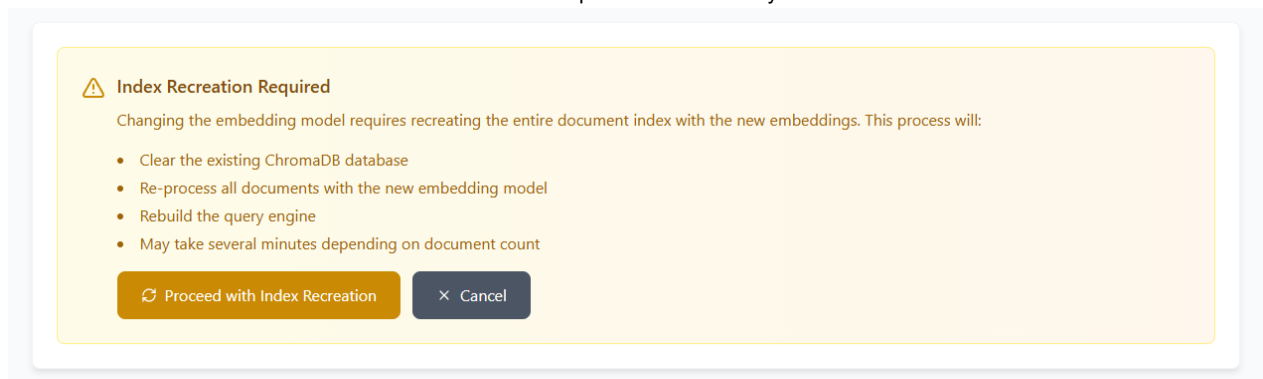


Figure 29: Embedding model Information box
 Author's source captured on the 08 July 2025

This page has one API that will send the current embedding selected on the backend to recreate the index.

3.2.8.1. Backend

When the user selects a new embedding and clicks the process button we perform these steps:

1. First, we call the method set embedding model that will do the following step:
 - Store the current number of documents for reference
 - Create the new embedding model
 - Check if we need to recreate ChromaDB and if the number of dimensions of the new model has changed.
2. Then we call the method to recreate the collection of indexes with the new embedding model.

3.2.9. Index Management pages

The index management on this prototype is mainly composed of 2 pages. The first page is the process page dedicated to the ingestion of websites or various other sources and the index management page is where the index itself can be managed by deleting a document or deleting the entire index.

3.2.9.1. Process page

The process page is the most important page of the RAG system because it is the only way we can process documents into our system. This page is composed of 2 main parts a website process part and a document process part. Here below we will talk more in detail about these 2 parts.

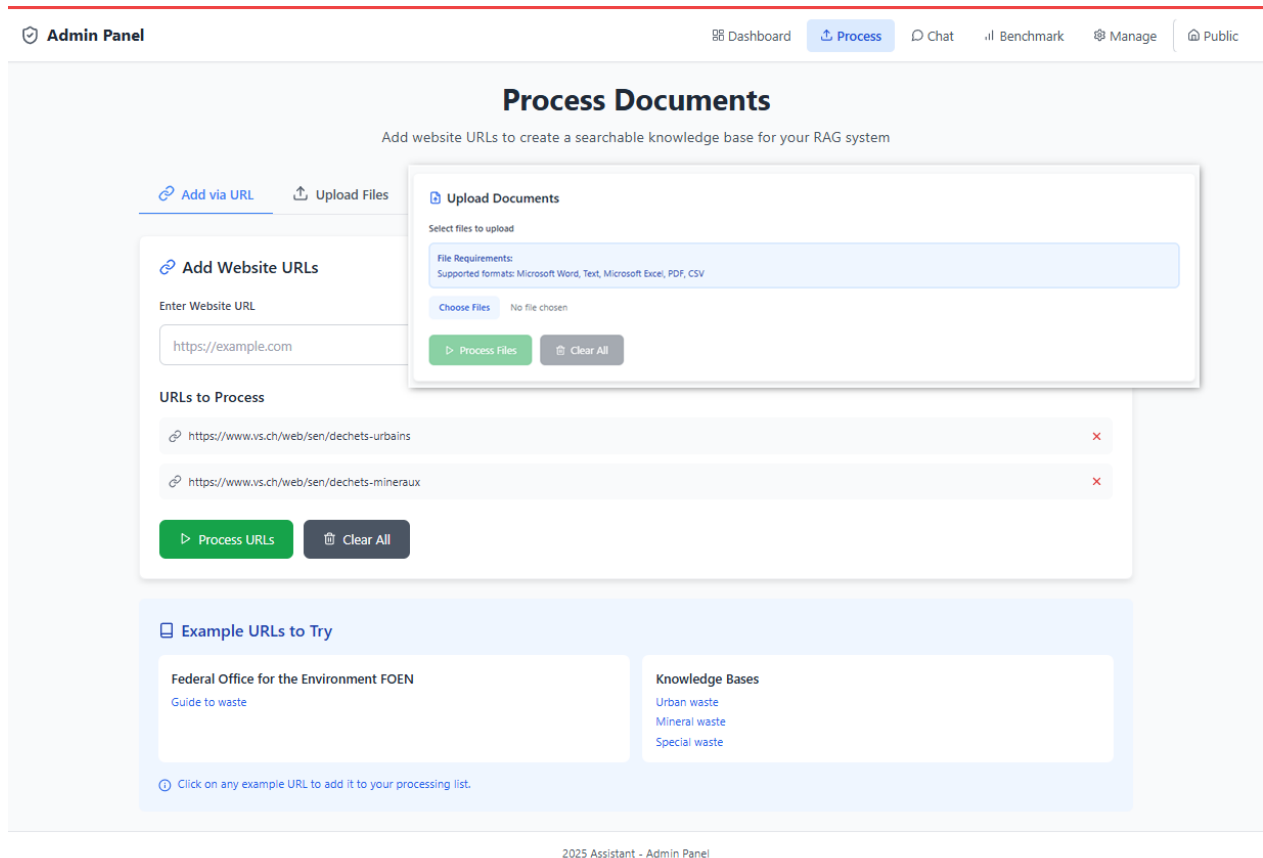


Figure 30: Screen of the process page
 Author's source captured on the 08 July 2025

This page contains 3 main parts:

- **Process of Website Page:** With this interface you can add multiple URLs to process and then process them in bulk.
- **Process of Files:** On this part of the interface, you can select multiple supported files and then process them.
- **Quick access to some URL:** This section has been developed for the purpose of the prototype to be able to easily add again the main URL needed for the project during the testing of the index deleting and more.

API Endpoint Usage

This page has 2 main routes:

- **Process-files**: accept a list file to be processed
- **Process-url**: accept a list of URLs to be processed

There is also another API call “file-formats” that get all the supported file format of the application.

As the logic of the URL and files are drastically different, we prefer to have them separated to have better readability and follow of the process.

Backend

Files

1. First, we receive file list via the “/process-files” route that will give us a list of files.
2. Then we validate format: check each file’s extension against the supported list returned by the “/file-formats” endpoint. Files with no valid extension are rejected here.
3. Then we checked if the file already existed or not. If it is new, we proceed to ingestion of it; if it already exists or not, if it is unchanged, we skip, otherwise if it has changed, we create a new document and delete the previous document.
4. The next step is we parse and extract the text of the file using LlamaIndex’s SimpleDirectoryReader which delegates the parsing logic to format-specific parsers (PDF → PDFMiner / PyMuPDF, DOCX → python-docx, etc.). During this process we add some metadata like the timestamp of the process or the original name of the document.
5. Then when the document is finished being read and saved on our side we pass on the creation of the index for the file. What we do concretely is we take the new file create an index with “VectorStoreIndex” and then add it to the query engine.
6. Once all these points are made the newly uploaded file is ready to be queried by the LLMs

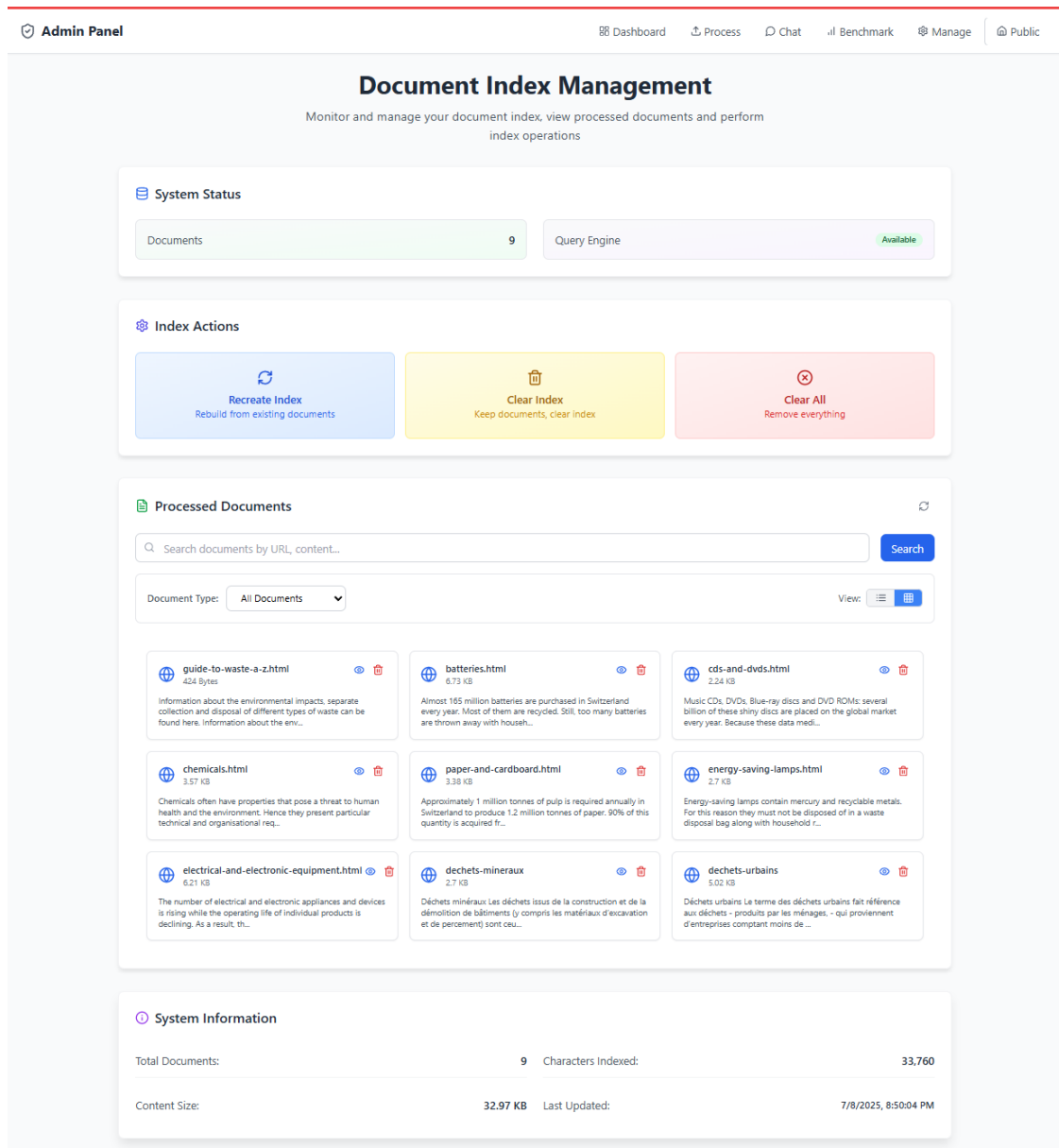
Website

The website process is really similar to the file process but as all the names and exact readers and parsers are not the same, we had to create a dedicated method for the website. The website actually uses the Trafilatura package that allows us to get information from websites. The website process is also easier as there is only one type of file to manage

We have decided to keep the file relatively simple, such as not creating chunks or more advanced parsing methods as the chabot must be able to get general information for his context not detailed information, such as exact sentences to respond.

3.2.9.2. Manage index page

The manage index page is a page to manage in detail all the index stores inside our application. This page offers multiple features such as deleting the whole index to see the full detail of an index or being able to delete the entire index or a specific one



Document Index Management
Monitor and manage your document index, view processed documents and perform index operations

System Status

- Documents: 9
- Query Engine: Available

Index Actions

- Recreate Index**: Rebuild from existing documents
- Clear Index**: Keep documents, clear index
- Clear All**: Remove everything

Processed Documents

Search documents by URL, content...

Document Type: All Documents View: [List] [Grid]

Document Name	Size	Preview
guide-to-waste-a-z.html	424 Bytes	Information about the environmental impacts, separate collection and disposal of different types of waste can be found here. Information about the env...
batteries.html	6.73 KB	Almost 165 million batteries are purchased in Switzerland every year. Most of them are recycled. Still, too many batteries are thrown away with househ...
cds-and-dvds.html	2.24 KB	Music CDs, DVDs, Blue-ray discs and DVD ROMs: several billion of these shiny discs are placed on the global market every year. Because these data medi...
chemicals.html	3.57 KB	Chemicals often have properties that pose a threat to human health and the environment. Hence they present particular technical and organisational req...
paper-and-cardboard.html	3.38 KB	Approximately 1 million tonnes of pulp is required annually in Switzerland to produce 1.2 million tonnes of paper. 90% of this quantity is acquired fr...
energy-saving-lamps.html	2.7 KB	Energy-saving lamps contain mercury and recyclable metals. For this reason they must not be disposed of in a waste disposal bag along with household r...
electrical-and-electronic-equipment.html	6.21 KB	The number of electrical and electronic appliances and devices is rising while the operating life of individual products is declining. As a result, th...
dechets-mineraux	2.7 KB	Déchets minéraux Les déchets issus de la construction et de la démolition de bâtiments (y compris les matériaux d'excavation et de perçement) sont ceu...
dechets-urbains	5.02 KB	Déchets urbains Le terme des déchets urbains fait référence aux déchets - produits par les ménages - qui proviennent d'entreprises comptant moins de ...

