

Advanced Database Manipulation with Python using DuckDB

YANG WANG

PTUA 10 MARCH 2025

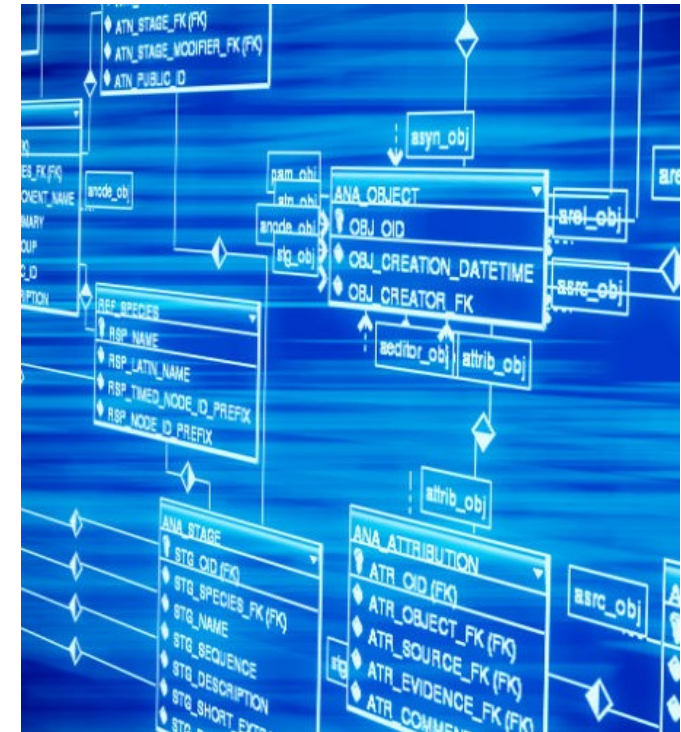
Database and Spatial DB

Database - Data management

- Collecting, storing, and using data securely and efficiently
- Inform decision making
- E.g ArcGIS -> file geodatabase; mobile geodatabase (SQLite)
- E.g QGIS -> PostGIS (spatial extension of PostgreSQL) and SpatialLite

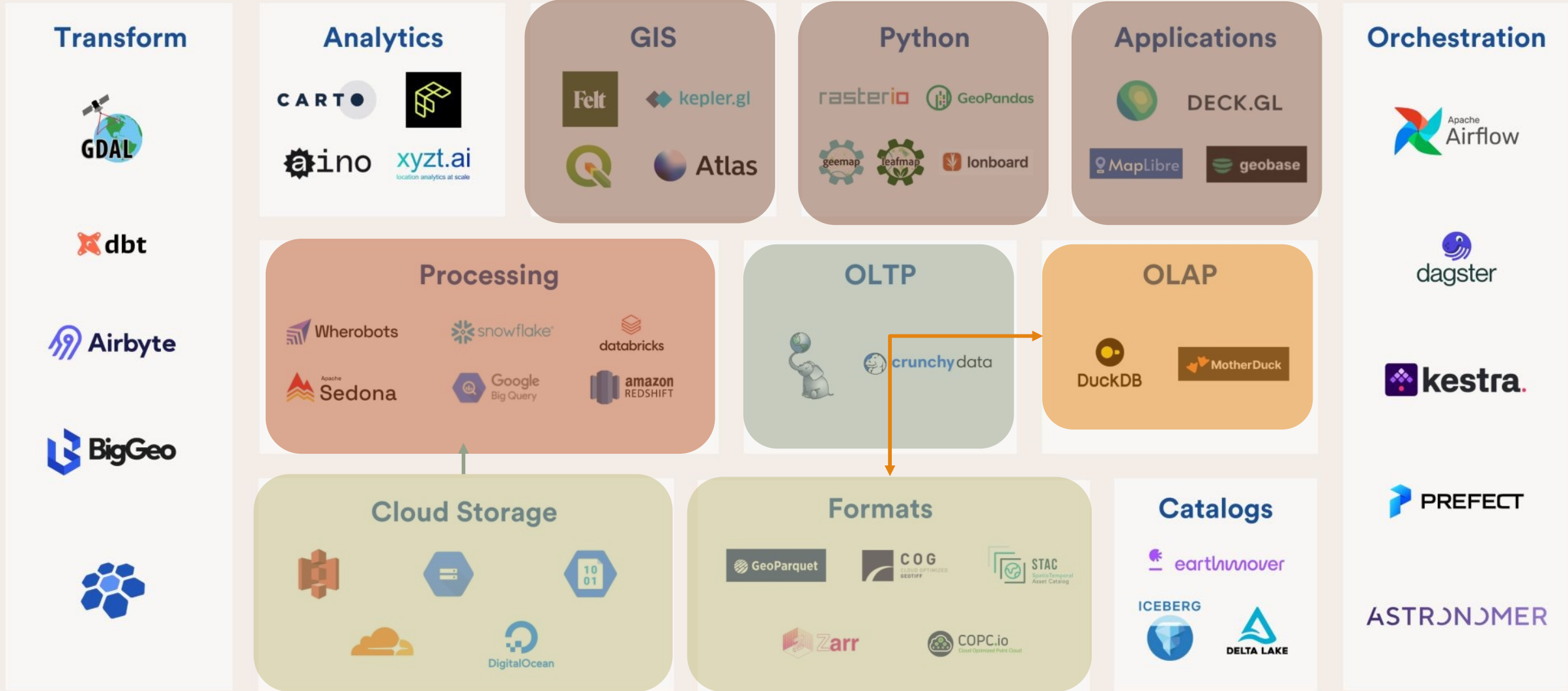
Spatial DB - Georeferenced spatial database

