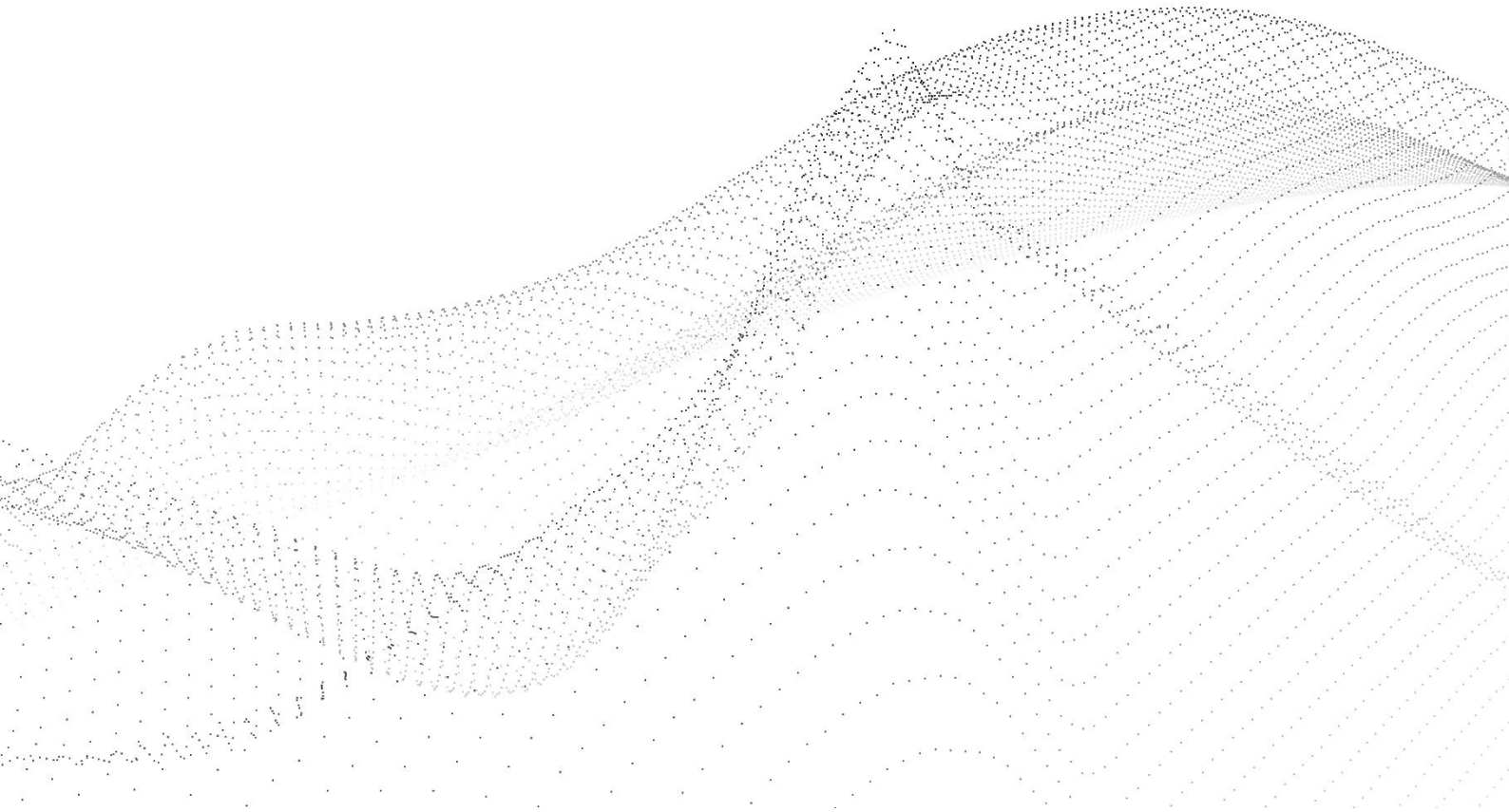
ATSYS



University of Adelaide

2025 Industrial Project

No-Code Solution for InfluxDB

2025 Industrial Project

1.1 Objective Summary

Develop a user-friendly, no-code web interface for InfluxDB[1], an open-source time series database. This interface should simplify the process of querying and trending data for users without programming expertise by providing the ability to trend without writing queries. Users should be able to trend by interacting with the interface through simple actions such as clicking or dragging-and-dropping.

