ARE-U-QUERYOUS?

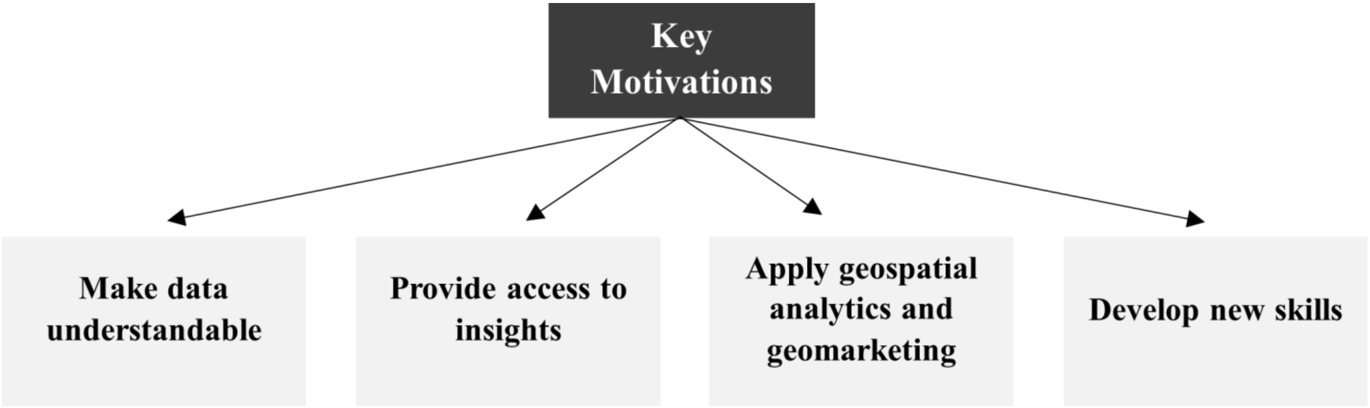
A web app to make open geospatial data accessible for all!

Nicolás Ezequiel D'Alessandro Calderon
Bachelor's degree in Techniques for Software Application Development
Universitat Oberta de Catalunya, 2025

1. INTRODUCTION: MOTIVATION AND GOALS

Problem: Open data is becoming more available, but it's **still hard to explore** or understand for non-expert users.

Proposed Solution: To **simplify access** to public geospatial data through a clean, accessible, and interactive web interface.



Key Motivations

Make data understandable

Provide access to insights

Apply geospatial analytics and geomarketing

Develop new skills

TARGET USERS AND SCOPE

- ✓ This platform was **designed for non-technical users** - like students, journalists, researchers, or citizens - who want to explore public geospatial data.

- ✓ It aims to make that **data easy to access and understand**.

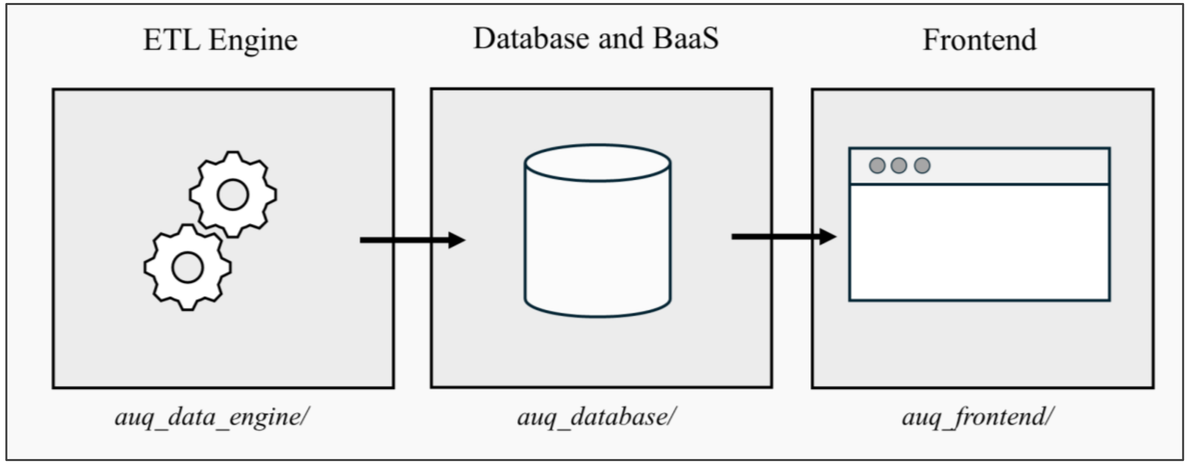
- ✓ The current version **focuses on Barcelona and Madrid**, with topics like economy, population, and surface.

- ✓ The system is **scalable** and ready to support more cities and datasets in the future.