System Information

- Total Documents: 9 Characters Indexed: 33,760
- Content Size: 32.97 KB Last Updated: 7/8/2025, 8:50:04 PM

Figure 31: Screen of the index page
Author's source captured on the 08 July 2025

This page has some main functionalities:

- Display of some key information system to see the number of document characters and the last update of the index
- Possibility to recreate the index for the debug purpose or being able to delete the entire index.
- It is possible to see the details of a specific index and be able to delete it.

API Endpoint

- **Status:** return the current status of the index if the index exists or not
- **Recreate index:** delete the actual indexes and recreate it with the actual embedding model.
- **Clear index:** delete the index of all the files but preserve the files
- **Clear all:** delete the index and delete all the files
- **List of documents + Filter:** show the detailed list of all the documents available and have some
- **Deletion of a document:** delete the document from the upload folder and delete it relative indexes.
- **See the details of a document:** Return the detailed content

Backend

Here we will not talk in detail about all the routes used as some of them like the get of the status or the list of documents are literally self-explanatory.

However, the route related to the indexes needs to have some more detailed information. First, when you want to recreate the indexes, first the program will remove all the actual indexes, check if the ChomaDB needs to be recreated to match the new number of dimensions and reindex the documents available.

The Clear all indexes are much simpler as it will just erase the whole index, and all the documents uploaded.

These functionalities are present because during the development of the prototype we have often had to recreate the index or remove everything to perform certain tests. These functions were helping us and can still help a developer if he needs to rapidly change the entire index.

The filter part for the document really helps to find the right document rapidly. We added this new part after we see that with our first version of this page without this function it was nearly impossible to find a document if you have more than 10 of them.

3.2.10. Chatbot interface

The admin chatbot interface allows us to have detailed and full control over all the document and model tools on the prototype. This allows us to have a complete interface that lets us test every aspect of the application.

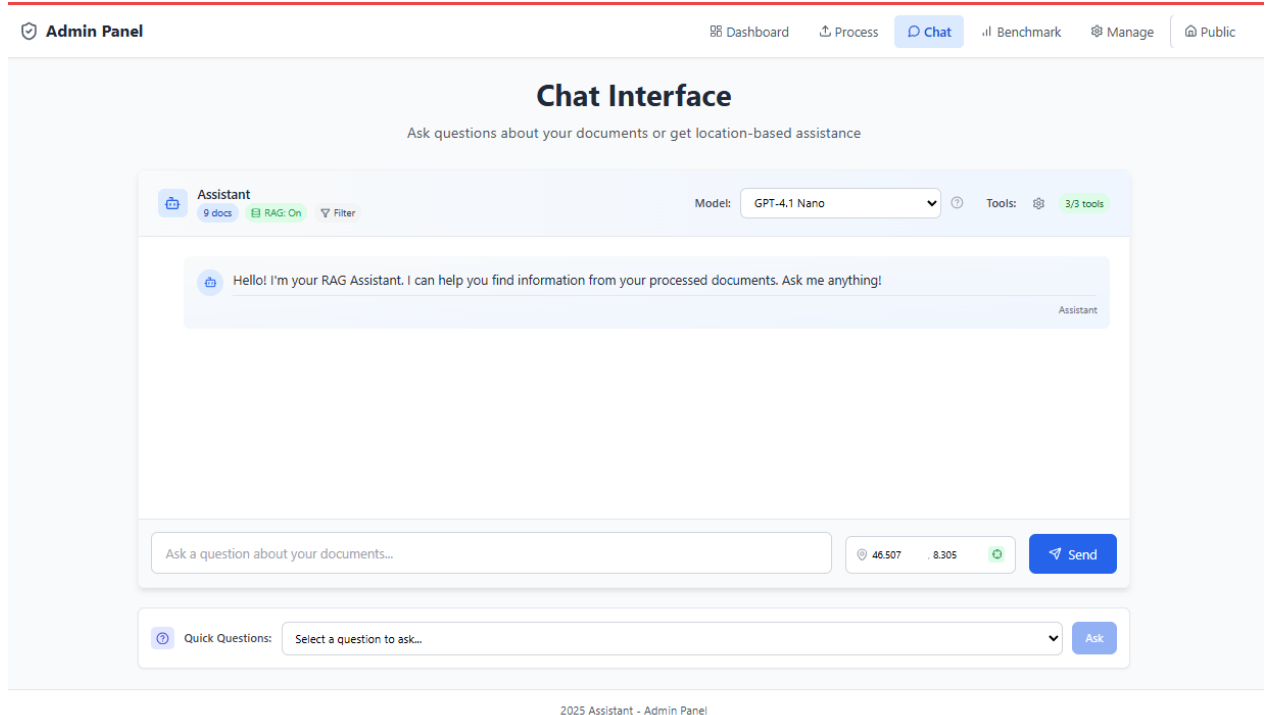


Figure 32: Screenshot of the Chabot admin page

Author's source captured on the 08 July 2025

This page really offers a lot of important functionality for our test. Here is the list of the main functionalities:

- Choose and filter documents used by the RAG system
- Choose of the model used to perform the generation of the response
- Choose of the tools activated or not
- The possibility to select your current localization or a determined localization
- Some quick questions to simplify the testing of the prototype

3.2.10.1. API endpoints

This page has a lot of API endpoints called and used:

- **Models list:** Fetches available models and providers.
- **Tools status:** Checks the status of the tools, including the GIS tool, to determine whether to show or hide related UI elements.
- **Status of the documents:** Retrieves the document count.
- **Provider availability:** Retrieves the API key status for different providers.
- **Chatbot message generation:** Sends chat messages to the backend for generating the answers.

3.2.10.2. Backend implementation

Here we will focus the explanation more on the chatbot endpoint as the other endpoints are straightforward and are just lists or status information. When the user writes a question on the interface and clicks on the send button, we perform some preliminary verification then we send all the information to the backend via the route "api/chat". Here below is the method used to call the route:

```
const response = await this.apiClient.sendChatMessage(message, selectedModel, mcpToolsEnabled, coordinates, selectedDocuments, enabledTools);
```

Here we really send everything to the backend, the actual message of the user, the selected model, all the tools enabled, the coordinates of the user, all selected documents, and the tools enabled.

Then at the route endpoint we get all this information and then do some verification to determine if every piece of data sent is correct to be able to call afterward the right tools. Here, depending on the tool activated it will start different modes. To see the detailed information about the tools see section 3.2.5, 3.2.6, 3.2.7.

Once the answer is generated the chat endpoint is responsible for extracting some information from the request and then returning the fully structured answer to the front. At each step of this process, we have a lot of verification to avoid the full app crashing.

3.2.11. Tracking of token usage

To be able to track all the tokens used by the whole workflow if a tool is activated or not, we have chosen to use a workflow tracker class that is passed at each step of every LLM call to keep track of the current count of tokens for each step, time passed, and the model used and the total pricing.

The actual method of adding this information is quite easy but it has been hard to implement as there are more than 10 LLMs calling at every part of the prototype. Here below is an example of an LLM call.

```
self.workflow_tracker.add_usage(  
    step_name=step_name,  
    input_tokens=input_tokens,  
    output_tokens=output_tokens,  
    cost=cost,  
    currency=currency,  
    model_id=model_id,  
    request_time=request_time  
)
```

This detailed data are also available on the export of the benchmark to be used for further analysis if needed.

3.2.12. Benchmarking Implementation

The benchmarking page is built to be really modular and allow the developer to quickly test, iterate, and modify the benchmark to obtain the result they want.

Model Benchmarking
Test and compare model performance across different question types

Benchmark Overview
Configure and run comprehensive model tests
17/17 models selected 187 tests selected ~10 min estimated

Model Selection
Choose which models to test

OpenAI

Model	Selected
GPT-4.1	☒
GPT-4.1 Mini	☒
GPT-4.1 Nano	☒
o3	☒
o4 Mini	☒

5 models

- gpt-4.1
- gpt-4.1-mini
- gpt-4.1-nano
- o3
- o4-mini

Select All Select None

Figure 33: Benchmark page choice of model

Author's source captured on the 15th of July 2025

This first part uses the configuration file of all the models to list all the models available and let the user choose which model to use to perform the benchmark. This is especially useful in the context of a new model release that would be compared to a previous benchmark.

GIS Location Tests
Location-based queries

Test Location
Latitude, Longitude
46.507 8.305

Questions

- ☒ Est-ce que je peux recycler des capsules a café?
- ☒ Kann ich Kaffee kapseln recyceln?
- ☒ Bonjour, j'aimerais jeter des piles et des lampes cassées ?
- ☒ Guten Tag, ich möchte Batterien und kaputte Lampen entsorgen ?

All None

RAG Document Tests
Document-based queries

Questions

- ☒ Où dois-je jeter mes vieux journaux et cartons d'emballage en Suisse ?
- ☒ Wo soll ich in der Schweiz alte Zeitungen und Verpackungskartons entsorgen?
- ☒ J'ai trouvé de vieux CD contenant des fichiers confidentiels et des traces de produits de nettoyage. Comment les jeter ?
- ☒ Ich habe alte CDs mit sensiblen Daten und Rückständen von Reinigungsmitteln gefunden. Wie kann ich sie entsorgen?

All None

Figure 34: benchmark page GIS and RAG questions

Author's source captured on the 15th of July 2025

Then you can choose between the predetermined GIS and RAG questions. Here we cannot modify the question and we need to rebuild the project to change a question, but it is not a problem as these questions are the standard questions for our benchmark.

General Knowledge Tests
General conversation

Questions All None

- ☒ How do you make an apple pie?
- ☒ What ingredients do you need for apple pie?
- ☒ Comment fait-on une tarte aux pommes?

Custom Questions
Add your own test questions

[+ Add Question](#)

No custom questions added
Click "Add Question" to create custom test questions

Figure 35: Benchmark page general and custom questions

As said above, the RAG GIS and general questions are not modified because of a dedicated custom section that lets you add the question you want to run the benchmark with. This is particularly useful to test a new tricky question or run a standard test if, for example, we have added new documents on the RAG and we want a rapid test to see if the different models are able to use this new context.

Benchmark Results
Test completed successfully

[Export Excel](#) [Clear](#)

6 Total Tests	6 Successful	5339ms Avg Response Time	\$0.0007 Total Cost
-------------------------	------------------------	------------------------------------	-------------------------------

Model	Category	Question	Response	Status	Response Time	Tokens	Cost
GPT-4.1 Nano openai	gis	Est-ce que je peux recycler des capsules a café?	**Assistant SIG - Points de collecte trouvés : ** ...	Success	9843ms	752	\$0.0

Figure 36: Benchmark page result table

Author's source captured on the 15th of July 2025

After we have run the benchmark, the interface will provide a global analysis of the run benchmark and display all the requests sent. This interface is really there to have a preliminary vision of the result rather than a complete analysis. To be able to do a complete analysis of the benchmarks you can download the Excel file containing all the detailed information about each query and some summary information. For more information about the look and information of the Excel file see Annex III.