The interface should also integrate with Grafana[2], enabling users to customise, save and edit trends with more features than the built-in trending tool of InfluxDB.

Further, the interface should be able to provide an alternative organisation for the data tags. Instead of grouping by the default buckets and measurements, users should be able to create their own logical hierarchy to organise the data.

Finally, the interface should have APIs available for users to query data programmatically either using the bucket and measurement, or with custom structure. The APIs should be optimised for speed, especially when querying a large amount of data.

1.1.1 InfluxDB

InfluxDB is a purpose-built time series database designed to handle high write and query loads. It is optimized for storing and analysing large volumes of timestamped data, making it ideal for applications such as DevOps monitoring, IoT sensor data, and real-time analytics.

1.1.2 Grafana

Grafana is a popular open-source platform for data visualization and monitoring. It's widely used to create dashboards that display time series data from various sources, including InfluxDB. Grafana offers a range of customizable charts and graphs, supports multiple data sources, and includes features like alerting and user-friendly dashboard creation.

1.2 Background

As an open-source database optimised for time series with a range of data connectors, InfluxDB is a good option for companies looking to store time series data with different origins. However, InfluxDB's internal language, Flux, presents a significant barrier for non-programmers due to its complexity and dissimilarity to SQL. Users unfamiliar with Flux often find it hard to query, explore, and trend data.

Users can create displays and trends in InfluxDB, but they are not shareable. Therefore, Grafana is often used in conjunction with InfluxDB for data visualisation. Unfortunately, the barrier also extends to Grafana given the lack to reference of Flux and limited debug functions in Grafana.

Moreover, the current data structure in InfluxDB stores data tags in measurements and buckets, but they cannot be easily reorganised based on their relationships. If the tags are created automatically, their name might even be meaningful, often users have to rename them in various Grafana displays, which is not controlled and hard to track.

1.3 Requirements and Specification

This project aims to address these limitations by creating an intuitive web interface that allows users to easily visualize data with Grafana without writing Flux code, and to create a custom structure to organise the data tags that is independent from the existing bucket and measurement.

This interface is independent from the inbuilt InfluxDB web interface.

The preliminary requirements include:

1. **Single-Page Design**: All functionalities should be available on one page for ease of use and quick access.
2. **Visual Query Builder**:
   - Intuitive selection of bucket, measurements and fields with actions such as click, drag-and-drop, etc.
   - Easy-to-use filters and time range picker
3. **Automatic Code Generation**: Based on user selections, the interface should generate the necessary Flux code in the background, or if the user chooses, display the code.
4. **Data Visualization**: Display the resulting data trends graphically, potentially leveraging Grafana's functionality.
5. **Authentication**: Implement secure authentication at the same level as the existing InfluxDB authentication system. Existing InfluxDB should be able to login using the same credential and have the same level of access and permission.
6. **Data Organisation:** Ability to reorganise data based on custom hierarchy instead of existing bucket and measurement. A data tag should be able to be present at multiple level of the hierarchy.
7. **(Extension) Grafana Integration:**
   - Connect to existing Grafana installations.
   - Allow users to save current displays to Grafana.
   - Enable editing of existing Grafana displays within the app.
8. **(Extension) API Connectivity:**
   - Create APIs for users to query data programmatically.

Future stretch goals include processing more complicated functionalities such as pivoting or joining fields from multiple measurements.

1.4 Scope Limitations

- The project will focus solely on query building and data visualization. Other InfluxDB functionalities, such as connectors, are out of scope for this interface.
- It is recommended to use InfluxDB OSS version 2.7[3] and Grafana v9.5.3[4].
- The preferred operation system for InfluxDB and Grafana installation is Windows.
- The web application should be developed in JavaScript, but there is no limitation on the server-side architecture or tech stack.

1.5 Deliverables

1. A functional, web-based no-code interface that generates InfluxDB queries, connects to Grafana and has proper authentication.
2. User documentation explaining how to use the interface.
3. Technical documentation detailing the implementation and integration with InfluxDB.

1.6 Reference

[1] InfluxDB Website: https://www.influxdata.com/products/influxdb/

[2] Grana website: https://grafana.com/grafana/

[3] InfluxDB OSS v2 documentation: https://docs.influxdata.com/influxdb/v2/

[4] Grafana v9.5 documentation: https://grafana.com/docs/grafana/v9.5/