Data Lakehouse Architectures Compared

Technologies, Formats, and Challenges

Student

Myriam Assunção

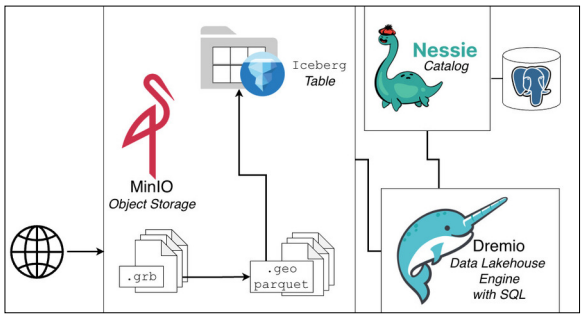
Introduction: In recent years, the demand for processing large volumes of data has increased significantly, driven primarily by machine learning and artificial intelligence initiatives. While traditional database clusters can support analytical workloads, providing consistent and concurrent access to shared data across multiple tools and user groups within a single platform remains a major architectural challenge. Data lakehouse architectures address this issue by combining the scalability and flexibility of data lakes with the reliability and performance of data warehouses.

Approach: This project investigates the data lakehouse paradigm through both theoretical analysis and practical implementation using open-source technologies. It examines architectural principles, core components, data and table formats, as well as the benefits and limitations of lakehouse systems by the example of Dremio as query engine, PostgreSQL as database system, MinIO as object storage and Nessie as catalog component. A case study on identifying optimal stargazing locations in Switzerland illustrates the interaction of typical lakehouse components. Open datasets, including streetlight data from OpenStreetMap and historical meteorological data from COSMO Regional Reanalysis, were ingested into object storage and transformed into analytical formats such as GeoParquet. Apache Iceberg was used to manage tables and metadata, while a SQL-based lakehouse engine enabled interactive querying. Throughout the implementation process, Nessie served as the catalog and version control layer. This enabled consistent management of table metadata and reproducible access to data through tagged dataset versions. The results were visualized using the Python library Folium, which generates interactive, browser-based maps as static HTML files.

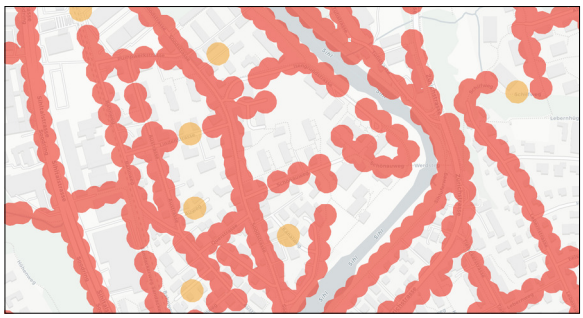
Result: GRIB files were successfully transformed into GeoParquet using Python libraries, and Iceberg tables were created via SQL for structured metadata management. Over 90K streetlights from OpenStreetMap were converted to GeoParquet and ingested into Iceberg. Additionally, three years of hourly COSMO-REA6 cloud cover data were processed from GRIB to NetCDF format, reprojected to the coordinate reference system Web Mercator, cropped to the boundaries of Switzerland, and aggregated weekly. This pipeline exposed conversion overhead but confirmed the analytical benefits of columnar storage and SQL predicate pushdown. Interactive SQL queries in Dremio validated the separation of storage and compute, while Nessie enabled reproducible data states through branching. However, limitations of the Community Edition and browser-based map rendering required dataset partitioning, illustrating practical scalability constraints.

The implementation highlights both the modularity and complexity of open-source data lakehouse ecosystems. Key challenges include data and metadata transformation overhead, performance trade-offs related to file formats and partitioning, and the operational effort required to integrate decoupled storage and compute layers. While open-source lakehouse solutions offer flexibility and reduce vendor lock-in, they require careful architectural planning and substantial engineering effort. No single configuration is optimal; decisions must align with workload characteristics and scalability needs.

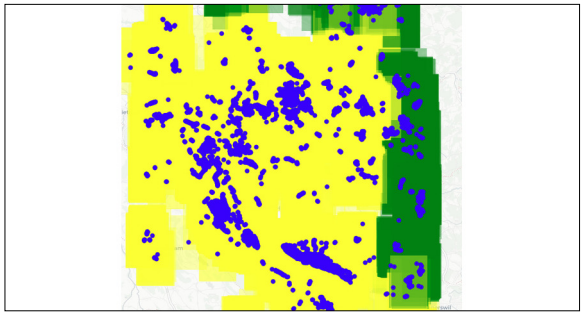
Workflow with technologies used in this project.
Own presentment



Section of streetlights in Zurich (Base map (c) OpenStreetMap). Red = near standing lamps. Orange = single lamp.
Own presentment



Streetlights & weather cells in Zurich. Yellow = Middle cloud coverage. Green = Low cloud coverage. Blue = Street light.
Own presentment



Advisor
Prof. Stefan F. Keller

Subject Area
Computer Science