

# DuckLake

## Rethinking Lakehouse Architectures

EuroPython 2026 – Kraków, Poland

15.07.2026

Marc-André Lemburg :: eGenix.com GmbH

# Speaker Introduction

## Marc-André Lemburg

- Studied Mathematics, Computer Science and Physics
- CEO eGenix.com GmbH
- **Senior Solution Architect, Consulting CTO and Coach**
- **EuroPython Society Fellow** and former Chair
- **Python Software Foundation Fellow** and former director
- **Python Core Developer** (Unicode, DB-API, platform module, ...)
- **Co-founder Python Meeting Düsseldorf**
- Based in Düsseldorf, Germany
- More details and contact: <https://malemburg.com/>



# Database

- ACID compliant data storage
- Typically **row based**
- Focus on transactions (**OLTP**)
- Persistent (or in-memory)
- **SQL** as standard query language
- Examples:
  - PostgreSQL
  - MySQL
  - Oracle / MS SQL
  - SQLite



# Data Warehouse

- Large scale, ACID compliant data storage for many different use cases
- Typically **column based**
- Focus on analytics (**OLAP**)
- **SQL** as standard query language
- Examples:
  - Snowflake
  - Google BigQuery
  - DuckDB



# Data Lake

- Data storage for huge amounts of data
- Usually file based (object stores)
- Focus on unstructured data
- Separates compute and storage
- Examples:
  - Amazon Athena
  - Apache Presto / Trino
  - Apache Iceberg
  - DuckLake

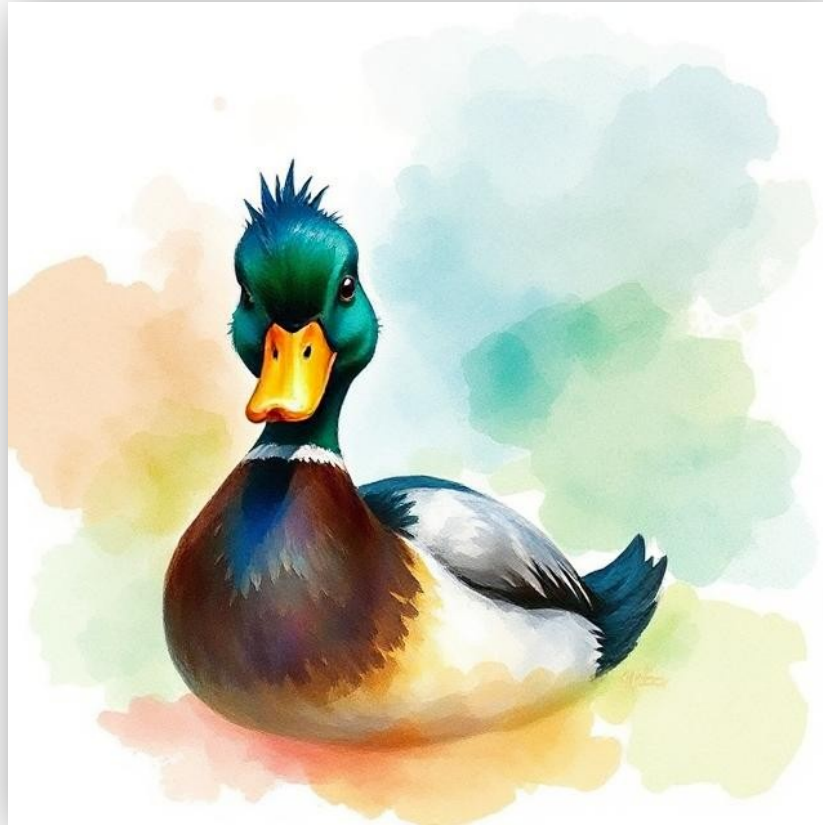


# Lake House = “Best” of Data Lake and Warehouse

- Data storage for huge amounts of data
- Diverse set of supported data types
- Focus on analytics (OLAP)
- Separates compute and storage
- SQL as standard query language
- Examples:
  - Databricks / Snowflake / etc.
  - DuckLake + DuckDB



# What do Ducks have to do with Databases ?



# DuckDB

- In-process analytics data storage (OLAP)
- Column based (Apache Arrow)
- Persistent and in-memory
- SQL as standard query language
- Single writer / multiple readers
- *Similar to SQLite, but for OLAP workloads*
- Great Python support