3.2.13. Public part

This part is what the citizens will see when they use the chatbot. The chatbot page for the user has been simplified to remove some admin functionality:

- It is not possible to filter the documents use
- It is only possible to choose between the best model based on our benchmark
- All the tools are by default activated no need for manual configuration
- The quick questions have been removed
- The list of metrics like the number of token pricing info is deactivated as the user does not need this information
- On the map integration only, relevant points are displayed

With this simplified version citizens can just ask a question and the chatbot will answer it. Here below is an example of a question asked on this interface:

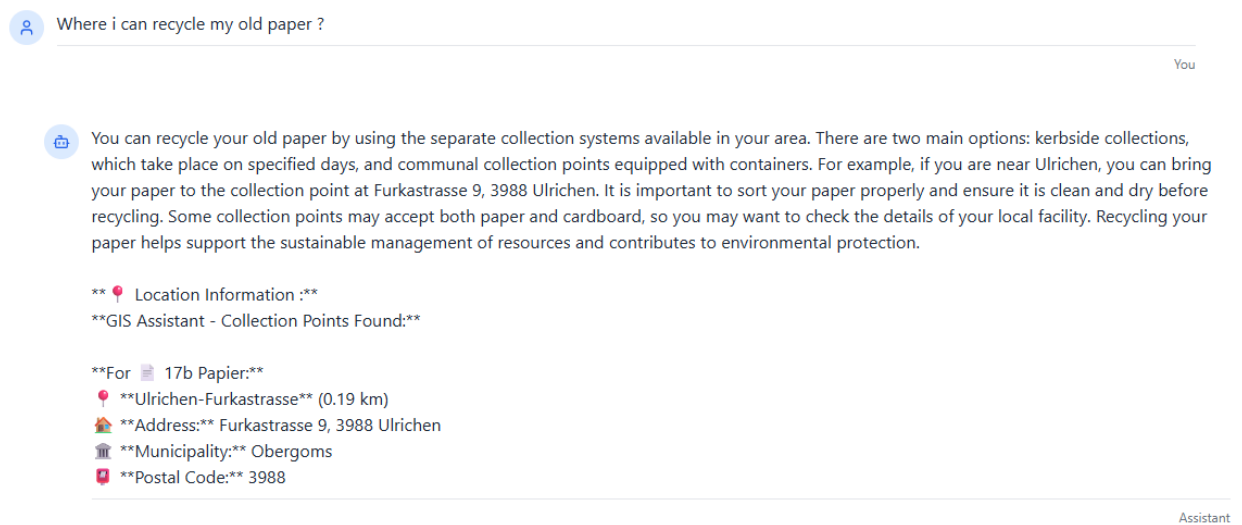


Figure 37: Screen of a answer of a question from the public chatbot page

Author's source captured on the 14 July 2025

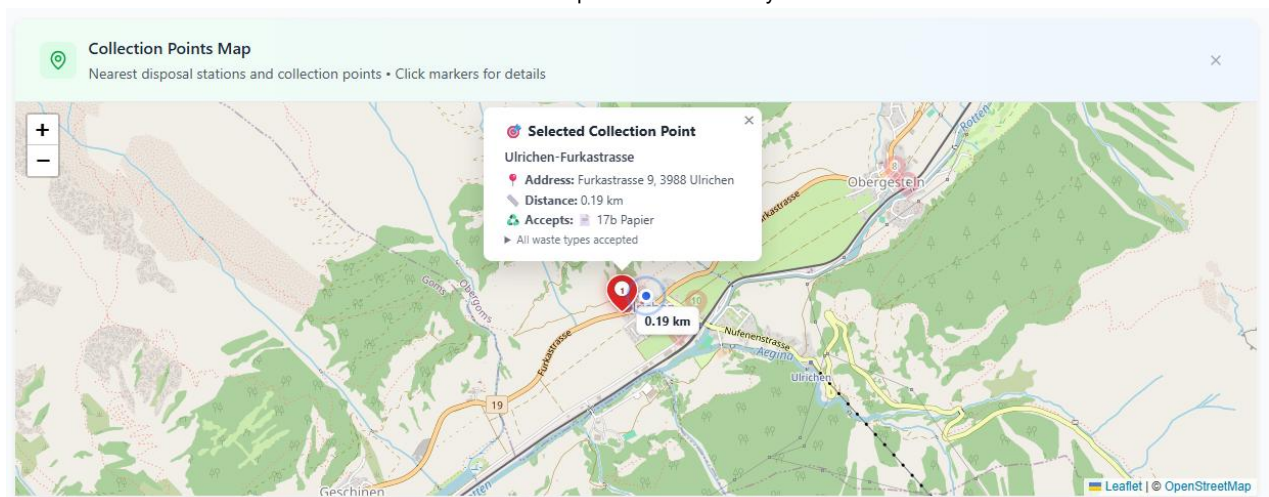


Figure 38: Map integration response example from public chatbot page with popup opened

Author's source captured on the 14 July 2025

Here the main goal of this page is to provide simple access to citizens without having too many options and losing the user. Also, this page uses exactly the same API endpoint from the administration page so if you need more detail see section 3.2.10.

3.3. Benchmarking protocol

3.3.1. Test questions

To identify the most effective model for our chatbot, we have designed a structured testing protocol to try to mimic a realistic use case encountered by citizens in a multilingual environment (German and French). The evaluation process will focus on 2 main aspects:

- **GIS Integration:** Test the chatbot's ability to understand spatial queries and return geolocated information accurately.
- **RAG Quality:** Evaluate the expected result from the chatbot

The main goal is not to test the model's reasoning process but more to see how each model interacts with the GIS layer and the public sources. Here below are the different prompts proposed for the benchmarking (Please note that the questions will be asked in French and German to better reflex the final implementation and the real model's performance):

3.3.2. Test Categories and Actual Questions

3.3.2.1. Localization based questions

These prompts are designed to test the chatbot's capacity to understand special environments and respond using geolocated data like the position of the nearest recycling points. These questions evaluate if the chatbot can interpret what type of waste is being referred to and if it is able to link this type of waste to an appropriate localization.

Prompt 1

French	Prompt: Est-ce que je peux recycler des capsules a café? User localization: 46.507° N, 8.305° E
German	Prompt: Kann ich Kaffeekapseln recyceln? User localization: 46.507° N, 8.305° E
Expected result	The cchatbot must indicate in the user's language the nearest recycling point for this type of waste if available in a short text response and then on an actual map. With the test dataset the answer shall be: Ulrichen-Flugplatz (0.4km as the crow flies) for the Coffee Capsule
Utility	This prompt allows us to test if the chatbot is capable of responding to a simple prompt with accuracy

Table 3: Localization based question – Prompt 1

Author's source

Prompt 2

French	Prompt: Bonjour, j'aimerais jeter des piles et des lampes cassées ? User localization: 46.507° N, 8.305° E
German	Prompt: Guten Tag, ich möchte Batterien und kaputte Lampen entsorgen ? User localization: 46.507° N, 8.305° E
Expected result	The Chatbot must indicate in the user's language the nearest recycling points for his 2 different types of waste firstly in a short text response and then on an actual map. With the test dataset the answer shall be: Ulrichen-Furkastrasse (0.26km as the crow flies) for the Batteries Ulrichen-Nufenenstrasse (0.45km as the crow flies) for the Broken light
Utility	This prompt allows us to test if the chatbot is capable of responding to a complex prompt including multiple type of waste and see if the chabot can answer that correctly.

Table 4: Localization based question – Prompt 2

Author's source

Prompt 3

French	Prompt: Comment faire une tarte aux pommes traditionnelle suisse ? User localization: 46.507° N, 8.305° E
German	Prompt: Wie macht man eine traditionelle Schweizer Apfelwähe ? User localization: 46.507° N, 8.305° E
Expected result	No localization point as the question is off subject
Utility	Controle that the chatbot not try to find point for non-waste related questions

Table 5: Localization based question – Prompt 3

Author's source

3.3.2.2. Informational-Based Questions

These questions focus on factual precision. They are designed to test if the chatbot can extract specific information from predefined documents (website and PDFs). Each answer is expected to be based on content from the sources to ensure the quality of the answer and avoid hallucinations of the model.

Prompt 1

French	Prompt: Où dois-je jeter mes vieux journaux et cartons d'emballage en Suisse ?
German	Prompt: Wo soll ich in der Schweiz alte Zeitungen und Verpackungskartons entsorgen?
Expected result	- Paper and cardboard should be collected apart from other waste - Most municipalities have drop-off containers at communal collection points. - About 90% of carboard and paper used in Swiss paper production comes from recycled wastepaper.
Based on	https://www.bafu.admin.ch/bafu/en/home/topics/waste/guide-to-waste-a-z/paper-and-cardboard.html
Utility	Test if the chatbot is able to answer basic RAG questions accurately

Table 6: Informational-Based question – Prompt 1

Author's source

Prompt 2

French	Prompt: J'ai trouvé de vieux CD contenant des fichiers confidentiels et des traces de produits de nettoyage. Comment les jeter ?
German	Prompt: Ich habe alte CDs mit sensiblen Daten und Rückständen von Reinigungsmitteln gefunden. Wie kann ich sie entsorgen?
Expected result	- CDs/DVDs can be recycled at specific communal collection points. - If chemical contamination is present on the CDs, they must be considered as hazardous waste. - For the data protection you must scrape off the metallic top layer to prevent your data to be recovered by other people - If chemicals have leaked on the CDs, consider it as chemically contaminated waste.
Based on	https://www.bafu.admin.ch/bafu/en/home/topics/waste/guide-to-waste-a-z/cds-and-dvds.html https://www.bafu.admin.ch/bafu/en/home/topics/waste/guide-to-waste-a-z/chemicals.html
Utility	Test if the chatbot can understand complex prompt and use multiple RAG sources to answers accurately

Table 7: Informational-Based question – Prompt 2

Author's source

Prompt 3

French	Prompt: Comment faire une tarte aux pommes traditionnelle suisse ?
German	Prompt: Wie macht man eine traditionelle Schweizer Apfelwähe?
Expected result	- Ethier say a default message or say that he has no information about the subject
Based on	No document needed
Utility	Control that the chatbot is not generating answers for off subject questions.

Table 8 : Informational-Based question – Prompt 3

3.3.3. Criteria

To be able to compare and run the benchmarking process for all the different models we have chosen 3 main criteria. Here below is the detailed table of all the criteria alongside their relative weight.

CRITERION	DESCRIPTION	WEIGHT
Speed	Total time used for generating an answer.	2
Cost	The total cost for generating an answer.	2
Accuracy of the answer	Did the chatbot respond correctly to the user	3

Table 9: Benchmarking criteria for each answer

Author's source

3.3.3.1. Speed

For a conversational chatbot, the time to answer one question is really important. A latency under 5 seconds usually feels instantaneous. Between 4-15 seconds the user has to wait a bit, but it is not a particularly big concern for the end user. But in more than 15 seconds the user starts to gain some frustration and may simply abandon the interaction. This threshold is not arbitrary chosen: research as on response time and display rate in human performance with computer shown that a “response times (over 15 seconds) are generally detrimental to productivity, increasing error rates and decreasing satisfaction.” (Shneiderman, 1984)

Our notation is based on these criteria. Models with a response time of up to 15 seconds will give a notation of one out of 5 and models below 15 seconds will have a score ranging from 5 to 2. Here below is the detailed table.

TIME (SECOND)	SCORE
0 – 3.749	5
3.75 – 7.499	4
7.5 – 11.249	3
11.25 – 15	2
> 15	1 (forced)

Table 10: Detailed points attribution for the speed criteria

Author's source

3.3.3.2. Cost

Deploying an AI-powered application in a production environment will expose the app to real-world traffic, and this new heavier usage of the application can lead to a rapid escalation of the application cost. Cost becomes more of a big concern as the application becomes widely adopted.

For an LLM, service costs are tied to the token pricing:

- Input token: number of tokens sent to the model
- Output token: number of tokens generated by the model

The total cost of IA models grows linearly with the combined pricing of input and output tokens. The actual pricing of the token, however, depends on the model type: a lightweight fast model or a complex reasoning model. These two types of models can have drastically different numbers of tokens generated and that can lead to 1 to 4 times more tokens being used.

Our objective is to have a criterion that must have the right balance between penalizing the model with higher pricing without being too harsh. To be able to assess that, first, the weight of this criterion is 2 out of 5, so it is not the highest priority.

The calculation of the score for the pricing is done like that:

```
=IFS(  
  [@[total_cost]] <= PERCENTILE.INC([total_cost]; 0,2); 5;  
  [@[total_cost]] <= PERCENTILE.INC([total_cost]; 0,4); 4;  
  [@[total_cost]] <= PERCENTILE.INC([total_cost]; 0,6); 3;  
  [@[total_cost]] <= PERCENTILE.INC([total_cost]; 0,8); 2;  
  TRUE; 1  
)
```

First, we determine the 5 different percentiles to be able to have roughly the same number of prices per category to have a scale that better represents the data rather than just taking the highest and lowest prices and creating 5 evenly spread categories that may result in one category being overrepresented.

3.3.3.3. Accuracy

The last criterion chosen is the accuracy of an answer. This criterion is a bit heuristic, but we have tried to make it a bit more formal. Accuracy for this benchmark really means how factually correct, well-structured and truly solving the user problem a specific answer is. The complexity of the answer is not the only important point, but the answer should also be organized so that a citizen can follow the logic easily. Here below are 2 examples to better represent the notation point.

- **Lower-accuracy example:**

Questions : « J'ai trouvé de vieux CD contenant des fichiers confidentiels et des traces de produits de nettoyage. Comment les jeter ? »

Answer: « Pour se débarrasser de vieux CD contenant des fichiers confidentiels, il est essentiel de s'assurer que les données ne puissent pas être récupérées. Pour cela, il faut frotter ou gratter la couche métallique sur la face supérieure du disque afin de rendre les fichiers illisibles. Une fois cette étape faite, vous pouvez déposer les CD dans certains points de collecte, chez des revendeurs d'électronique ou lors de journées de collecte organisées par votre commune ou votre canton. Veillez à bien endommager les disques pour garantir la confidentialité de vos données. »

Here the general answer is good for the questions, but seems to lack some guided steps and the overall answer feels a bit chunky. As this answer clearly responds to the question but lack in a structural way we have graded this with a notation of 3.

- **Higher-accuracy example:**

Questions : « J'ai trouvé de vieux CD contenant des fichiers confidentiels et des traces de produits de nettoyage. Comment les jeter ? »

Answer: « Pour éliminer ces CD en toute sécurité :

2. Protection des données

- Posez le disque face imprimée vers le haut et grattez soigneusement la fine couche métallisée (par exemple avec un cutter ou une grosse paire de ciseaux).
- [...]