- Collecting, storing and using geographic data especially in GIS
- Normally relational and object-relational DB
- Functions e.g. measurement, geoprocessing, geometry, etc
- Spatial index to optimize spatial query by multi-dimensional ordering
 - BSP-tree, K-d tree, m-tree, etc
 - PostGIS -> R-tree

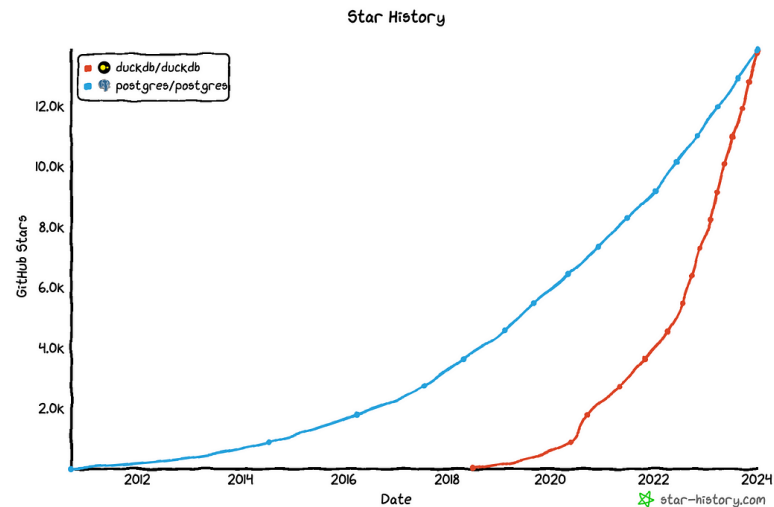


Geospatial Data Stack

Late 2024



forrest.nyc



AN IN-PROCESS DB
(THAT RUNS IN THE
PROGRAM ITSELF, IT
HAS NO
INDEPENDENT
PROCESS,
RESEMBLING SQLITE)



FOR OLAP
(ADJUSTED TO
ANALYTICAL LOADS),



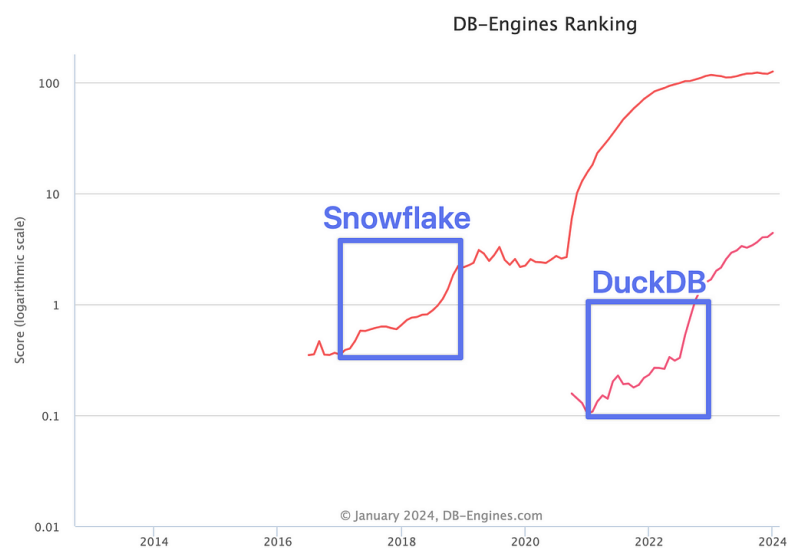
HANDLES DATA IN
TRADITIONAL
FORMATS (CSV,
PARQUET),

