

Acquirium: A No-Code Web Interface for Exploring Water Treatment Data

Clients:

Mete Saka, CS@Mines, saka@mines.edu

Dr. Gabe Fierro, CS@Mines, gtfierro@mines.edu

1. Background

Water treatment plants generate enormous amounts of operational sensor data. Accessing this data requires dealing with the metadata that explains it such as which sensor sits on which equipment, what units a reading is in, and how equipments are connected.

Our lab develops **Acquirium** (<https://github.com/DataDrivenCPS/acquirium>), an open-source Python framework that solves this by pairing a knowledge graph of plant metadata with time series measurements in one system. It runs from a single command, stores everything locally in embedded graph and time series databases, and ingests live or simulated plant data through drivers.

Today, the only way to get data out of Acquirium is to write Python code against its client API. The audience for this software are plant operators, water utility engineers, and environmental engineering researchers who are typically has little to no coding experience. This project focuses on closing this gap by introducing a no-code interface.

2. Project Description

Students will design and build a lightweight web interface that ships inside Acquirium python package itself, so that a non-programmer can explore plant metadata, build queries visually and retrieve operational data. The interface should be served by Acquirium's existing FastAPI server and launched with a single command.

The team is expected to design and implement the interface end to end including user research with our water-domain collaborators, UX design, frontend implementation, the supporting API endpoints, packaging, and documentation.

2.1 Core Goals

- **No-code query builder:** Construct queries over the plant metadata and data by pointing and clicking or text matching, with the generated query visible so that users who do write code can copy it into their own scripts.
- **Exploration interface:** Build an interface that allows users inspect intermediate results and guide users while building queries. Such as show what matches at after each selection, let users see what filtering options exist or which paths are available for the next step.
- **Documentation and guided tutorials.** Serve our existing documentation within the web interface, with runnable walkthrough examples over bundled example data and queries.

2.2 Bonus Goals

- **Embedded visualization and dashboards:** Implement a visualization interface to plot sensor time-series data in browser. This plots should be generated by the query interface to retrieve the data, and saved as a simple Grafana-style dashboard that can be reopened.
- **LLM integration:** A chat interface that uses an LLM to build queries or retrieve data directly. This interface can use a locally running model, or connect to an LLM API.

2.3 Packaging Requirements

- The interface ships as part of the Acquirium Python package. Installing PyPi Acquirium package should install the UI.
- The interface must work against Acquirium's default API on a laptop, offline, without any external database and no Docker requirement at runtime.
- Publishing a Docker image and Compose profile as a convenience option is welcome, but must remain

optional.

- Cross-platform (macOS, Linux, Windows), with tests and CI integration matching the existing repository conventions.

3. Deliverables

- Working, documented web interface merged into the upstream open-source repository as reviewed pull requests.
- Automated tests and updated user documentation (README plus in-app help).
- A short demo walkthrough using the example deployment, suitable for showing to non-technical water stakeholders.

4. Desired Skill Set

- Python; some exposure to a web framework such as FastAPI or Flask is a plus.
- Frontend web development — the team chooses the stack (React, Svelte, HTMX, plain JS all fine) as long as it packages cleanly.
- Linux, Docker, Git
- Interest in UX design and willingness to do usability testing with real non-technical users.
- Curiosity about data science, databases, knowledge graphs, RDF/SPARQL, ontologies, and time series data. No prior experience expected.
- No water or environmental engineering background is needed.

5. Preferred Team Size

3–5 students.

6. Location of Work and Mentorship

Weekly meetings with the client can be held on campus or over Zoom, and day-to-day work can be done wherever the team prefers. Beyond the scheduled meetings, we work on Acquirium daily and are available between sprints for design review, walkthroughs of the graph and API internals, and code review on pull requests.

7. NDA and Intellectual Property

No NDA is required and no assignment of intellectual property is requested. Acquirium is MIT-licensed and developed in the open; student work will be contributed upstream under the same license. Everything the team builds will be publicly visible on GitHub and citable in a portfolio.