3. Gestion des traces de produits de nettoyage

- Essuyez les résidus liquides avec du [...]
- [...]

4. [...]

Résumé : retirez ou détruisez la couche de données, neutralisez les traces de produit, puis déposez les disques dans un point de collecte ou chez un commerçant d'électronique. Ainsi, vos informations restent confidentielles et la matière plastique est valorisée. »

The answer above is clearly more structured and helps the citizen to better understand the answer by providing step-by-step explanation and more guidance. As this answer is better structured and has a better overall quality it has been graded with a 5.

As this criterion is the most important point for this project we have decided to give this criterion a slightly higher weight in the global notation. These criteria will be graded 3/5 of the overall notation to represent their major importance.

These 2 answers have been extracted from the benchmarking result that can be found in the Annex III.

3.4. Benchmarks interpretation

This section is dedicated to the detailed analysis of all the results obtained from the benchmarking of all the models explained inside this bachelor thesis.

3.5. Benchmarks Results

The following section will discuss the detailed results we obtained after analyzing all the answer responses from our semi-automated benchmarking process. First, we will explain to you a quick analysis of the important points found during the detailed analysis of the accuracy of each answer for both the RAG and GIS parts. Then we will examine the global result of each criterion explained before and see the most important trend or key points. Lastly, we will present you with the final pondered results of all the models to be able to determine the best model for our application.

Before our testing we would like to include the model Devstral from Mistral AI but we face a problem to tp access his API. Devstral is faster and a bit less performant than Mistral medium, so the result will be quite similar to Mistral medium.

3.5.1.1. GIS

When we have run all the tests for the different models about the GIS question, we arrive at a weird constant. All the models have performed exactly the same and as expected for finding the right collection points. At first, we saw it and we thought that it was a bug because practically all the answers were similar. With more investigation we have discovered and understood the following thing:

The GIS workflow is not really affected by the LLM and a lot of the steps done inside this workflow are not LLM-based. The only 3 parts where LLMs have been used are first for the choice of the right tool at the right time then to determine the type of waste and finally to translate the answer into the user's language.

Of our different tests we have found that the only real part that can be affected by the choice of the model is waste categorization. For instance, some of the lightweight models like gpt-4.1-nano have some lack the understanding and classification of the different questions. Here below is a detailed comparison between 3 different models:

The question that has been asked is the following "J'ai trouvé de vieux CD contenant des fichiers confidentiels et des traces de produits de nettoyage. Comment les jeter ?" And below is the guessed waste type for each model.

MODEL	GUESSED WASTE TYPE	COMMENT
gpt-4.1-nano	17b Paper	The model seems to interpret the documents as a type of waste and not properly understand it inside the context
gpt-4.1	"4 Special waste" and "19 generic wastes"	These two models seem to capture relatively correctly the type of waste as the CD doesn't have a proper type that seem right to select both generic waste and special waste for the cleaning products
o3	"4 Special waste" and "19 generic wastes"	

Table 11: Example of waste classification for different models

Author's source

Here we can clearly see that the important part where LLM matters is the choice of waste type. From more detailed information about where the GIS workflow has been implemented you can see the section 3.2.6

3.5.1.2. RAG

When we finished the benchmark on the RAG prompt, we received results that were close to what we were expecting. Smaller models and non-reasoning models were a bit behind in the accuracy of the answer with smaller, less structured answers that were not as helpful for citizens. But on the other hand, the time to generate the answer and the related pricing were really lower compared to the reasoning model. One of the biggest problems we see inside the answer is the comprehension of the context of the query to answer it with detailed information; sometimes the IA seems to be a bit too generalist. But we think this kind

of issue is not linked to the question posed but more to the different system prompt that may need to be tweaked or reflected in more detail.

3.5.2. Criteria analysis

3.5.2.1. Accuracy

The accuracy of the model, as explained earlier, was evaluated using a scale of 1-5 for each request. The total score represents how well the model tested responds to the questions. The total number of questions for each model was 4 RAG questions, 4 GIS questions and 3 general questions. The maximum possible score is $(4+4+3) * 5 = 11 * 5 = 55$.

Below is the accuracy score for each model tested, sorted from the highest to the lowest:

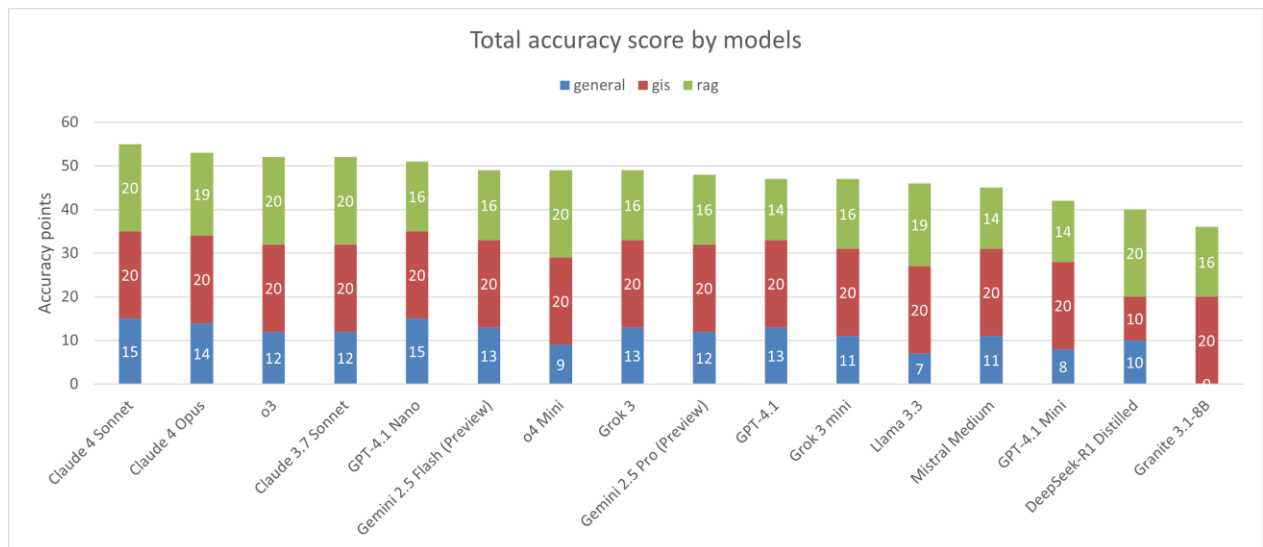


Figure 39: Total Accuracy score by models

Author's source, from the benchmark result in annexes

From these results, we can see that the models from Anthropic generally achieve better scores than those from other companies. From the Anthropic models, Claude 4 Sonnet achieved a perfect score across all categories.

For the best reasoning model, OpenAI's o3 achieved third place with a score of 52. Lastly, the first open-source model available here is Mistral Medium, which scored relatively low compared to the others. Its performance was degraded in the RAG evaluation due to a lack of proper structuring inside the generated answers.

One of the points that we were the most surprised about is with the general questions to test if the system does not respond to completely unrelated waste management questions and we see that some of the models still give an answer, but they are not permitted to do that in their system prompt. Especially for the small model Granite 3.1 8b that didn't get any points for these particular categories. Granite 8b hadn't been explained earlier and it is just used here as a test for a really small model.

3.5.2.2. Speed

The results of the average response time for each model are quite normal and as expected. This score is the combined average of all the response times of the GIS, RAG, and general questions.

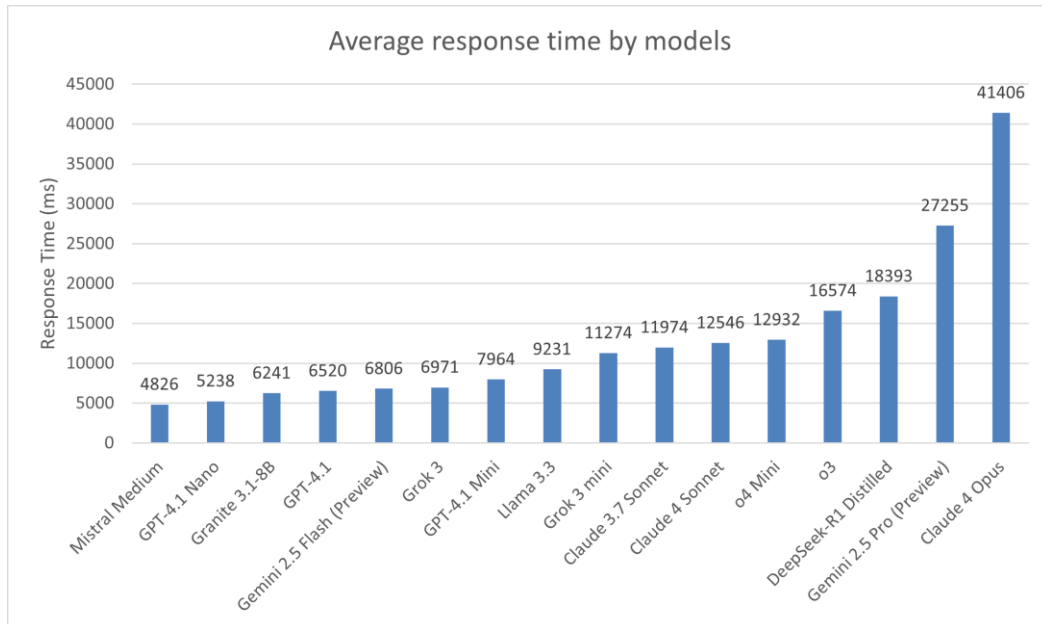


Figure 40: Average response time by models

Author's source, from the benchmark result in annexes

To be able to analyze this result we have pondered the response time to assign a score of 1 to all the response times over 15 seconds and otherwise create 4 equal categories. Here is the average score for each model:

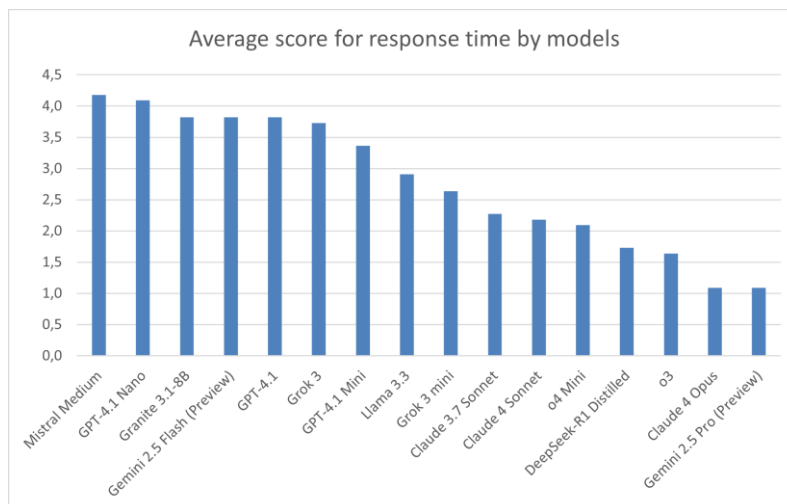


Figure 41: Average score for response time by models

Author's source, from the benchmark result in annexes

Here we can see that the bigger the model is the longer it takes to generate its answer. We can also see that the open-source model like Mistral Medium has a really good score. Lastly the biggest problem of the reflexional model is shown here: models like o4-mini or o3 take more time to generate a complete response.

3.5.2.3. Cost

The cost criterion is one of the most important criteria when it comes to deploying the project to a real environment but it's correct to give pricing the most important weight as it will result in a lower quality response. For this benchmarking the average pricing has been assigned to one of 5 categories that are equally spaced and based on the cheapest and most expensive model. Here one again we haven't set an upper limit for the pricing of the model to privilege the accuracy of the answer over the pricing.

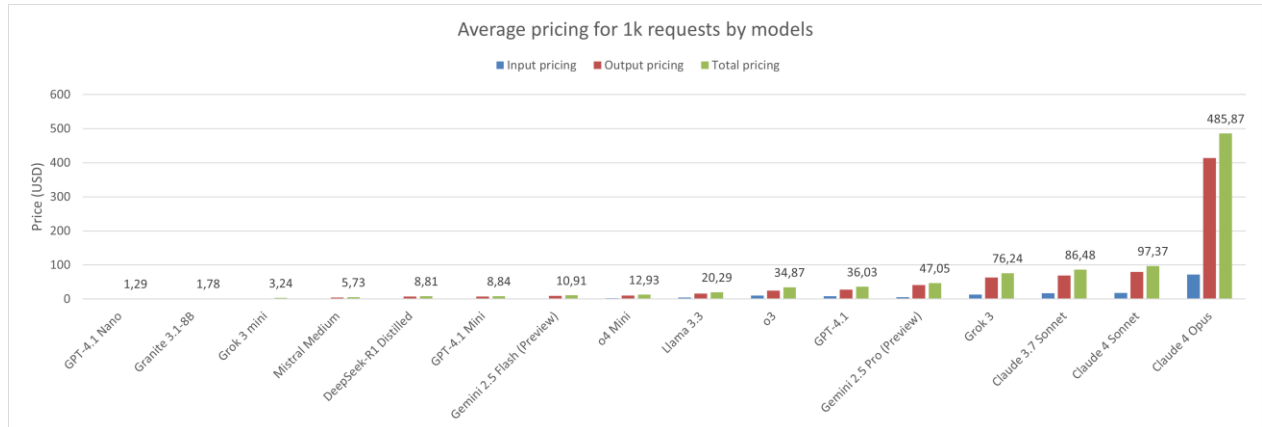


Figure 42: Average pricing for 1k requests by models

Author's source, from the benchmark result in annexes

From what you can see from the following results we have tested a broad range of models with completely different pricing; for example our cheapest model is practically 400 times cheaper than our most expensive model.