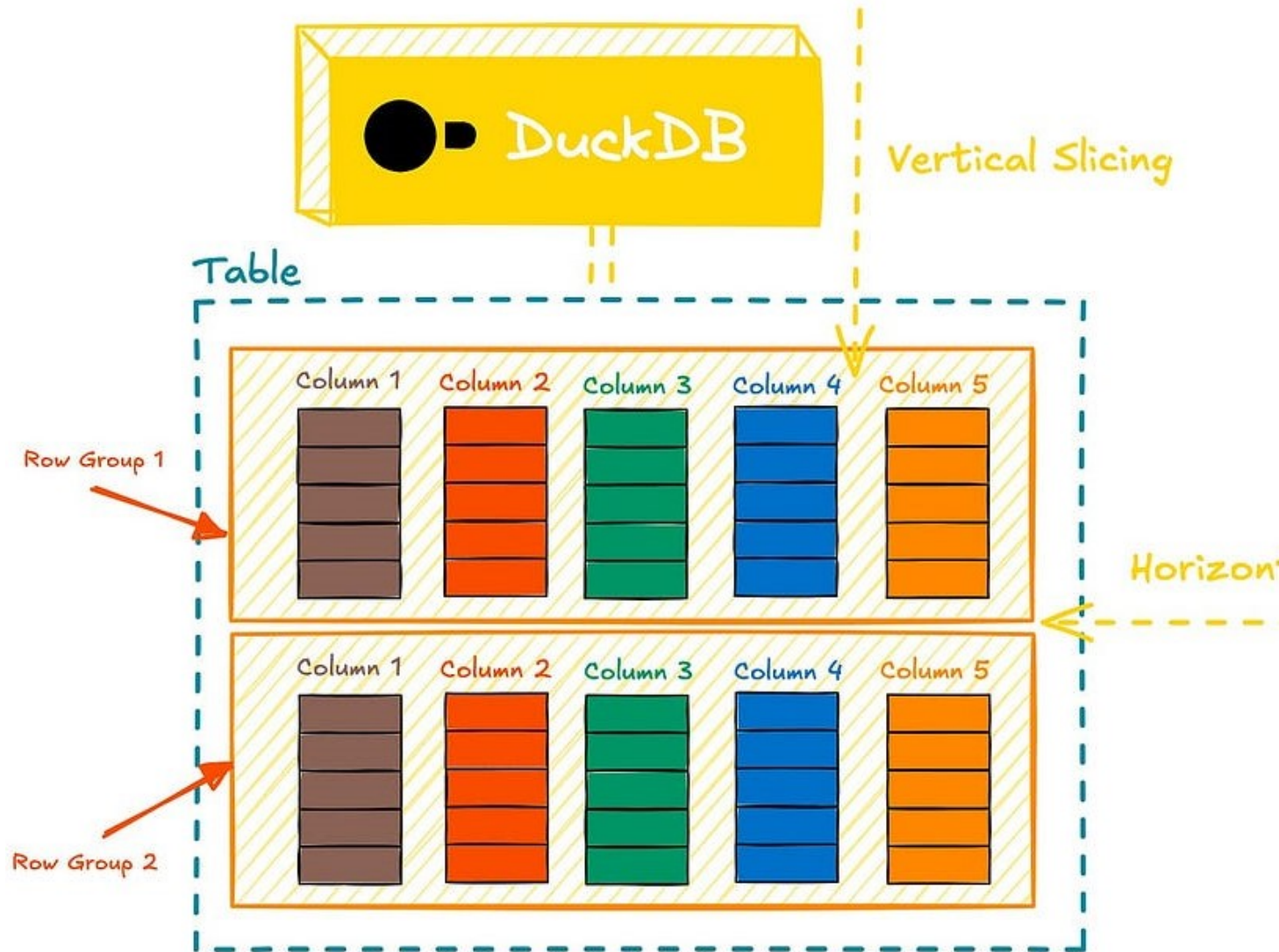


OPTIMIZED TO
HANDLE LARGE
VOLUMES OF DATA
(PARQUET) USING
THE POWER OF A
SINGLE MACHINE
(THAT DOESN'T
NEED TO BE VERY
POWERFUL).



[HTTPS://DUCKDB.ORG/](https://duckdb.org/)