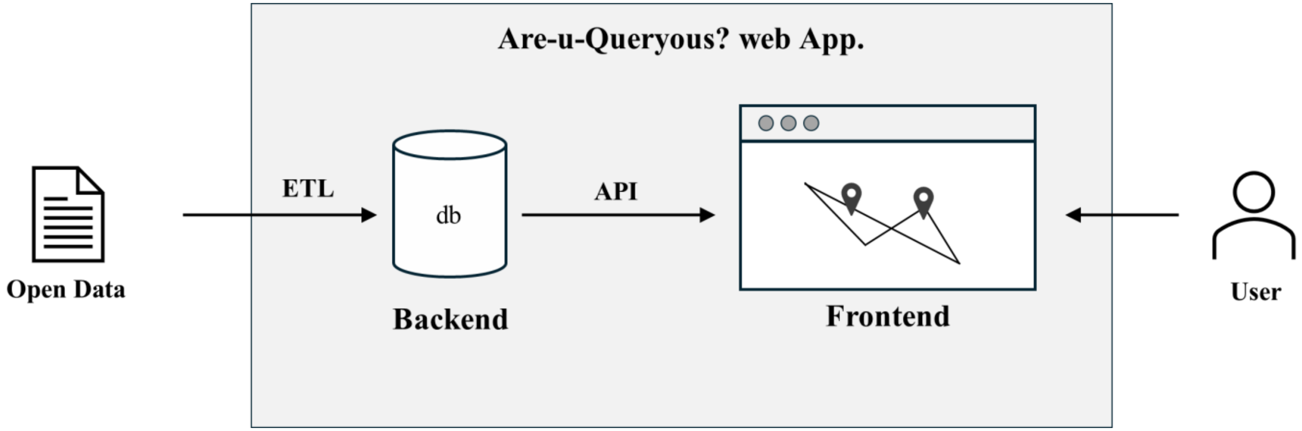
2. SYSTEM DESIGN

The system is designed as a **modular architecture** with three main components

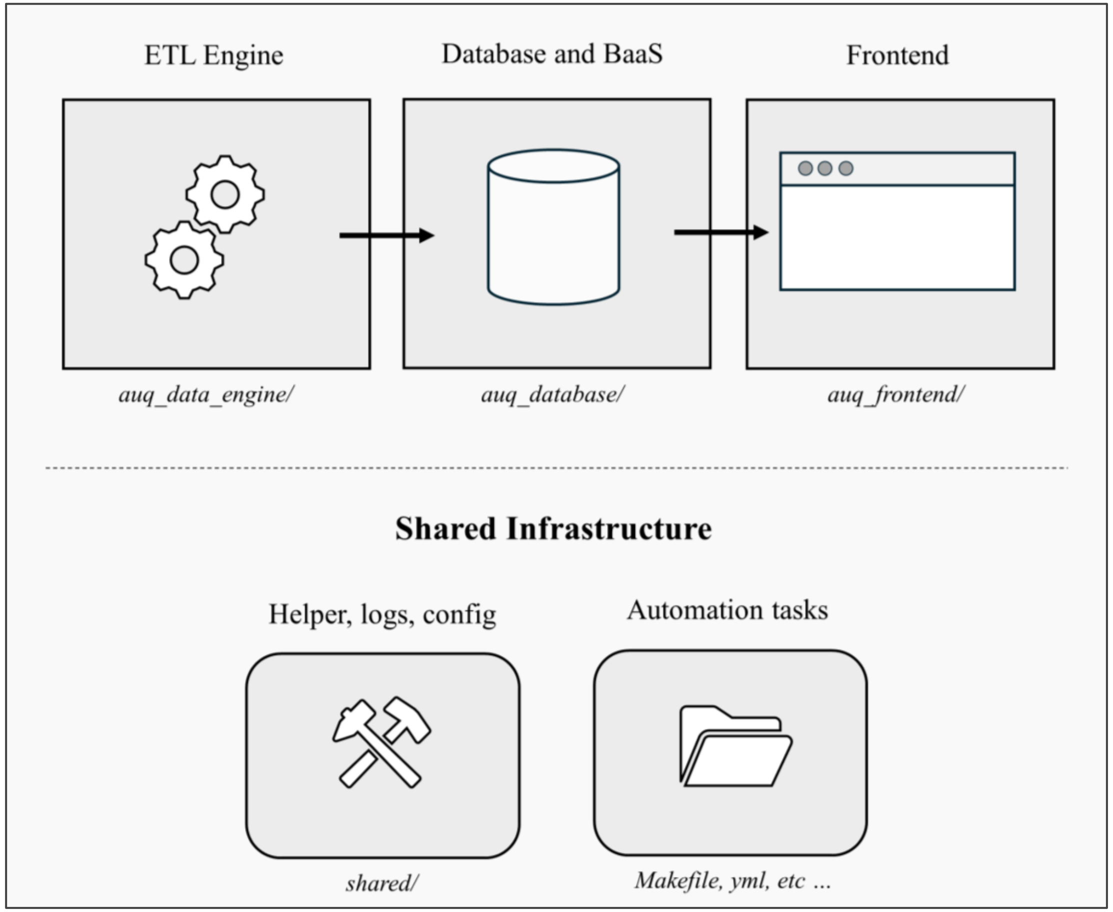
1. ETL engine.
2. Database and backend.
3. Frontend interface.