From this analysis we can clearly see 3 distinct groups: first there is the cheaper model that ranges from GPT-4.1 Nano to Gemini 2.5 Flash (Preview) then there is the balanced model that ranges from o4 Mini to Gemini 2.5 Pro (Preview) and lastly there is the costliest model that ranges from Grok 3 to Claude 4 Opus.

From this analysis we can clearly see the advantage of smaller models because they cost less for each token generated. Higher models with their built-in chain of thought take more tokens to generate an answer and are pricier.

3.5.3. Global analysis of the results

To be able to choose and compare all the different models to find the most performant one we have decided to ponder our 3 criteria as the following:

- Accuracy counts as 40% of the total notation
- Pricing and Cost each count as 30% of the total notation

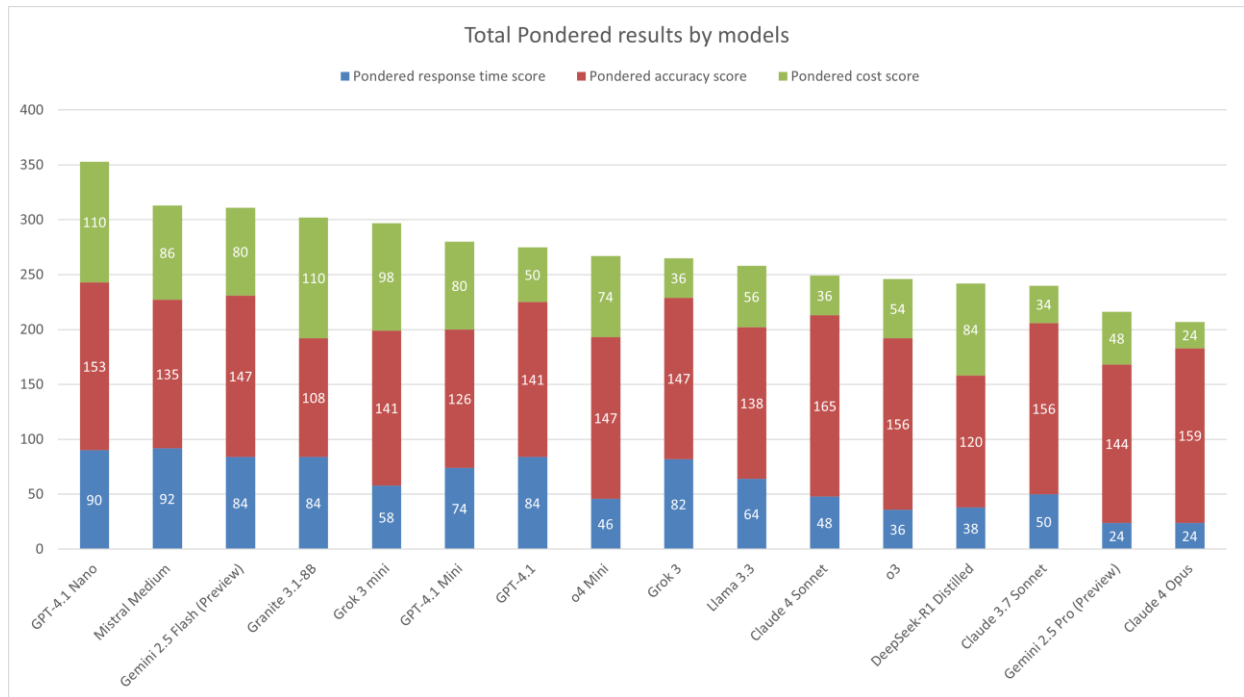


Figure 43: Best model comparison

Author's source, from the benchmark result in annexes

With this analysis we can see that even though reasoning models like o3 and Claude Opus 4 have higher accuracy scores, their lack of response time and higher pricing cost make them not the best models for our project.

This analysis shows that the best models for our prototype are models that are fast, cheap and achieve acceptable accuracy. The best model is GPT-4.1-Nano and the best open-source model is Mistral-Medium.

If we go a little further in our analysis, we can see that for the GIS questions only the pricing and the response time really affect the model. Our only weird case was with Deepseek-R1-Distilled, which had misunderstood if the query was a GIS query and decided to give a default answer. This means that the model has some importance for the accuracy particularly for the detection of the GIS request, type of waste and answer translation. But these points have clearly less importance compared to the pricing and response latency.

For RAG questions a model that has even the smallest chain of thought could really improve the accuracy and structure of the answers.

3.5.4. Final choice of the model

After the analysis of the benchmarking result it is hard for us to say GPT-4.1-nano is the best performing model for everything and should be used for the application.

First, yes, this model has a really good balance between its pricing, accuracy and response time but may categorize the type of waste wrongly as explained inside section 3.5.1.1. The main advantage with this model is the application will work correctly, be reliable and not cost too much money.

However, with this benchmark we have discovered new points that make the choice of only one model for the entire process a bit outdated. For instance, GIS query seems to be less affected by the model accuracy compared to the RAG questions. Moreover, inside the GIS query only some of the steps need to have a more performant model.

Our benchmarking process is not wrong at all because, yes, it allows us to have the most performant model across all the modes available on the application, but it could have been better to test more individual components to be able to determine the best model for each element.

Of the benchmarking above and with all the tests done during the development of the chatbot, it seems to be interesting and useful to have different models for the different tasks of our applications. Our choice for each part is the following:

- For the Detection of the waste type we will privilege correct nonreasoning model like GPT4.1 or Grok 3
- For the language translation the cheapest and fastest model or even a local model so either GPT4.1 nano or Mistral-medium or Devstral
- For the RAG retrieval and answer formation a cheap reasoning model like o4-mini or o3 or a higher and non-reasoning model like Claude 3-7 or 4 sonnet.

In conclusion this benchmark really shows how each model has some key features and key downsides that need to be kept in mind to find the best model for each purpose.

For our current prototype we will stick to the result found with the benchmark, as it will need a lot of adaptation of the code to be able to define a model for each part of the workflow. To allow the user to have at least some choice we have decided to include the top 3 models found on the benchmark that include GPT-4.1-nano, Mistral medium, and Gemini 2.5 Flash (Preview).

We also think that we can partially improve the scoring of certain parts like the general questions or the general structure of the RAG response by creating a more advanced system prompt. However, as this takes a lot of time and does not necessarily lead to a better score, we have decided to only make it a mention here and not test it in practice.

3.6. Prototype Deployment

3.6.1. Containerization and Platform Choice

3.6.1.1. Docker

Docker is an open-source platform that helps to automate the deployment of applications or prototypes by creating portable containers. Each container contains the application code alongside it, with its dependencies and libraries, that ensure a consistent development and production environments.⁸⁶ In our context, Docker help us to:

- **Reproducibility:** The exact same image can run identically on any host that supports Docker.
- **Isolation:** Dependencies like Python packages, Node.js or Tailwind are encapsulated, which avoids conflicts with the host.
- **Portability:** Docker images can be easily pushed to any container registry (e.g. Google Cloud run) and deployed anywhere.

3.6.1.2. Google Cloud Run

Google Cloud Run is a service of Google that allows you to deploy Docker containers on the Google Cloud easily. We have chosen Cloud Run because of these advantages:

- **Generous free tier:** Google Cloud Run offers a generous free tier that includes up to 2 million requests per month and 36 0000 vCPU-seconds of compute time. This makes it a great choice for our prototype because it will avoid charges.⁸⁷
- **Easy to deploy:** With Google Cloud CLI it is possible to deploy a container with only a few commands as it will be explained in the section below. That makes it easier to deploy the prototype again if this is needed.
- **Automatics Scale:** With google cloud run it is possible to scale automatically the resource needed for the application as the number of users grow or reduce. This makes an ideal choice for future applications.⁸⁸
- **Pay-per-use billing:** With Google Cloud Run you can pay only for the exact CPU, memory, and request volume your application used. This greatly reduces the total cost of the system but requires you to wait a bit longer when you open the web application.

⁸⁶ Docker Desktop: The #1 Containerization Tool for Developers | Docker. (2025, April 7). <https://www.docker.com/products/docker-desktop/>

⁸⁷ Cloud Run pricing. (n.d.). Google Cloud. Retrieved July 2, 2025, from <https://cloud.google.com/run/pricing>

⁸⁸ About instance autoscaling in Cloud Run services | Cloud Run Documentation. (n.d.). Google Cloud. Retrieved July 2, 2025, from <https://cloud.google.com/run/docs/about-instance-autoscaling>

3.6.2. Local Configuration Changes

To prepare our prototype to be deployed on Cloud Run, we had to make some adjustments in the Dockerfile and project setup.

3.6.2.1. Flask Port Adaptation

Flask's default development port is 5000. Cloud Run, however, runs by default on the port 8080. To be able to resolve this issue we have decided to change the port on the local side as it is the easiest way to change the port. Inside the Dockerfile and the server.py we have made these changes:

```
# Set environment variables
ENV FLASK_APP=server.py
ENV PORT=8080

# Expose port
EXPOSE 8080

# Run the application
CMD exec gunicorn --bind :$PORT --timeout 600 --workers 2 --threads 8 server:app
```

Code 1: Dockerfile port exposition

Author's source

Inside the Docker file we have exposed the port 8080 and also set the same port to the Gunicorn server.

```
port = int(os.environ.get("PORT", 8080))
app.run(debug=True, host='0.0.0.0', port=port, use_reloader=False)
```

Code 2: server.py port configuration

Author's source

And in the server.py file we have set up the 8080 port for the start of the Flask application.

3.6.2.2. Tailwind Installation via NPM

To be able to properly deploy the CSS with Tailwind we had to install it using NPM instead of relying on the CDN.

First, we had to install Tailwind by using NPM packages. Then, we created 2 files `tailwind.config.js` and `postcss.config.js` to be able to make the necessary configuration of Tailwind. Lastly, we have added 2 new lines inside the `package.json` file to make aliases to the command needed to compile Tailwind. Here below are the 2 commands added; one of them launches the script once and the other one launches it as soon as the project has changed.

```
"build:css": "npx tailwindcss -i ./src/input.css -o ./static/css/tailwind.css",
"build:css:watch": "npx tailwindcss -i ./src/input.css -o ./static/css/tailwind.css --watch",
```

Code 3: alias for the Tailwind CSS compilation

Author's source

3.6.3. Deployment with Google Cloud Run

This section is a step-by-step explanation of the process needed to deploy the prototype on Google Cloud Run. You can find similar explanations in the main readme of the project. Here we will talk a bit more about the parameter configuration for Google Cloud deployment.

3.6.4. Build and push the Docker image

To be able to build and push the Docker image to the Google Cloud infrastructure, you first need to configure the Google Cloud project access with the google cloud init command and then build and push your image.

```
# Authenticate Docker to push to GCR
gcloud init

# Build the container image locally
docker build -t gcr.io/your-project-id/flask-app .

# Push the image to Google Container Registry
docker push gcr.io/your-project-id/flask-app
```

Code 4: Command needed to build and push a docker image to the google cloud

Author's source

3.6.4.1. Deploy to Cloud Run

To actually deploy the container to the Google Cloud Run you need to run the following command:

```
gcloud run deploy flask-app \
  --image=gcr.io/dynamic-gis-rag-agent/flask-app \
  --platform=managed \
  --set-env-vars="$envVarsString" \
  --allow-unauthenticated \
  --region=us-central1 \
  --cpu=2 \
  --memory=2048Mi \
  --concurrency=10 \
  --max-instances=3
```

Code 5: Deploy command Google Cloud Run

Author's source

Here below is a detailed explanation of the parameter use and purpose :

PARAMETER	EXPLANATION
--image=gcr.io/your-project-id/flask-app	Container image to deploy
--platform=managed	Requests the fully managed Cloud Run, this is the version where the server stops when it is not used.
--set-env-vars=...	Comma-separated list of environment variables, for example the AI API key. These must be the same as the variables defined in our local. env to ensure the app has the correct configuration at runtime.
--allow-unauthenticated	Grant public access to the project not restricted with IAM.
--region=us-central1	Region where the service will be hosted. To be able to use the free tier we had to choose us-central1 but a better choice should be closer to the end user
--cpu=2	Allocation of two virtual CPUs per container instance for compute-intensive tasks like embedding task.
--memory=2048Mi	Ram allocation per docker instance.
--concurrency=10	Number of requests that can be made in parallel
--max-instances=3	Be able to have 3 users at the same time it the number of docker run the same time

Tableau 1: Parameters list Google Cloud Run deployment command

Author's source

3.6.5. Limitation of the current version deployed

Besides the containerized version of the prototype, which has been successfully deployed and demonstrated interesting capabilities there remains a prototype version that has some limitations. Here are some explanations of the main limitations.