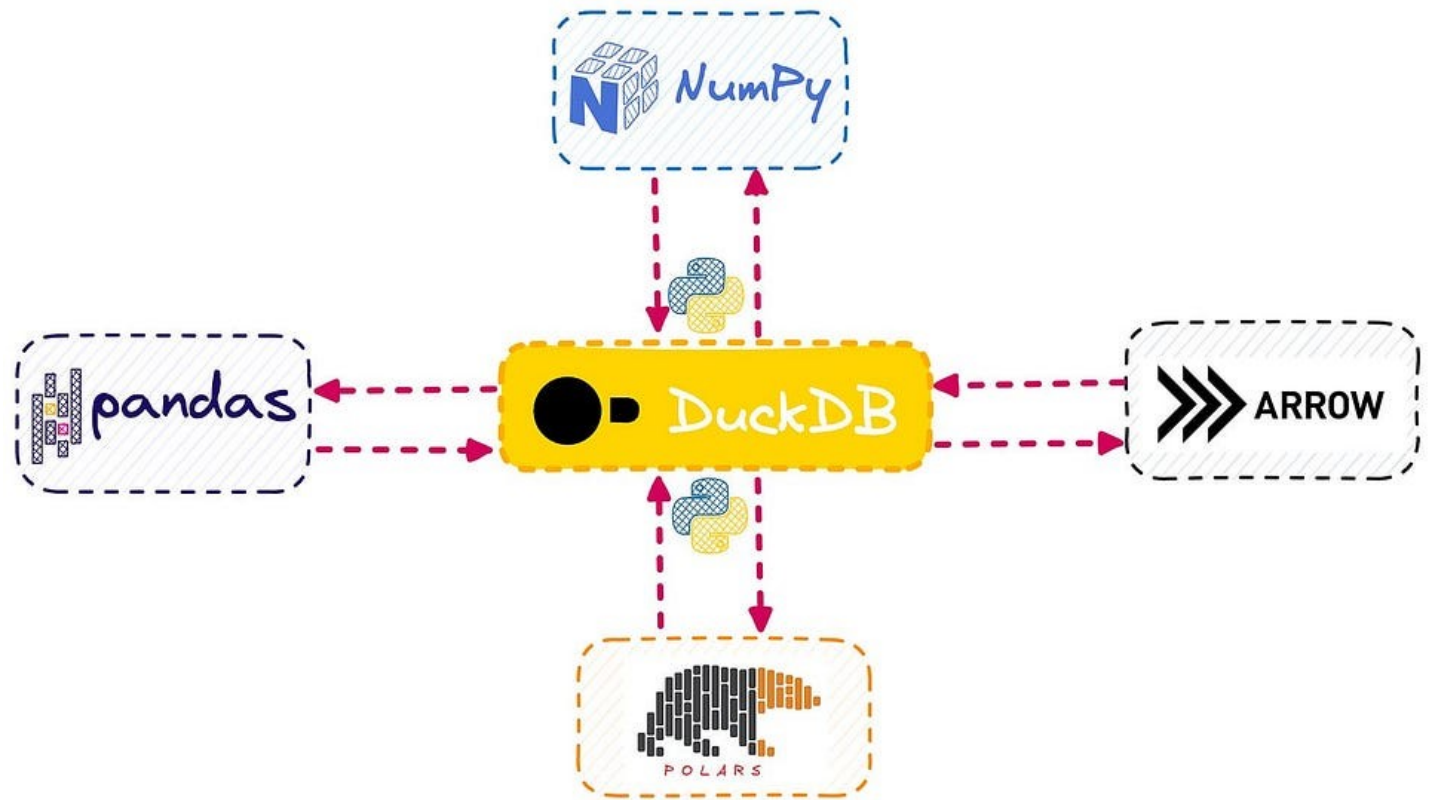


A Columnar OLAP Database

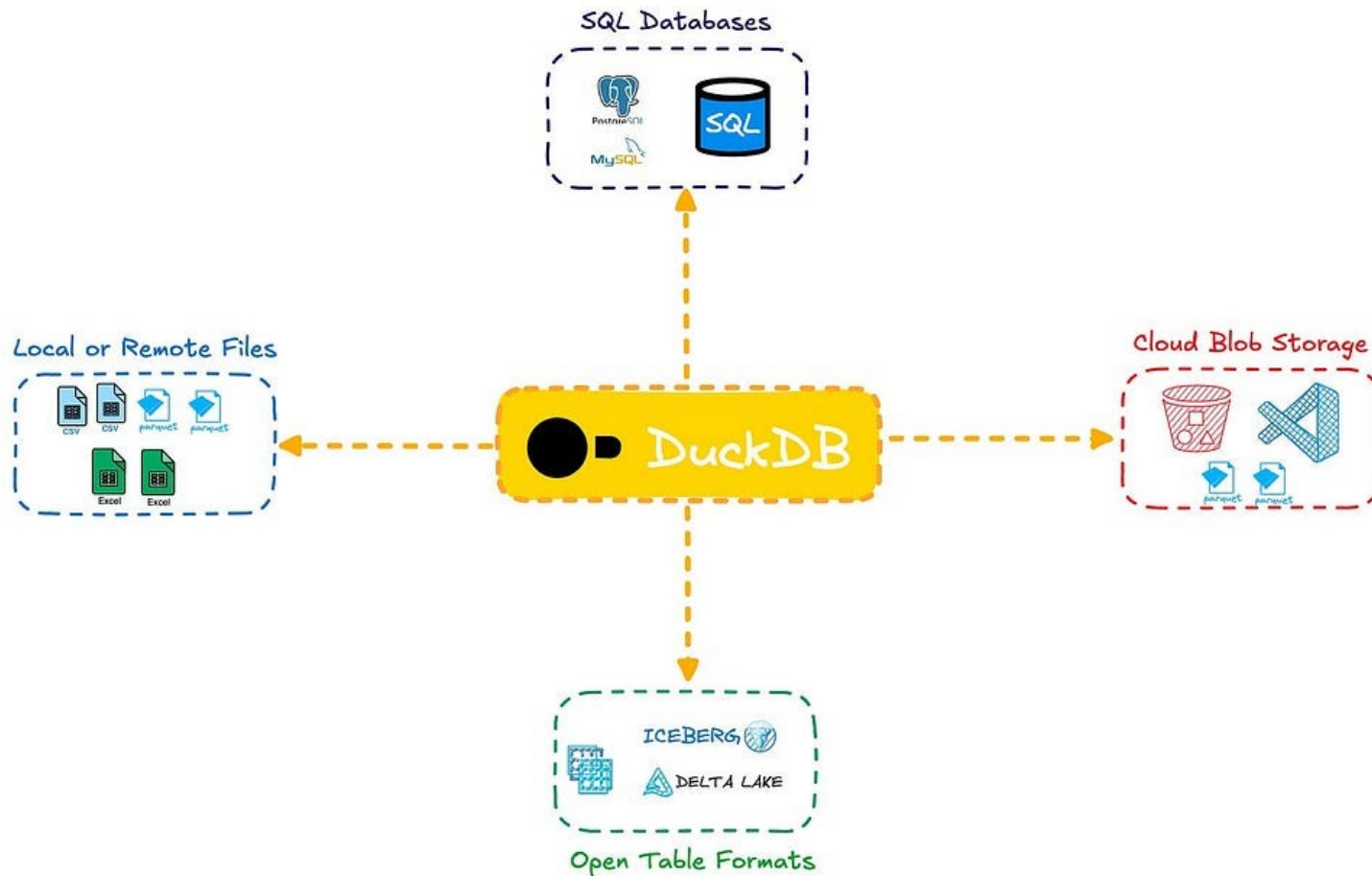
- A row-columnar structure
- Data is sliced into *row-groups*
- Within each group, columns are stored separately and compressed
- Similar to popular binary formats like Parquet and ORC.