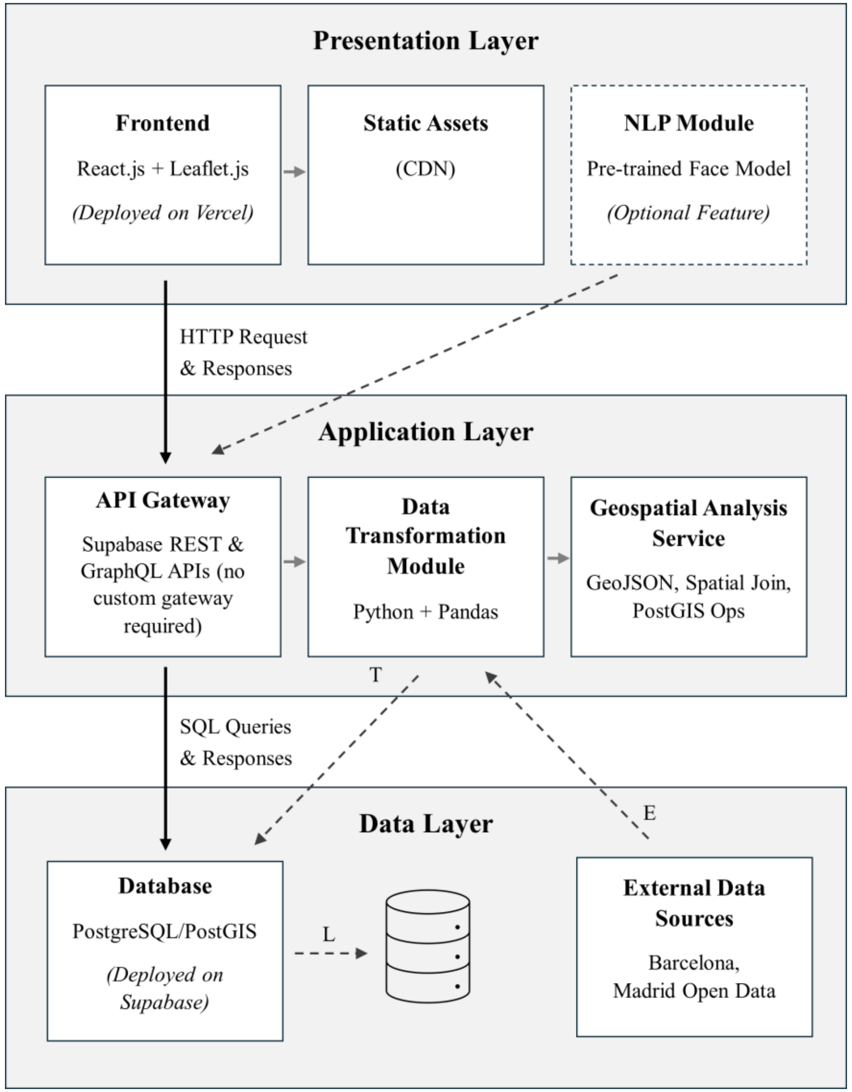
- ✓ First, the **ETL** engine **collects and transforms open data** from public sources, preparing it for the database.
- ✓ The **backend** is built using PostgreSQL and PostGIS, and is hosted on **Supabase BaaS**, which also provides authentication and APIs.
- ✓ The **frontend** is developed with **React and Leaflet** and is deployed on Vercel. It connects directly to the backend using Supabase's REST and GraphQL APIs.



All components are **connected** through a clean API layer and a **shared infrastructure** like logs, helpers, and automation scripts.



This modular design keeps the system **scalable and easy to maintain**.



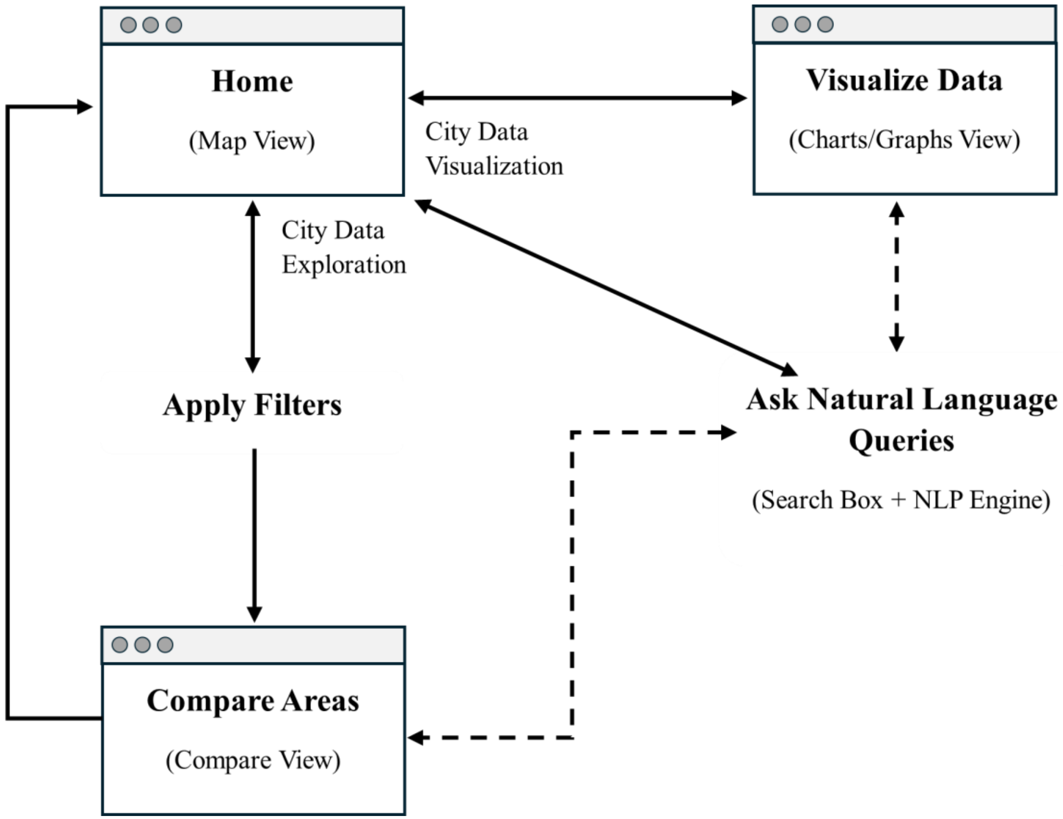
3. USER ROLES

The platform supports **three types of users**:

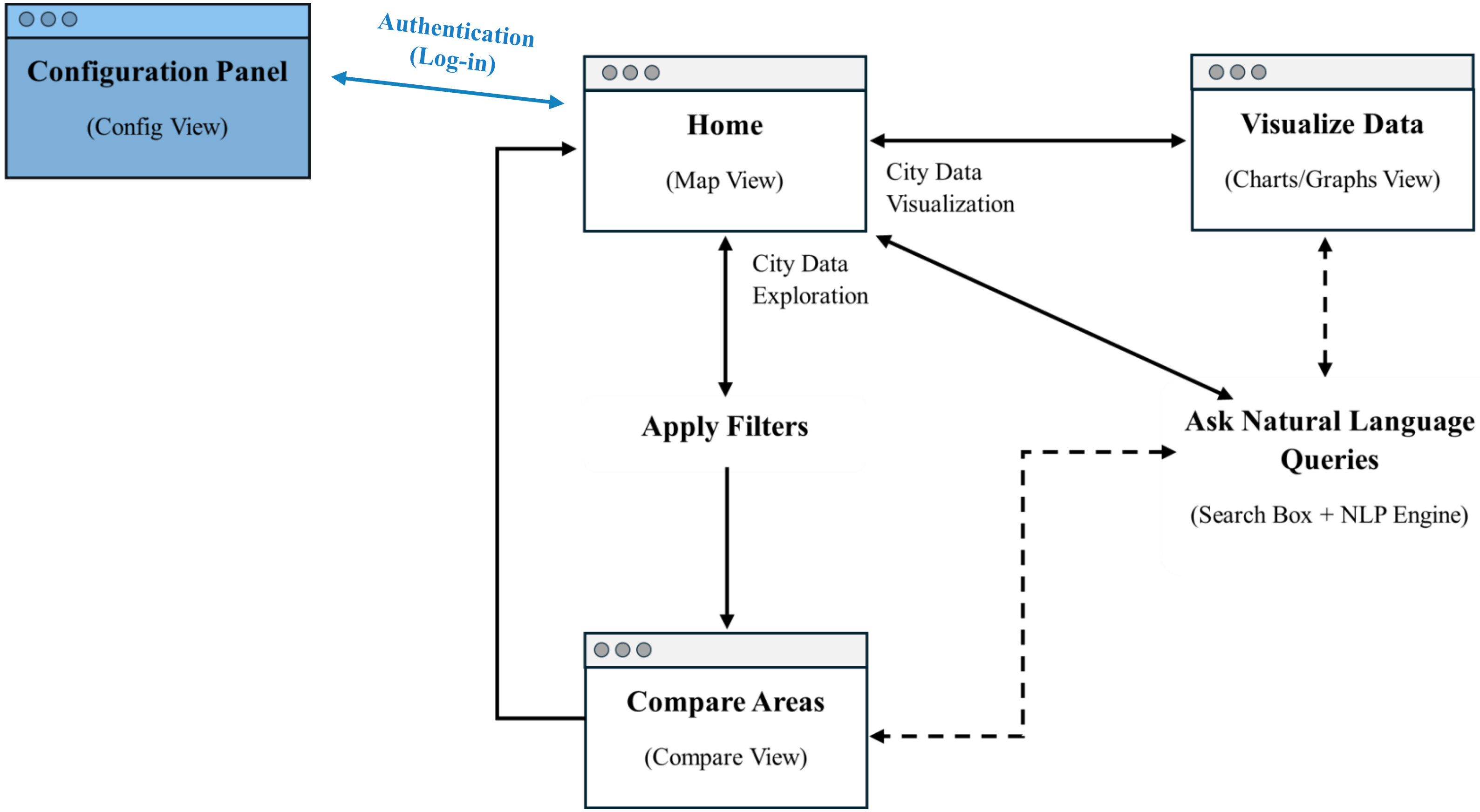
Anonymous	Authenticated	Admin
Explore data	Save preferences	Upload datasets
View map	Access more tools	Manage filters & features
No login	**Optional login**	**Monitor usage**

This separation keeps the app **secure, flexible, and easy** to use for each audience!