- **No persistent data:** All the data contained in the application such as the embedded files or website are stored in the local of the container instance. Therefore, if the container instance is being updated or recreated all the data inside the container instance will be lost. This is a really important concern for a production-ready application as the data inside the system cannot be lost at every update.
- **Minimal security and no user management:** There is currently no authentication or authorization layer on the prototype all the users can access all the parts of the application with no restriction. The only real security and confidentiality measure done for this project has been the hiding of all the API keys and the strict confidentiality usage of the GeoJSON data.
- **Lack of Performance Optimization:** The current prototype implementation has not been optimized at all, and some parts of the interface or some processes inside the code are not optimal. For example, the application heavily does small API calls for things instead of one big call that is then caught through the life of the page.
- **Monolithic deployment:** The deployment of the project has been done with no real care for the separation of the concern. The front end, backend, and DB are all inside the same container. This is not the way to do this, but the goal of the project is to determine whether it is possible to use GIS data with an LLM. Moreover, this version of the application is still a prototype.

3.7. Integration to citiwatts

One of the parts of the bachelor thesis was to see who and if it is possible to integrate the developed chatbot inside the citiwatts platform. The following section will be dedicated to presenting to you how it will be possible to implement it into the platform

3.7.1. Citiwatts

“citiwatts is an open-source online platform developed as part of the project OpenGIS4ET. It builds on the H2020 Hotmaps platform, which created a foundational toolbox for heating and cooling planning across Europe.”⁸⁹

This platform offers building blocks to be able to display geographical information like heating and colling demand and access to energy potential for any European region.

For this bachelor’s thesis this project will serve as a base for potential future integration on this platform

3.7.2. Mockup

3.7.2.1. User-Interface

The user interface that needs to be integrated into the platform is quite minimal as the localization feature is already available on the project and the only real interaction that the citizen needs is a place to ask their questions and a simple way to see the answer generated. To be able to visualize more effectively how it could resemble the citiwatts interface, we have created an example mockup.

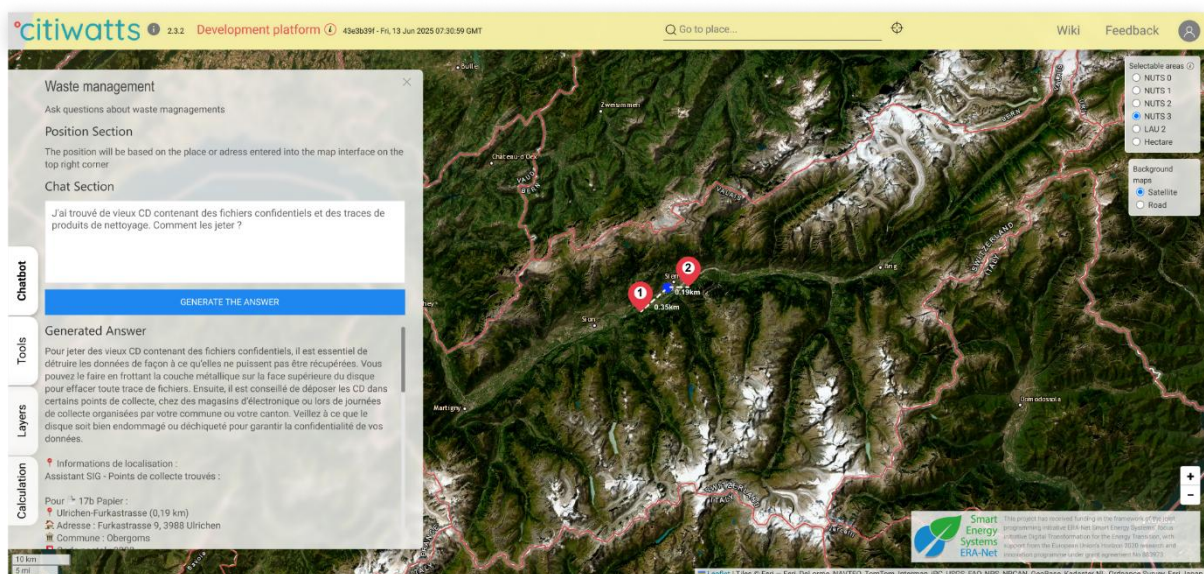


Figure 44: Mockup of the chatbot on citiwatts

Author's source, background captured on citiwattsdev <https://citiwattsdev.hevs.ch/map>

The integration could be quite simple to add if it is considered as a main, feature; we could add a new tabulation for the actual chatbot. On this new tabulation we will be able to add an input box where the user would be able to ask a question to the chatbot. The chatbot will use the prompt asked by the citizen and get the actual position of the user on the map with the go the place search bar.

Once the question is generated, we will first display the textual part on the left side and bring the interactive component directly onto the map to be able to localize the nearby collection point that accepts the waste type described in the user prompt. As well as the current chat interface, the pin on the map will be able to be clickable to display additional information about the collection point.

⁸⁹ Start. (n.d.). Citiwatts. Retrieved July 9, 2025, from <https://project.citiwatts.eu>

3.7.2.2. Administration panel

The main part of the integration process will not be the user interface but the creation of the administration pages. This part will not be required at first as we can use the interfaces on the developed interface to be able to manage the chatbot but in a more final version a dedicated interface on the citiwatts platform could be developed.

To be able to see what the requirement of this interface will be we have created a global mockup of one of the pages that will need to be developed:

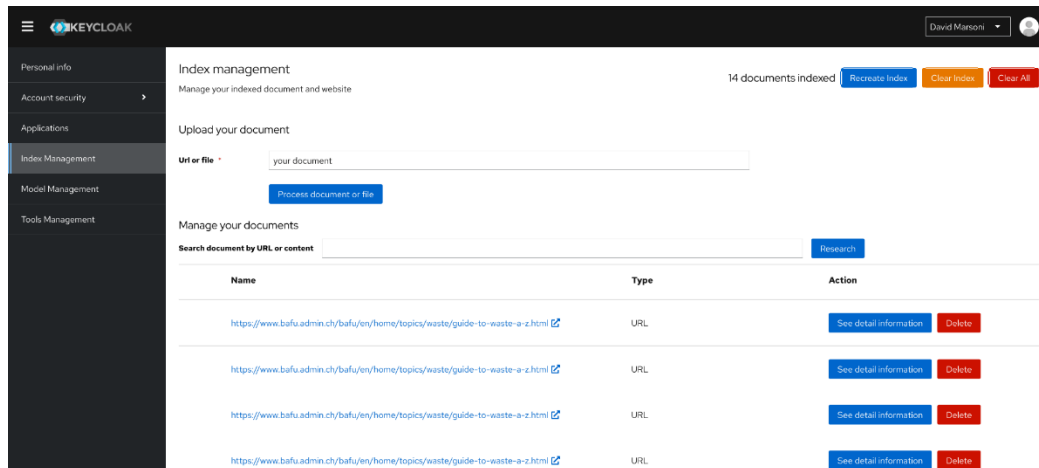


Figure 45: Mockup Admin panel citiwatts

Author's source, based on captured citiwattsdev account admin pages

This administration interface will need to have at least 3 main pages here above we have decided to make the mockup of one of them as the other one could just be created based on the existing pages of the prototype.

The 3 main pages will be one page to upload and manage the files, one to be able to manage the different models and embeddings and one dedicated to the management of tools like filtering the document available or not for the user on the application.

3.7.3. Specific adjustment and key point of interest

The main adjustment that would have to be done on the project to be able to use it on the citiwatts platform is the security part. For the moment all API endpoints are accessible by everyone and there is no check of any security at all. This for a deployed application with real users will be a treat for sure because anyone with enough knowledge could literally delete all the indexes or launch thousands of queries at the same time.

To avoid this, most of the API endpoints will need to be accessible to specific users and the number of requests that a user can make on the application, for example in one minute should be configured to avoid a user running a lot of queries.

Another point that could cause problems is the lack of management features on the administration panel such as activating or deactivating a tool for the whole application with persistence or being able to choose the documents that can be used for the RAG interface for the whole application.

Other points, like being able to add your custom documents, will need to change in deep actual management of the files, and this will require time to implement.

4. Conclusion

This bachelor's thesis evaluates whether a modern large language model can be combined with dynamic GIS data to deliver accurate and interactive waste-management assistance for the citizens. This thesis confirms that LLMs can efficiently understand and manage dynamic GIS data. This simplifies how citizens can access waste management information.

Benchmark results demonstrated are key findings. First, light way models such as GPT-4.1 nano and Mistral Medium achieve remarkable scores on our benchmark for being quick and inexpensive to generate answers. Secondly, mid-tier reasoning models particularly o3 and Gemini Pro 2.5, performed better when handling complex retrieval tasks that require a good context understanding.

A complete prototype was successfully developed that integrated a graphical chatbot interface and a complete administrative panel built using Flask, Python, Tailwind, and JavaScript. The backend of the prototype combined LlamaIndex tools and ChromaDB to deliver precise GIS answers and comprehensive RAG responses. This prototype was then packaged with Docker and deployed on Google Cloud infrastructure. This final deployment proves the feasibility of the proposed architecture.

The building of the different tools addresses concerns regarding the confidential GIS data. It was found that managing sensitive data does not pose significant issues if proper tooling with separation of sensitive data processes in dedicated functions is used. This ensures that the IA model never has direct access to the entire confidential geopoint list.

Overall, this thesis established that the optimal solution for the model choice is not a single model but rather a combination of models with a powerful reasoning model for the RAG model, and a fast response model for the GIS queries. This will result in a system that will be more performant and achieve better accuracy in complex queries without increasing the cost of each query.

In summary this thesis confirms that LLMs are able to understand and manage dynamic GIS data in order to simplify access to waste management information. By combining a targeted model selection, a flexible and robust RAG system, and an interactive map for the GIS data, we created a good basis for next-generation services for citizens about waste management.

4.1. Next steps

Building on the prototype presented here, several improvements with future work can help to transform this prototype into a robust, production-read service, and even extend its capabilities.

4.1.1. Integration Into citiwatts

The next immediate step will be to embed our GIS-enhanced chatbot into the existing citiwatts platform. Rather than a standalone prototype it could become a widget or a new tabulation within the citiwatts user interface. This will require modifying the citiwatts platform and the prototype APIs to be able to add an authentication system so administrators could manage the application while citizens would only have access to the chatbot integration.

4.1.2. Keep it up to date

As models are currently evolving rapidly new functionalities and breakthroughs are presented this bachelor thesis can rapidly become outdated. For instance, only during the month of June and July the model Grok 4 and GPT 5 have been announced. Grok 4 is not already available for general public. New open-sources models from EPFL has also been announced and will be available during the summer.⁹⁰

⁹⁰ Anchisi, M., & Meyer, F. (2025). *A language model built for the public good*. <https://actu.epfl.ch/news/a-language-model-built-for-the-public-good/>

5. Project Management

To organize this work effectively for this bachelor's project, several key elements have been implemented. The scrum methodology was chosen to manage the project and provide an iterative and agile improvement on the project. Despite this project being an individual project, the scrum practices helped me to have structured and continuous project tracking.

The project began with a preliminary sprint (Sprint 0) dedicated to understanding the project preparing the environment, analyzing the different technologies and filling the state of the art of the report. The following week, they are then organizing into sprints each one with a main goal. At the end of the project, a dedicated wrap-up sprint was used to be able to finalize the documentation and correct the remaining bug on the prototype.

Throughout the project, a logbook was maintained to record daily progress. The resume of this logbook can be found on Annex II. The full backlog is available on annex I.

5.1. Branching strategies

For this project I use a simple branching strategy that contains only 2 branches:

- **Main:** this branch contains a stable and complete version of the prototype.
- **Dev:** This is where the actual work of the prototype is done adding new features or correcting bugs.

This was chosen like that because I'm alone and the project is mainly a prototype. A more complex strategy is therefore not needed for this project. This method really helps me to focus more on the actual code and deliver useful features rather than micromanaging a tonne of branches that would have been a waste of time. Otherwise, if this project is developed with multiple people, it will be important to change this branching strategy.

5.2. Sprint 0

Sprint 0 is a sprint of preparation that has been done for a full month, starting from 12 May 2025 to 3 June 2025. The goal of this pre-sprint phase was to gain a global understanding of the bachelor thesis mandate before doing any code. To make my life easier I have decided to split this massive pre-sprint into 4 1-week sprints where my work changed between organizational set-up, research, and PoC, to be able to have a good base for the delivery sprints that will follow.

The 4 sub-sprints have been focused on various tasks. Here below is a recap of those sprints:

SUB-SPRINT	WEEK	MAIN TASKS
0.1	Mon 12 → Sun 18 May	Kick-off, backlog skeleton, first research
0.2	Mon 19 → Sun 25 May	Deep dive into models, first PoCs
0.3	Mon 26 May → Sun 1 Jun	Documentation polishing, embedding model testing
0.4	Mon 2 → Sun 8 Jun	Quick QGIS test, State of the Art wrap-up

5.2.1. Sprint 0.1 (May 9 – 15, 2025)

The Sprint 0.1 Started with the formal kick of meeting with David Wannier to present the overall bachelor thesis alongside the main goals and tasks of the project. During this meeting we discussed the access I have needed for the project and if there was confidential data.