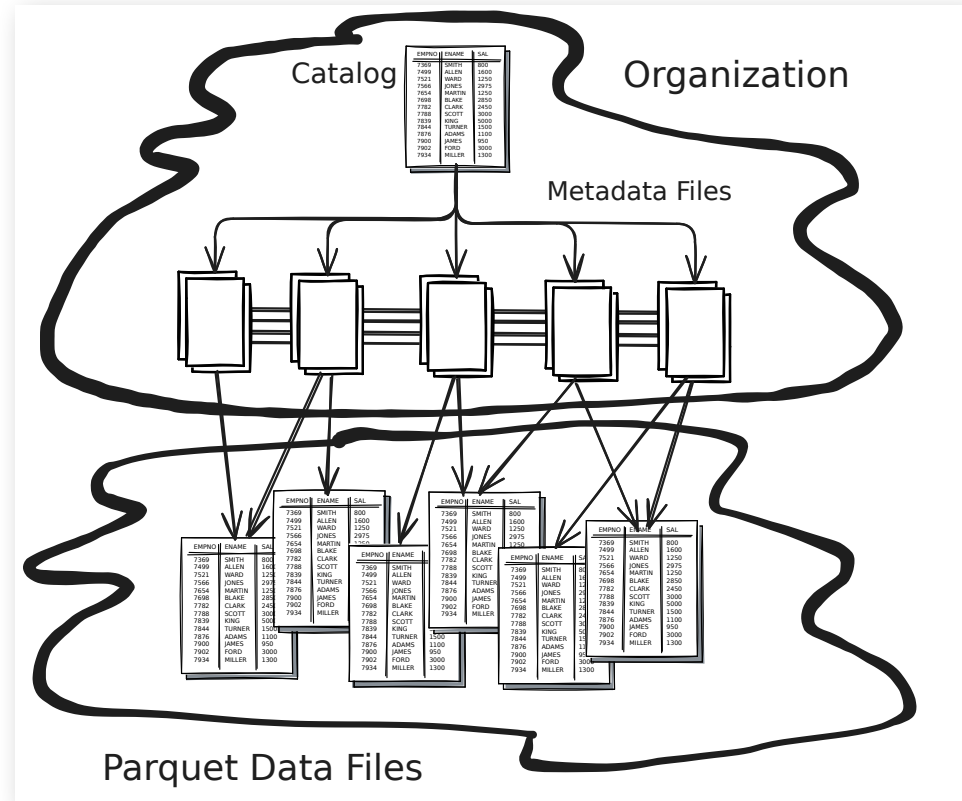


**“Ducks and Lakes” seems to be a good fit**



# Apache Iceberg / Delta Lake

- Storage part of a data lake or lake house
- OLAP table format
- Open source file formats
- Can handle petabytes of data
- (Can be) Used by
  - Snowflake
  - Databricks
  - Apache Presto / Trino



# DuckLake

- **Extension for DuckDB**
- In-process compute
- **Stores data in Parquet files** and manages these (ACID compliant)
- Storage can be an object store, local or network file system
- **SQL** as standard query language
- Great Python support