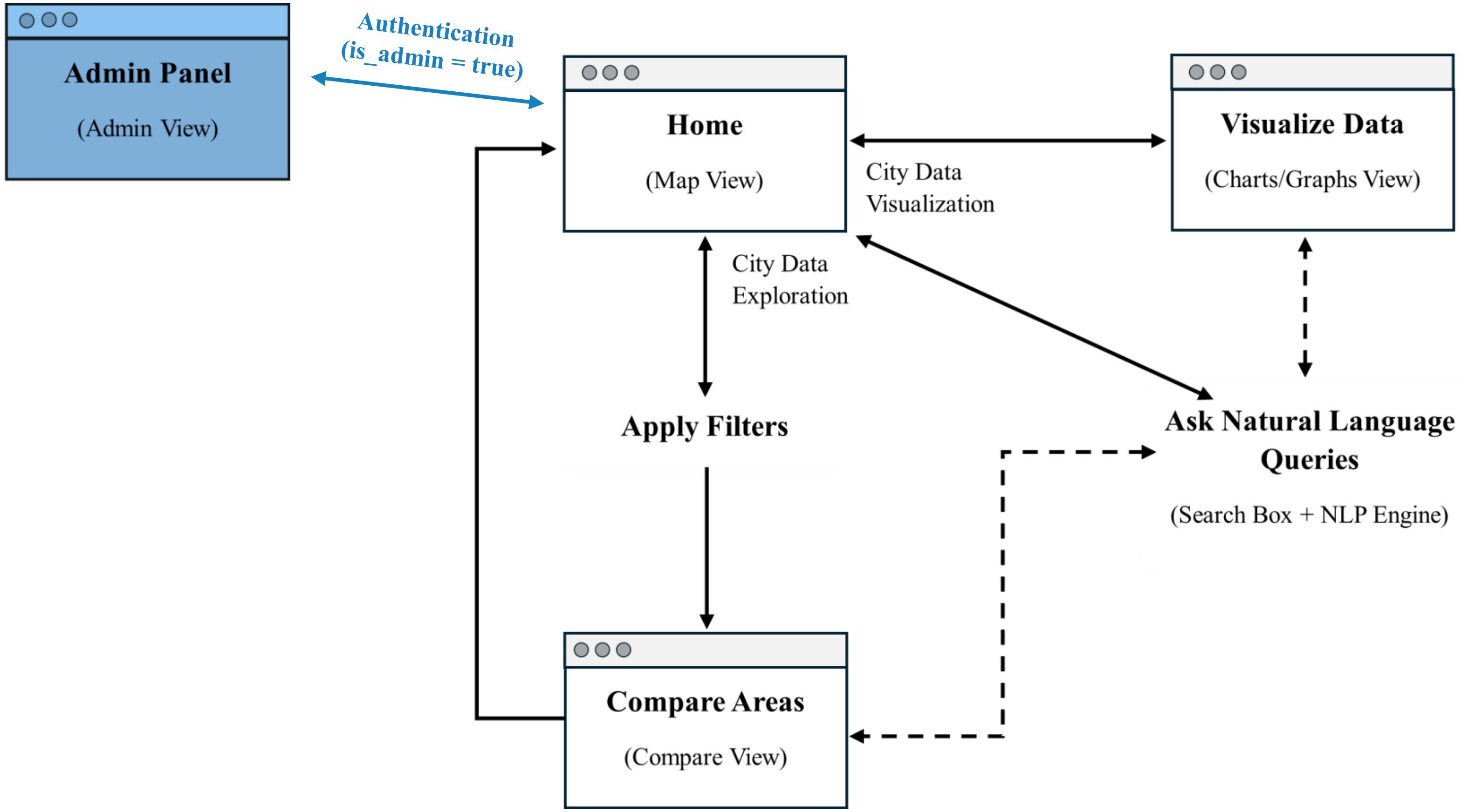
Anonymous users can explore the map and view public data without logging in.



Authenticated users get access to personalized views and saved preferences.



Admin users can manage global configuration and monitor platform activity.