The next task I did was the first backlog of the project that gave me a global vision of the task time and stuff I will need to do for this bachelor thesis. In parallel with this task, I have immersed myself in the subject of waste management. Next at the end of the week I have made 2 sections that detail the current website available to find information and geographical information about waste management. The two websites I have analyzed are [swissrecycle.ch](https://www.swissrecycle.ch) and [recycling-map.ch](https://www.recycling-map.ch).

5.2.2. Sprint 0.2 (May 16 – 23, 2025)

The second was mainly focused on 2 main tasks. On one side I have mainly focused on the state-of-the-art part, looking at academic papers on AI-driven waste management, the LangChain and LlamaIndex frameworks, embedding models and early proof of concepts. On the other hand, I have test the Devstrall model API and looked at the recycling map website mobile app. On this part I have also stated the creation of an early chatbot prototype.

One of the background tasks of this week was the gradual filling of the model's section of the state of the art with all the actual models available.

5.2.3. Sprint 0.3 (May 24 – 30, 2025)

During sprint 0.3 I have continued my background task of filling in the detailed information about the different models. Alongside that, on the state-of-the-art part I have done research on embedding models to determine which one is the best one for our case and fill out a section dedicated to the ChromaDB vector database.

The most practical stuff I have done during this sprint was testing different embedding models created by small-scale PoC tests where I compare roughly some embedding models and also tested some RAG queries with them.

5.2.4. Sprint 0.4 (May 31 – June 3, 2025)

For the last sprint the sprint 0.4 of the preparation part I have polished and cleaned a lot of part of the state of the art. I have filled a lot of sections related to the general concept of AI, LLM or their relative history. I have also explained some concepts like RAG or the token principle.

On the more technical part I have continued my PoC with tests of some embedding and various other small-scale tests. Lastly at the end of the week I have done some tests with QGIS and one of the extensions they have called IntelliGEO.

5.3. Sprint 1

Sprint 1 lasted from Wednesday, June 4, 2025, to Wednesday, June 11, 2025. It was the first sprint of the project. During the sprint planning, five user stories were selected, for 19 story points. This decision was made based on the overall project estimate of 87 story points, which includes 10 story points for additional features. As the project is planned, four sprints is a good estimate to start with. The average of 77 story points over 4 sprints is 19.25 SP.

Order	Title	State	Effort	Assigned To	Rem...
1	> As a waste management department, I want to use standardized map data (GeoJSON)	Done	1	Marsoni David	
2	> As a waste management department, I want to use interactive maps and test their functionalities	Done	1	Marsoni David	
3	> As a waste management department, I want to list which AI tools can handle maps and questions	Done	1	Marsoni David	
4	> As the waste management department, I want to create a test prototype to enrich the chatbot with public data, in order to provide citizens with more detailed and useful information.	Done	13	Marsoni David	
5	> As the waste management department, I want to analyze how well the chatbot uses the public data	New	3	Marsoni David	6

Figure 46: Sprint 1 User Stories from Azure DevOps
 Autor's source

The main goal of the first sprint was to create and implement the RAG prototype and select the best-performing model and do some preparation for the map understanding with GeoJSON and Leaflet.

During the sprint the first 2 user stories #1 and #2 have been completed and allowed to have a better vision of the GeoJSON structure and how to display this information on a map. Then the majority of the time was spent on the development of the RAG prototype.

The creation development took more time than planned at the start due to the complexity of creating a complete application with a lot of detail like the choice of the model or embedding model being able to process the website and store the indexes. This took a long time to understand and create.

This sprint was also a bit particular because alongside this sprint we had an oral examination where we had to prepare a presentation which led to having less time to do this sprint. Also, during the same period, I had some problems with my school computer and had to reinstall it completely, which also took some precious time.

User story #3 was completed and allows us to have a fulfilled list of all the models currently available. This allows us to know which model will be integrated into the prototype.

Then the user story #4 was complete and allows users to have a chatbot that can ingest PDFs and website to be able to query these documents after that with the chatbot. The main interface was also developed during these user stories.

As our goal of the first phase of the project is to perform a benchmark of multiple models available, the creation of the RAG prototype has included support of multiple models.

The user story #6 was, however, not completed at all. This was due to a lack of time to complete this task and a bad estimate of the overall time to work on the sprint.

Lastly this end of the first sprint was a bit particular as I have made my own review and retrospective, and due to the 1 week sprint and the problem encountered with my PC I had completely forgotten to ask for a meeting with Wannier. This is clearly a point that I have changed for the next sprints.

5.4. Sprint 2

Sprint 2 lasted from Thursday, 12 June 2025 to Thursday, 19 June 2025. A During the previous sprint I was not able to finish the number of story points, but that was mostly due to my analysis of a lack of time to work on the project on my side. As with the previous sprint, the total story points planned was also 19 SP.

The main goal of this sprint was to integrate the GIS part of the project alongside determining the last question and criteria to finally perform the benchmark of the project. In detail his second sprint was composed of the following user stories:

	Order	Title	State	Effort	Assigned To	Rem...
1	>	As the waste management department, I want to set up a local testing environment	Done	8	Marsoni David	
2	>	As the waste management department, I want to define what characteristics of a model makes a good chatbot	Done	3	Marsoni David	
3		As the waste management department, I want to select the best-performing model	New	3	Marsoni David	
4		As the waste management department, I want to analyze how well the chatbot uses the public data	New	3	Marsoni David	
5		As the waste management department, I want to test several AI models in a local environment	New	2	Marsoni David	

Figure 47: Sprint 2 User Stories from Azure DevOps

Author's source

During this sprint I discovered that the original 8 SP for us #1 was once again clearly underestimated even though I was first thinking it would not be too complicated as the base of the prototype had already been developed but the total time passed on this task was way higher than expected. At the end of this sprint, I managed to finish this user story.

The other user story, #2, I have achieved was to resolve to define the characteristics of what makes a good IA model for our purposes, and this task went as planned.

User stories #3, # 4, and #5 have not been started at all for this sprint due to the wrong estimation of the first user stories.

During my first proper sprint review with Wannier I presented the user stories done for this sprint and also showed him the global prototype. as it was the first time, I showed it to him.

For the sprit retrospective, we discussed the problem encountered during the sprint and the solution found.

5.5. Sprint 3

Sprint 3 lasted from Monday 23 June to Sunday 13 June 2025. This sprint started with the sprint planning with Pannier where we discussed the following things. For the backlog we see that some user stories about the ability to add Word and Excel documents were missing and we also discuss planning a user story to deploy the prototype, so it is not anymore on my computer.

We also have discussed about the point on the bachelor thesis to integrate the chatbot into the citiwatts platform and we agree that the most important thing was first to finish the benchmarking and then deploy the project.

Then we agreed to set up a velocity of 16 for this sprint and chose the following user stories:

	Order	Title	State	Effort	Assigned To	Rem...
1	>	As the waste management department, I want to be able to upload Excel documents	Done	2	Marsoni David	
2	>	As the waste management department, I want to be able to upload Word documents	Done	2	Marsoni David	
3		As the waste management department, I want to select the best-performing model	New	3	Marsoni David	
4		As the waste management department, I want to analyze how well the chatbot uses the public data	New	3	Marsoni David	
5	>	As the waste management department, I want to test several AI models in a local environment	New	2	Marsoni David	

Figure 48: Sprint 3 User Stories from Azure DevOps

Author's source

This print started well with the integration of both Word and Excel document ingestion for the chatbot RAG tool. These 2 integrations have completed the user stories #1 and #2.

After that I move to the use stories #3, #4, and #5 where I had to run and analyze the results of all the models chosen to be able to determine the most suitable model for our project. When I started developing the necessary components to get my metrics like the time to generate the answer or be able to calculate the exact price of a query.

To implement the time to generate the full answer by the different model was quite easy and did not cause too many problems. However when I started the implementation of the total number of token users and the resulting pricing for each request I spent hours and hours to changing a big part of the code because I saw that to be able to track all the tokens for every part of the application, I would need to add new functions and functionalities in many places inside the code. At the end of the sprint, I was not at the end of these implementations so user stories #3, #4, and #5 have been started but not finished.

During the sprint review with Wannier we discussed the newly supported file format and after that we directly switched to the sprint retrospective. During this retrospective I have seen that I may not have taken time to implement correctly the tracking of tokens, and I found a new way to do it that I needed to test.

5.6. Sprint 4

Sprint 4 lasted from 30 June 2025 to Sunday 6 July 2025.

During the sprint planning we saw what was the most important task that needed to be done in order to be able to have a viable prototype at the end of the project and we decided to choose the following user stories:

Order	Title	State	Effort	Assigned To	Rem...
1	> As the waste management department, I would like to be able to deploy the prototype	Done	8	Marsoni David	
2	> As the waste management department, I want to analyze how well the chatbot uses the public data	Done	1	Marsoni David	
3	> As the waste management department, I want to test several AI models in a local environment	Done	1	Marsoni David	
4	> As the waste management department, I want to select the best-performing model	Done	1	Marsoni David	

Figure 49: Sprint 4 User Stories from Azure DevOps

Author's source

The estimated velocity was 10 but in order to finish both the prototype and deploy the project I have decided to work a bit harder and allocate a bit more time to this sprint to be able to finish everything. This sprint has been planned at the end with 11 SP. This was possible because I had the need to re-evaluate the remaining work on the 3 user stories related to the benchmarking.

The first user story, 1, was the deployment of the project. This US was terminated in time and was even a bit shorter than planned.

The remaining user stories #3, #4, and #5 were completed by the completion of the benchmarking page and the analysis of the result for each model. These user stories take a bit more time than expected because I hadn't planned to do so much work on the Excel file to have comprehensive and usable results.

During the retrospective with Wannier I presented the benchmarking result alongside the version deployed of the project, that was already sent on the end of Tuesday.

By the end of this sprint, I achieved completing all the user stories planned and I believe my final velocity is around 11 or 12 SP.

5.7. End of the project

After the last retrospective there were some minor adjustments that needed to be made in order to have a proper interface for the client and an administration panel for the management of the prototype. So, during the first and second days of the last week I spent time on the project adding these missing features and deployed the solution directly.

Another minor part was to create some mockup of how to integrate this chatbot into the citiwatts, platform, which took me half a day with the documentation part. One mockup was done before during a sprint because I was wondering how it may look on citiwatts.

The rest of the week was dedicated to finishing the documentation filling all the missing or remaining sections of the documentation like adding a new section for a part of the prototype or correcting the order of some section. Then during the 2 last days before the end of the bachelor thesis I wrote up the conclusion and then did the poster of the bachelor thesis.

6. Bibliography

- Binjos, A. (2023). *Integration of Artificial intelligence for the exploration of energy transition solutions*. HEG Sierre IT magnagement. Swiss Open Access Repository. <https://sonar.ch/hesso/documents/326783>
- DeepSeek-AI, Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., & Xu, R. (2025, January 22). *DeepSeek-R1: Incentivizing reasoning capability in LLMs via Reinforcement Learning*. <https://doi.org/10.48550/arXiv.2501.12948>
- Hadi, M., Tashi, Q., Qureshi, R., Shah, A., Muneer, A., Irfan, M., Shaikh, M., Akhtar, N., Wu, J., & Mirjalili, S. (n.d.). *Large Language Models: A Comprehensive Survey of its Applications, Challenges, Limitations, and Future Prospects*. <https://doi.org/10.36227/techrxiv.23589741.v3>
- Hou, X., Zhao, Y., Wang, S., & Wang, H. (2025, April 06). *Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions*. <https://doi.org/10.48550/arXiv.2503.23278>
- Kaplan, J., McCandlish, S., Henighan, T., Brown, T., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. (2020, January 23). *Scaling laws for neural language models*. <https://doi.org/10.48550/arXiv.2001.08361>
- Kulkarineetham, S., Supanich, W., Prasatkaew, W., & Mareo, P. (2025). Leveraging AI Chatbots to Foster Engaged Community Waste Management: A Case Study of Takhian Tia Community. *2025 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON)*, (pp. 499-503). <https://doi.org/10.1109/ECTIDAMTNCN64748.2025.10961958>
- Mohaimenul Azam Khan, R., Md. Saddam Hossain, M., Kaniz, F., Nur Mohammad, F., Sadman, S., & Most Marufatul, J. (2024, February 13). A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges. *IEEE Access*, 26839 to 26874. <https://doi.org/10.1109/ACCESS.2024.3365742>
- Muennighoff, N., Tazi, N., Magne, L., & Reimers, N. (2023, March 19). *MTEB: Massive Text Embedding Benchmark*. <https://doi.org/10.48550/arXiv.2210.07316>
- Olawade, D., Fapohunda, O., Wada, O., Usman, S., Ige, A., Ajisafe, O., & Oladapo, B. (2024, June 01). *Smart waste management: A paradigm shift enabled by artificial intelligence*. <https://doi.org/10.1016/j.wmb.2024.05.001>
- Shneiderman, B. (1984, September 02). Response time and display rate in human performance with computers. *ACM Computing Surveys*, 16(3), 265-285. <https://doi.org/10.1145/2514.2517>
- Unlu, E. (2023, August 01). *Structural Embeddings of Tools for Large Language Models*. (arXiv, Editor) <https://doi.org/10.48550/arXiv.2308.00447>
- Wang, L., Yang, N., Hang, X., Yang, L., Majumder, R., & Wei, F. (2024, February 08). *Multilingual E5 Text Embeddings: A Technical Report*. <https://doi.org/10.48550/arXiv.2402.05672>
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Chi, F., Le, Q., & Zhou, D. (2023, January 10). *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. <https://doi.org/10.48550/arXiv.2201.11903>
- Xuezhi, W., & Zhou, D. (2024, May 23). *Chain-of-Thought Reasoning Without Prompting*. <https://doi.org/10.48550/arXiv.2402.10200>
- Yu, D., Bao, R., Mai, G., & Zhao, L. (2025, February 4). *Spatial-RAG: Spatial Retrieval Augmented generation for Real-World Spatial Reasoning questions*. <https://doi.org/10.48550/arXiv.2502.18470>