Interoperable SQL-Powered Dataframes

- ❑ Integrates seamlessly with popular dataframe libraries like **Pandas** and **Polars**, allowing efficient in-memory operations
- ❑ Run SQL queries directly on Python dataframes. You can query **Pandas**, **Polars** and **Apache Arrow** dataframe objects as though they were SQL tables.
- ❑ Some frameworks such as Apache Arrow, DuckDB uses zero-copy mode for fast conversion



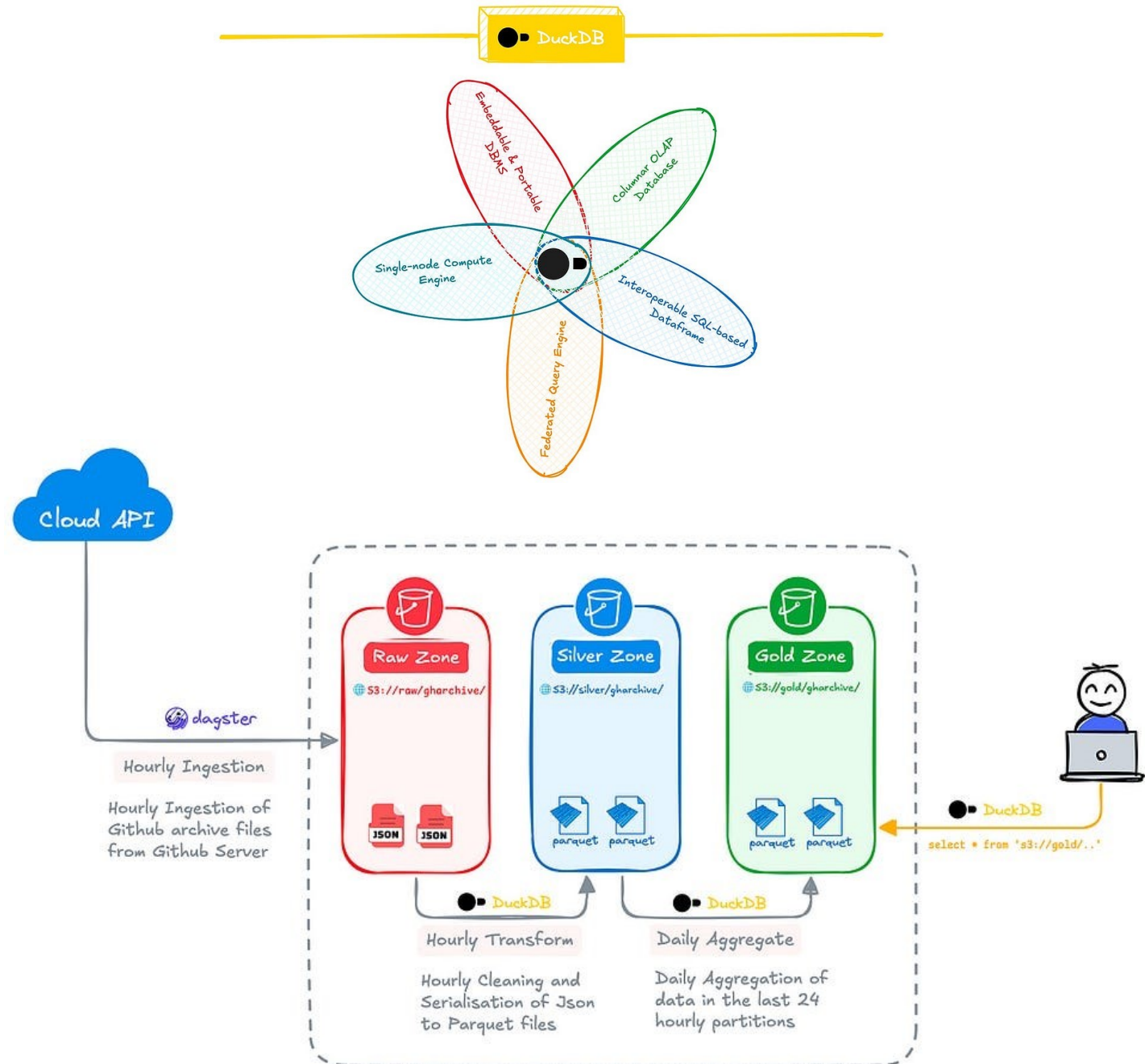
A Federated Query Engine



- ❑ Efficient way to query external data systems through its [extensions](#).
- ❑ Directly query DBMS systems like **MySQL** and **Postgres**, open data files like **JSON**, **CSV**, and **Parquet** files stored in cloud storage systems like **Amazon S3**, and modern open table formats like **Apache Iceberg** and **Delta Lake**.
- ❑ Create persistent **Views** over the external tables or data files, like a ***read-only external table***.

A Single-Node Compute Engine

- ❑ Perform ephemeral batch transformations.
- ❑ Efficiently serialise raw data (e.g., JSON or CSV) into optimised formats like Parquet, and then transform or aggregate that data.
- ❑ E.g. aggregate 100M rows in 1min



DuckDB – spatial extension

Benefit

- operate, transform and join your geospatial data alongside your regular, unstructured or time-series data using DuckDBs rich type system and extensions like JSON
- spatial queries involving geometric predicates and relations translate surprisingly well to SQL, which is all about expressing relations after all!
- all the other benefits provided by DuckDB such as transactional semantics, high performance multi-threaded vectorized execution and larger-than-memory data processing.

What in it?

- Similar to database systems such as [PostGIS](#) or [Spatialite](#).
- geospatial libraries, [GEOS](#), [GDAL](#) and [PROJ](#), which provide algorithms, format conversions and coordinate reference system transformations respectively