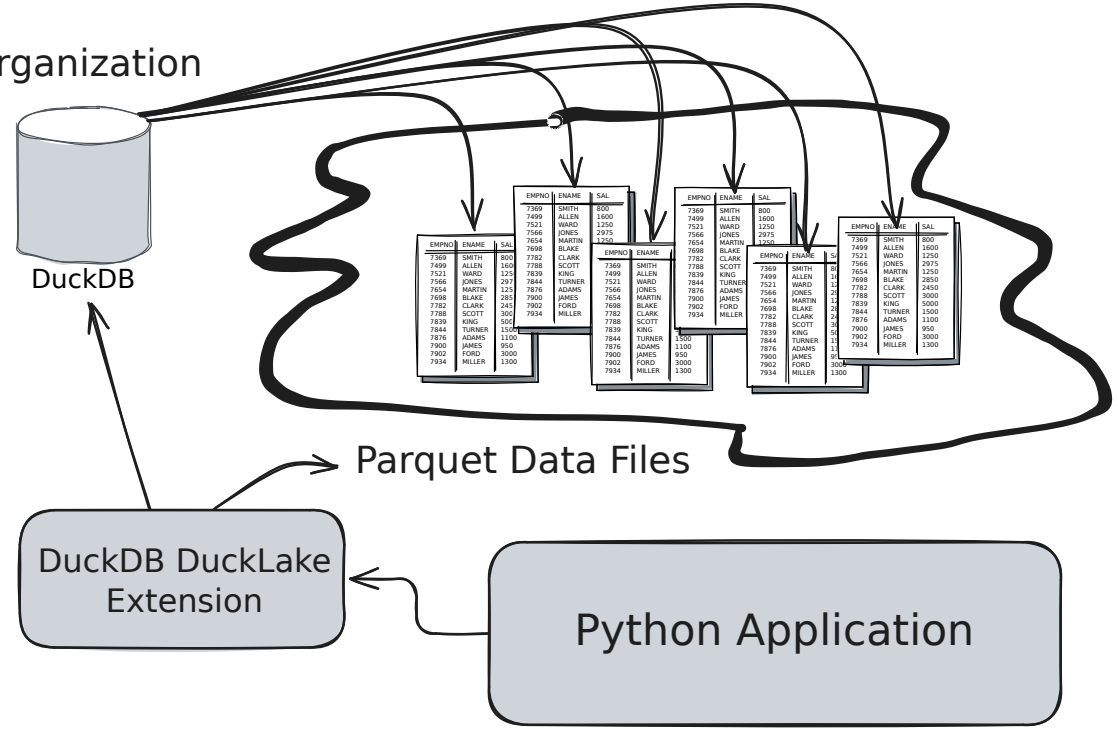


# DuckLake Architecture

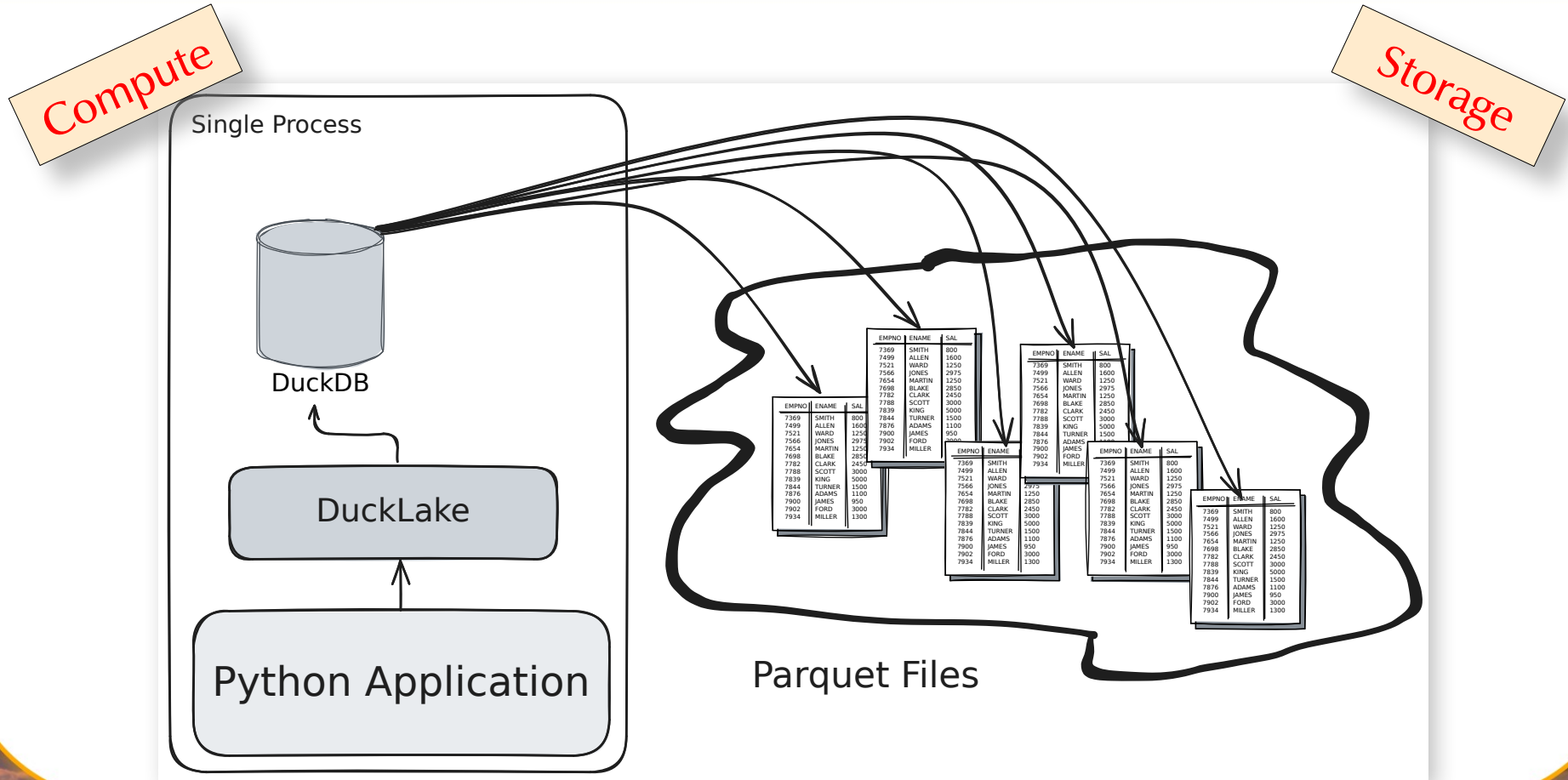
- Catalog and metadata stored in an SQL database
  - PostgreSQL
  - DuckDB
  - SQLite
- Data stored using Parquet files
- Completely open architecture