Annexes

Annex I: Complete product backlog

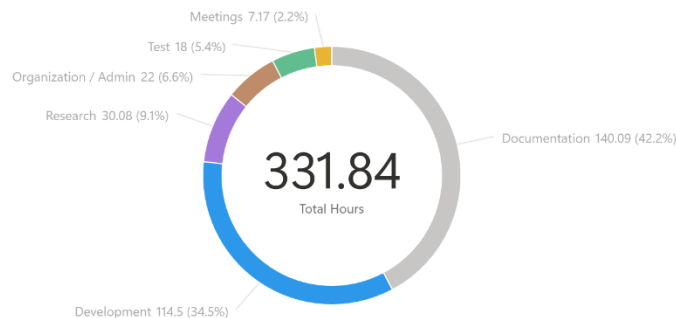
ID	Work Item Type	Title	Assigned To	State	Effort
1	Product Backlog	As a waste management department, I want to use standardized map data (GeoJSON)	Marsoni David	Done	1
2	Product Backlog	As a waste management department, I want to use interactive maps and test their functionalities	Marsoni David	Done	1
3	Product Backlog	As a waste management department, I want to list which AI tools can handle maps and questions	Marsoni David	Done	1
4	Product Backlog	As the waste management department, I want to create a test prototype to enrich the chatbot with public data, in order to provide citizens with more detailed and useful information.	Marsoni David	Done	13
5	Product Backlog	As the waste management department, I want to analyze how well the chatbot uses the public data	Marsoni David	Done	1
6	Product Backlog	As the waste management department, I want to set up a local testing environment	Marsoni David	Done	8
7	Product Backlog	As the waste management department, I want to define what characteristics of a model makes a good chatbot	Marsoni David	Done	3
8	Product Backlog	As the waste management department, I want to test several AI models in a local environment	Marsoni David	Done	1
9	Product Backlog	As the waste management department, I want to select the best-performing model	Marsoni David	Done	1
10	Product Backlog	As the waste management department, I want to create a module that lets the citizens search information's about waste management and have a dynamic map in response	Marsoni David	New	13
11	Product Backlog	As the waste management department, I want to create a module that lets the citizens search information's about waste management and have detailed information about his request such as information about the waste	Marsoni David	New	8
12	Product Backlog	As a digital services team, I want to include the map chatbot inside the Citiwallis platform	Marsoni David	New	8
13	Product Backlog	As a citizen, I want to ask to a chatbot where I can bring my waste	Marsoni David	New	5
14	Product Backlog	As a citizen, I want to ask questions about waste management	Marsoni David	New	3
15	Product Backlog	As a citizen, I want to have the itinerary to the nearest collection point to find this center more easily.	Marsoni David	New	5
16	Product Backlog	As a tester, I want to conduct a test session with customer to test the finish chatbot	Marsoni David	New	5
34	Product Backlog	As the waste management department, I want to be able to upload Excel documents	Marsoni David	Done	2
35	Product Backlog	As the waste management department, I want to be able to upload Word documents	Marsoni David	Done	2
43	Product Backlog	As the waste management department, I would like to be able to deploy the prototype	Marsoni David	Done	8

Annex II: Time tracking

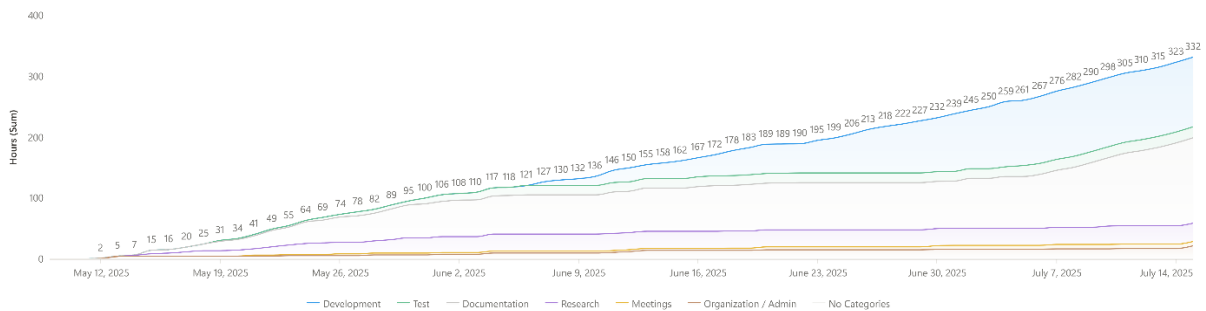
This section will contain a simplified presented way of the lookbook that has been recorded daily during the entire bachelor thesis. I have decided not to include the entire notebook detail as I think it does not add any value to the project.

To keep my current work each day I have used notion which allows you to create graph automatically based on the data written on a table. This allowed me to have a noted version that is not formatted of my current work that then I have categorized into 6 majors sections: Meeting Test, Organization, Research, Development and Documentation. These 6 categories allowed me to have a global vision of my work.

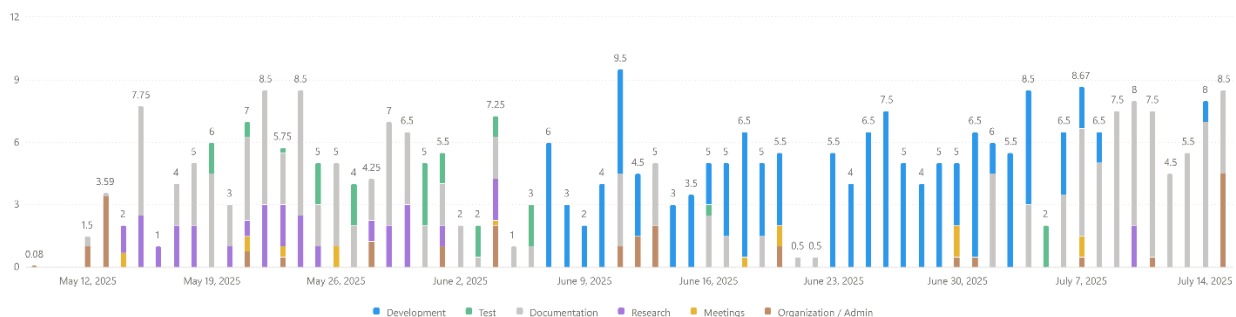
The first graph is the total number of hours passed on my bachelor thesis which is 331.84 hours.



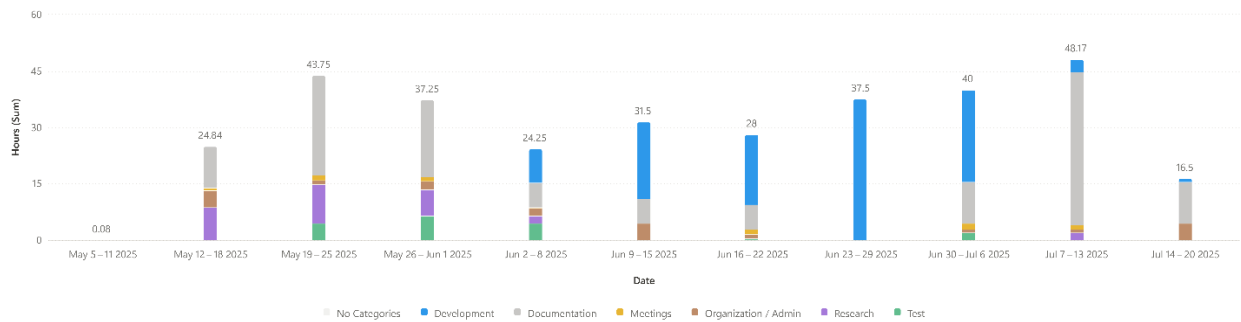
This next graph shows a cumulative version of my work hours. What is interesting with this graph is we can see when I have passed more time on a particular task and see if I worked consistently along the entire project. From the graph below we can see that I started this bachelor thesis a bit slower and started really to find a good consistency around week 4.



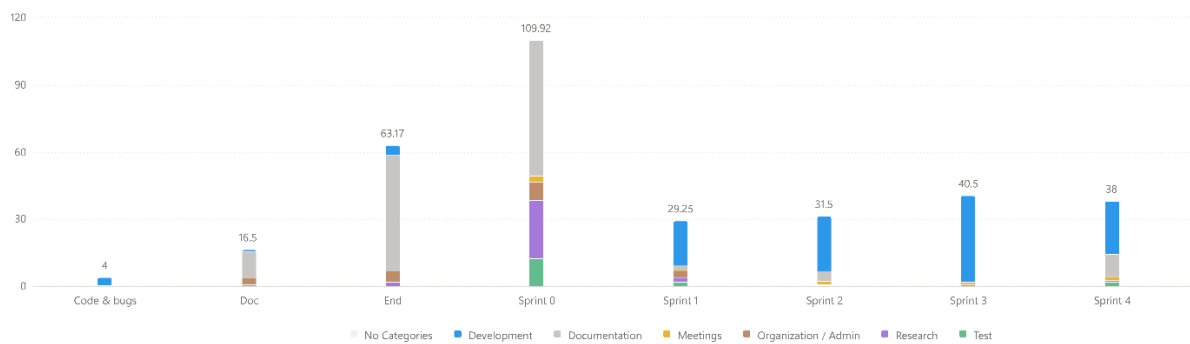
The Next graph is my detailed work each day during all the bachelor theses. From this graph we can see quite easily when the different sprints start to end or when it is the weekend. This graph is important to have all the necessary details of my logbook.



The last 2 graphs I would like to present to you to have full vision of my work is first the number of hours spent each week on the bachelor thesis. This clearly shows some bad weeks I had like the week between June 2 and June 8 when I had my PC problems.



The last graph is the vision of my work for each sprint including the first sprint and the last sprint of the project. The two categories' Docs and Code & bug are here because I had some time during a sprint to do some other work on the project that were not related to any sprint.



Annex III: Excel file of the detailed benchmarking result

In this annex, will be display a partial part of the excel file “benchmark_results_2025-07-04T13-05-39.xlsx” used to analyze the result of the will be displayed.

Result pages



Detail page

1	timestamp	model_id	model_name	provider	full_model_name	category	question_id	question	question_description
2	2025-07-04T12:01	gpt-4.1	GPT-4.1	openai	openai gpt-4.1	gis	gis_1	Est-ce que je peux recycler c Coffee capsule recycling query in French	
3	2025-07-04T12:01	gpt-4.1	GPT-4.1	openai	openai gpt-4.1	gis	gis_2	Kann ich Kaffee kapseln in req Coffee capsule recycling query in German	

response	success	Accuracy	response_time_ms	response_time_category	input_tokens	output_tokens	total_tokens
Assistant SIG - Points de collecte trouvés : Pour ☕ 20 capsules de café : 📍 Ulrichen-Flugplatz (à 0,35 km) 📍 Adresse : Seestrasse 92, 3988 Ulrichen 🏘 Commune : Obergoms 📮 Code postal : 3988 **GIS-Assistent - Sammelstelle gefunden.**	TRUE	5	8533	3	565	175	740
Für ☕ 20 Kaffee kapseln: 📍 **Ulrichen-Flugplatz** (0,35 km) 📍 **Adresse:** Seestrasse 92, 3988 Ulrichen 🏘 **Gemeinde:** Obergoms 📮 **Postleitzahl:** 3988	TRUE	5	8777	3	565	189	754

AB		AC		AD	
enabled_tools		raw_pricing_data		raw_token_usage	
"gis_tool":true,"rag_tool":false,"requ("base_input_cost_per_1k":0,"base_output("base_input_cost_per_1k":0,"base					
"gis_tool":true,"rag_tool":false,"requ("base_input_cost_per_1k":0,"base_output("base_input_cost_per_1k":0,"base					

7. Author Declaration

I hereby declare that I have carried out this bachelor's thesis independently, without any assistance other than those explicitly mentioned in the references, and that I have used only the sources expressly listed. I have used artificial intelligence (AI) tools such as ChatGPT, Claude, and Gemini to help reformulate sentences correct grammar and to better understand certain concepts relevant to this work. I also used GitHub Copilot during the development of the prototype and related module, as a coding assistant. These tools were used strictly for support purposes, and all content, ideas, and technical decisions are entirely my own work.

I will not provide a copy of this report to any third party without the joint authorization of the supervising professor and the professor responsible for the bachelor's thesis, except for the individuals who provided essential information for the completion of this work and whom I list below:

- David Wannier
- Abdullah Binjos

Date and location

Ayent, le 16.07.2025

Signature