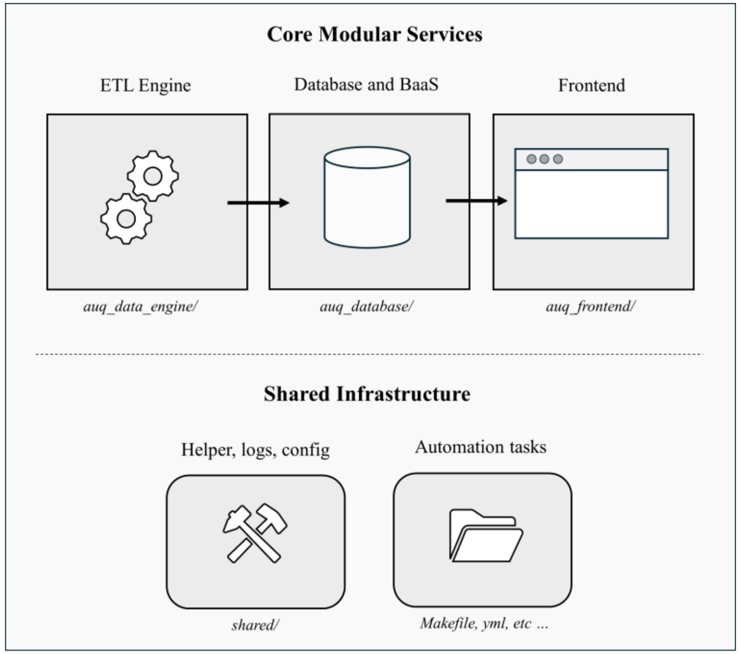


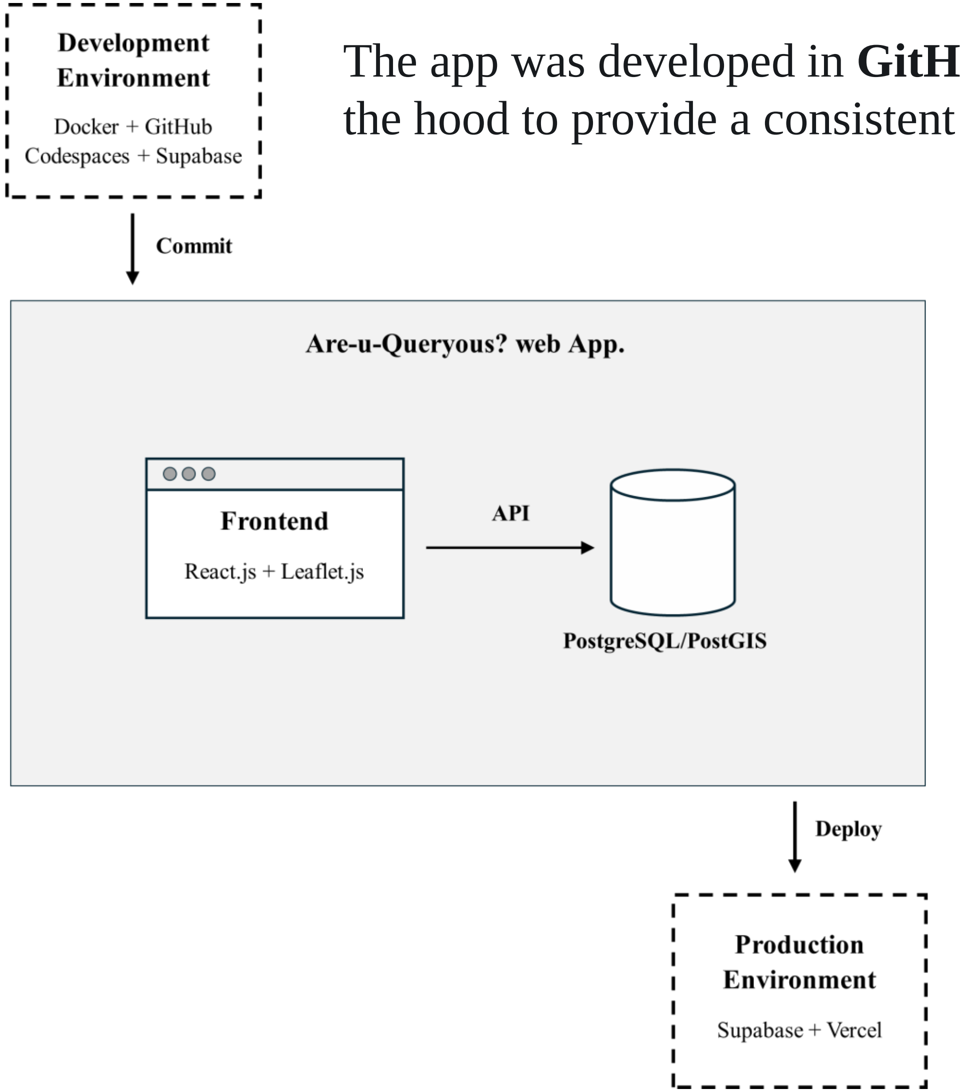
4. METHODS AND RESOURCES

The project **follows a monorepo structure**, where all components (database, ETL, frontend, and backend placeholders) **live in a single GitHub repository.** Each module is isolated in its own folder, but they all share helpers, logs, and configuration under a common shared/ directory.

Also, the development has applied modern development practices:

- Version control with Git and GitHub
- Semantic versioning and commit templates
- GitHub Actions to automate the ETL pipeline
- A Makefile to simplify local tasks like reset, test, or deploy





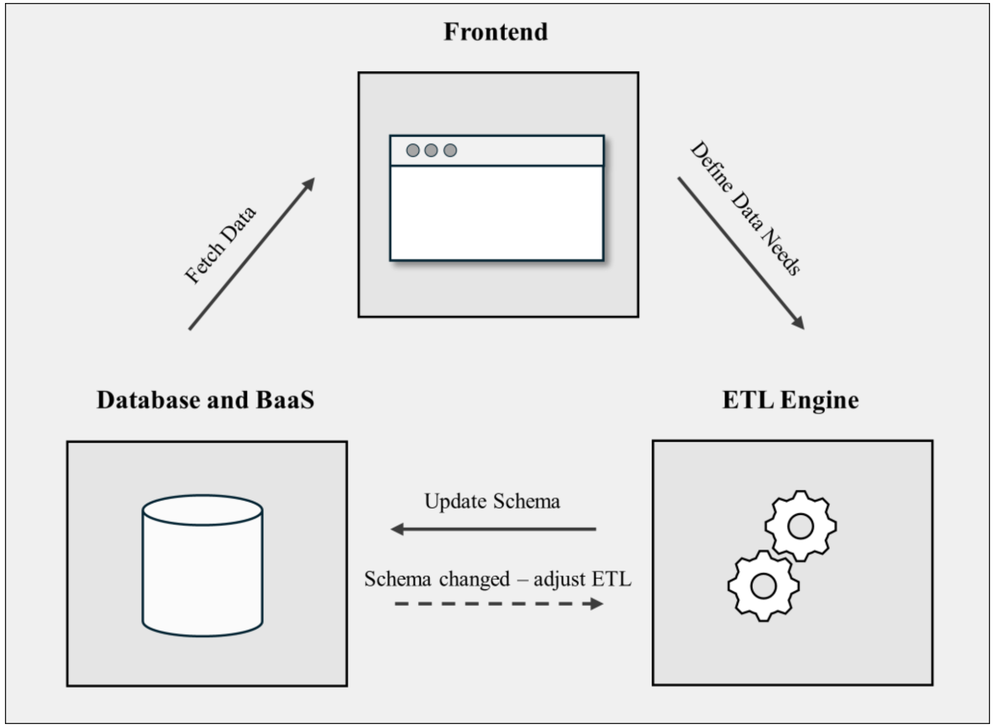
The app was developed in **GitHub Codespaces**, which uses Docker under the hood to provide a consistent environment.

All changes are version-controlled and pushed to **GitHub**.

From there, the system can be deployed automatically to production using **Supabase** for the backend and **Vercel** for the frontend.

The development started with the **initial plan and schema defined during the design phase in CAT2.**