## DuckDB DuckLake

### Organization



# DuckLake as an in-process Lake House



# DuckLake Features

- **Snapshots**
  - Committed transactions
- **Schema Evolution**
  - Schema changes do not require rewriting data
- **Time Travel**
  - Access any version of the data lake
- **Persistent Configuration**
  - Configuration stored in the catalog



# DuckLake More Features

- **Multiple Writer and Readers**
  - with conflict resolution
- **Compression and Encryption**
  - via support in Parquet files
- **Partitioning**
  - can be changed without rewriting data
- **Importing existing Parquet files**
  - Using shallow copy
  - Iceberg compatibility



# DuckLake 1.0 – Production Ready Lake House

- **Variant types**
  - Semi structured data
  - Think JSON, but typed, stored in binary and with optimized queries
- **Geometry types**
  - for spatial data
  - Query based on geometry
- **Inlining data**
  - Store small updates directly in the catalog
  - CHECKPOINT to flush data into storage



# DuckLake 1.0 – More features

- **Sorted tables**
  - DuckLake keeps the data sorted for you
  - Improves queries that rely on sorting
- **Bucket partitioning**
  - Partition based on hashed values
- **Deletion vectors**
  - Mark data as deleted
  - Postpone the data rewrite to later compaction



# DuckLake Roadmap

- **Git like branching**
  - For data rather than code
- **Materialized views**
  - With incremental refresh
- **Quack protocol support**
  - Use DuckDB clusters via Quack
- **Variant inlining**
  - Already partially supported (DuckDB)
- **Permissions and roles**
  - Already partially supported (Postgres)



# DuckLake in Action

- Task: Load **German weather data** (DWD)
  - Daily measurements for all measurement points
  - Some going back to 1781
  - 2.4GB raw CSV data, 1.6GB cleaned up
- **DuckDB**: 435MB
- **DuckLake**: 242MB
  - 4.1MB metadata DuckDB
  - 238MB Parquet files



# Conclusion

- DuckLake takes a **fresh approach to the lake house idea**
  - No complex metadata file setups as in Apache Iceberg
- **Great analytics and ETL support via DuckDB**
  - Fast thanks to Apache Arrow
- **Production ready**

*No ducks were harmed in the making of this presentation.*



**Main takeaway:** Never stop **learning** and **trying out new things**



These  
are  
exciting  
times

**Thank you for your attention !**



Time for discussion

# Contact

**eGenix.com Software, Skills and Services GmbH**

Marc-André Lemburg

Pastor-Löh-Str. 48

D-40764 Langenfeld

Germany

eMail: [mal@egenix.com](mailto:mal@egenix.com)

Phone: +49 211 9304112

Fax: +49 211 3005250

Web: <https://www.egenix.com/>

LinkedIn:



# References

- Several photos taken from Pixabay and Unsplash
- Many images were generated using DeepAI text2img
- Some screenshots taken from the mentioned websites
- All other graphics and photos are (c) eGenix.com or used with permission
- Details are available on request
- Logos are trademarks of their respective trademark holders