- leverage GDAL to provide a set of table and copy functions that enable import and export of tables from and to 50+ different geospatial data formats (so far!), including the most common ones such as Shapefiles, GeoJSON, GeoPackage, KML, GML, WKT, WKB, etc.

Not yet super mature

- leveraging a lot of great open source software, e.g. [GEOS](#), which is the same spatial engine that PostGIS uses
- leveraged to great effect was OGR/GDAL. You can easily import or export any format that OGR supports
- not ready to commit to full compliance with the OGC Simple Feature Access

Overall benefit for us

‘The geospatial world need to meet people more than halfway, enabling them to stay in their existing workflows and toolsets.’

---Chris Holmes (Product Architect @ Planet, Board Member @ Open Geospatial Consortium, Technical Fellow @ [Radiant.Earth](https://radiantearth.com))

Useful resources

Official site

- <https://duckdb.org/>
- <https://duckdb.org/docs/>

Official training site

- <https://motherduck.com/>
- <https://motherduck.com/duckdb-book-brief>

Open Geospatial Solutions <https://geog-414.github.io/>

- https://geog-414.github.io/book/duckdb/01_duckdb_intro.html
- Open Geospatial Solutions youtube channel
- <https://youtu.be/A4TOAdsXsEs?si=SUiBe3FHSijIMyRK>

awesome DuckDB

- <https://github.com/davidgasquez/awesome-duckdb>

Peer-Reviewed Papers and Thesis Works

[Runtime-Extensible Parsers](#) (CIDR 2025)

[Robust External Hash Aggregation in the Solid State Age](#) (ICDE 2024)

[These Rows Are Made for Sorting and That's Just What We'll Do](#) (ICDE 2023)

[Join Order Optimization with \(Almost\) No Statistics](#) (Master thesis, 2022)

[DuckDB-Wasm: Fast Analytical Processing for the Web](#) (VLDB 2022 Demo)

[Data Management for Data Science - Towards Embedded Analytics](#) (CIDR 2020)

[DuckDB: an Embeddable Analytical Database](#) (SIGMOD 2019 Demo)

Reference

<https://alirezasadeghi1.medium.com/duckdb-beyond-the-hype-8b1e59360cf3>

<https://mihaibojin.medium.com/duckdb-the-big-data-rising-star-71916f953f18>

<https://medium.com/radiant-earth-insights/duckdb-the-indispensable-geospatial-tool-you-didnt-know-you-were-missing-5fe11c5633e5>