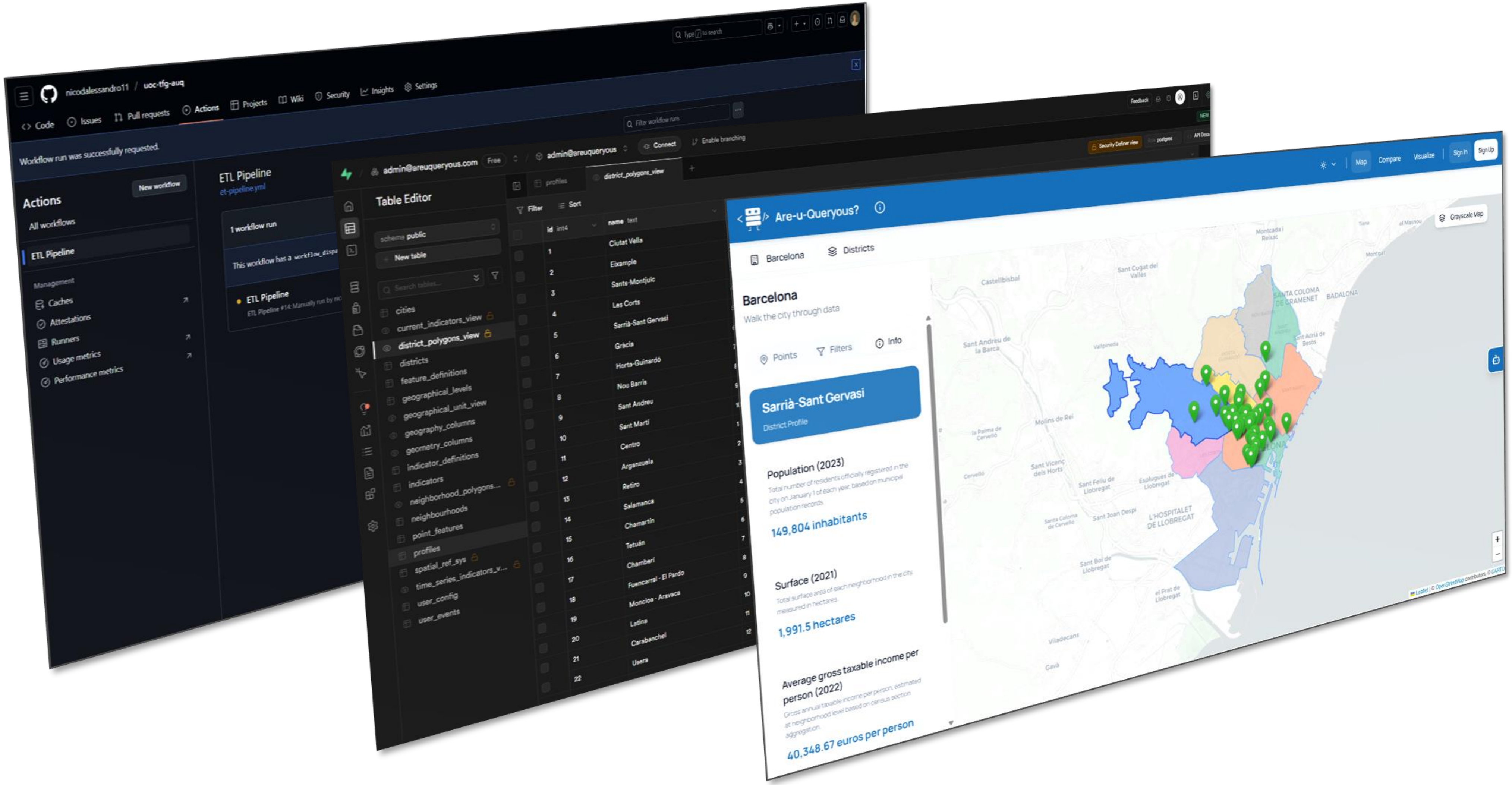
1. Based on this design we built the **AUQ Database**, with spatial support and test data.
2. Then the **AUQ Data Engine**, loading open datasets to fit that initial schema.
3. Next, the **AUQ Frontend**, where **we started using a feedback loop**:



Whenever the **frontend required a new feature, we adjusted the schema** and the pipeline!

This iterative method **helped align every component with the real needs** of the app!

5. RESULTS: AUQ WEB APP



Workflow run was successfully requested.

This workflow has a workflow_dispa

id int4	name text
1	Ciutat Vella
2	Eixample
3	Sants-Montjuïc
4	Les Corts
5	Sarrià-Sant Gervasi
6	Gràcia
7	Horta-Guinardó
8	Nou Barris
9	Sant Andreu
10	Sant Martí
11	Centro
12	Arganzuela
13	Retiro
14	Salamanca
15	Chamartín
16	Tetuán
17	Chamberí
18	Fuencarral - El Pardo
19	Moncloa - Aravaca
20	Latina
21	Carabanchel
22	Usera

Barcelona

Walk the city through data

Population (2023)

Total number of residents officially registered in the city on January 1 of each year, based on municipal population records.

149,804 inhabitants

Surface (2021)

Total surface area of each neighborhood in the city, measured in hectares.

1,991.5 hectares

Average gross taxable income per person (2022)

Gross annual taxable income per person, estimated at neighborhood level based on census section aggregation.

40,348.67 euros per person

5.1 AUQ DATA ENGINE

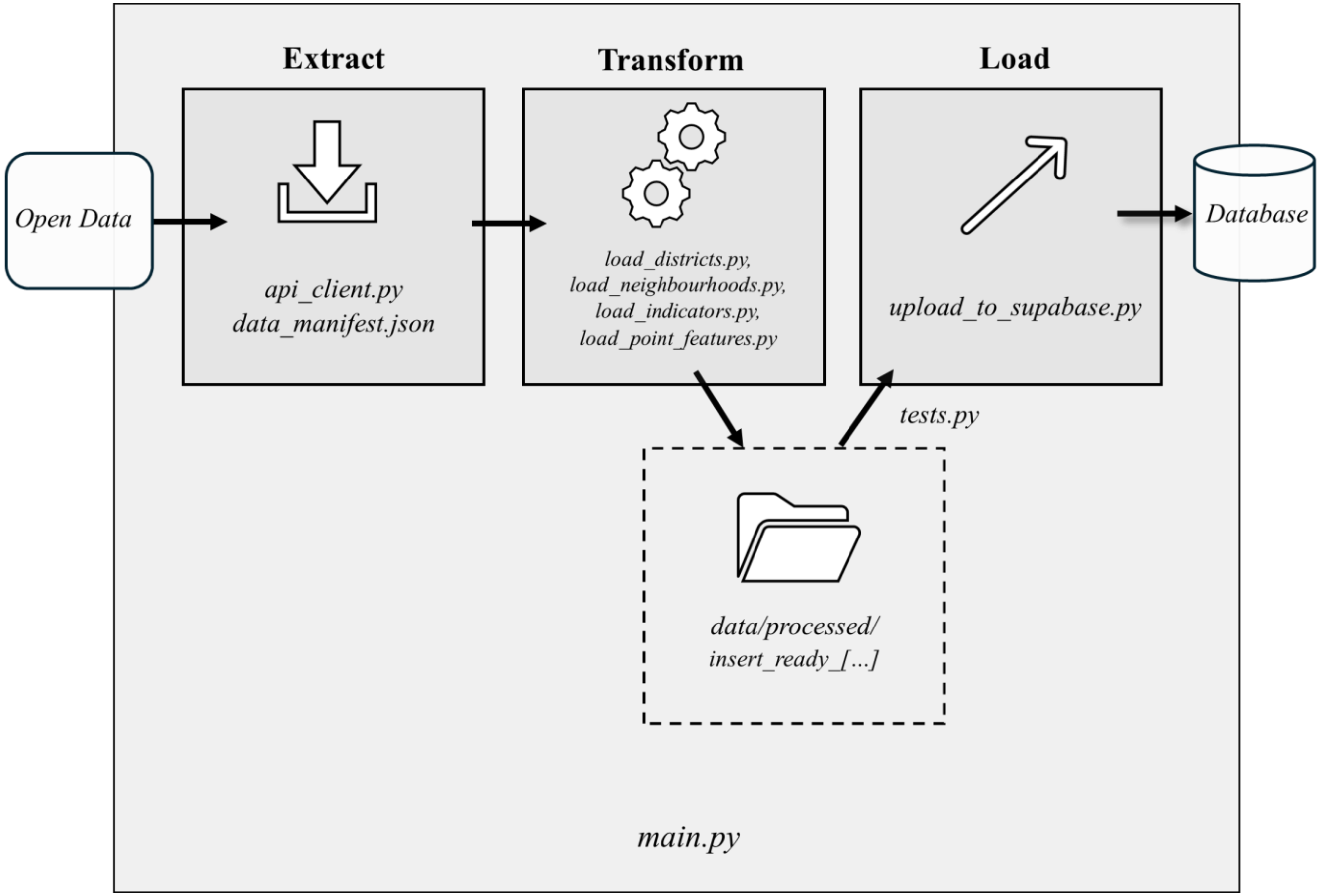
The **AUQ Data Engine** is a custom ETL pipeline developed in Python.

- ✓ It **reads open datasets** from Barcelona and Madrid portals, **cleans and transforms** the data using Pandas and GeoPandas, and prepares it in a standard format.

- ✓ It also **handles spatial joins and geometry** generation for the maps.

- ✓ The process is **fully automated using GitHub Actions** (so every time we push a new file or script, the pipeline runs, uploads the data to Supabase), and makes it ready for the app.

5.1 AUQ DATA ENGINE



5.2 AUQ DATABASE AND BACKEND

The **AUQ Database and Backend is built on Supabase**, using PostgreSQL with PostGIS extensions.

- ✓ This allows to **run spatial queries directly in the database** without a separate backend server.
- ✓ It also provides **ready-to-use APIs**, authentication, etc (all configured through SQL and policies).
- ✓ This made the backend **easy to maintain** and very **scalable**.

5.3 AUQ FRONTEND

The **AUQ Frontend is a React app** using Leaflet for the interactive maps and TailwindCSS for styling.

- ✓ It **dynamically loads layers** and filters based on metadata stored in the database.
- ✓ It **supports three main views**: the map, a compare view, and a visualization view with charts.
- ✓ All map layers are **interactive and fully synchronized** with user filters.
- ✓ The app is deployed on Vercel and loads instantly thanks to code splitting and CDN caching.

6. CHALLENGES & SOLUTIONS

- ✓ **Inconsistent and low-quality open data**
  Addressed with a flexible and reusable ETL pipeline.

- ✓ **Frontend bugs**
  Solved through better state management, TS linting, and rendering control strategies.

- ✓ **Flexible filtering across datasets**
  Designed a dynamic system driven by metadata stored in the database.

- ✓ **Tight deadlines**
  Switched to a frontend-first approach and relied on Supabase BaaS to accelerate development

Challenges were turned into design decisions that made the system simpler, faster, and more robust!

7.1 CONCLUSION

- ✓ **Functional MVP delivered**
  A full web platform with maps, filters, charts, and admin tools

- ✓ **Robust, modular architecture**
  Based on ETL, Supabase BaaS, and a React + Leaflet frontend

- ✓ **Inclusive by design**
  No login required, accessible UI, low barrier to entry

7.2 NEXT STEPS

- ✓ **Natural Language Query (NLP)**
  Let users ask questions like “Where is population growth highest?” using simple text

- ✓ **Add More Cities & Datasets**
  Expand coverage to new locations and indicators (e.g. health, environment)

- ✓ **Improve Mobile UX**
  Optimize layout, interactions, and performance for smaller screens

HAPPY QUERYING!

Thanks!

Nicolás Ezequiel D'Alessandro Calderon
Bachelor's degree in Techniques for Software Application Development
Universitat Oberta de Catalunya, 2025