# Product Vision

Tobias Schoofs

April 20, 2023

# NoWDB

$$\text{NoWDB} = (graph + timeseries + \text{SQL} + Lua) \times Python$$

- A data processing platform
- For large amounts of *timeserish* data
- That can be used with standard interfaces (SQL)
- Emphasising data science on client side
- Emphasising big-data on server side

# Timeseries (1/2)

- Growing amount of data are timeseries *to some extent*
- Important driver: IoT
- But also: "traditional" industries:
  - IT infrastructure providers,
  - Energy,
  - Manufacturing ("Industry 4.0"),
  - Retail,
  - Fintech,
  - ...
- Timeseries data have special requirements:
  - Fast ingestion ($> 1M/s$)
  - Scalability of queries ($> 1Bmetrics$)
  - Special handling of time dimension
  - Timeseries are less rigid compared to relational data (time points may be lost or duplicated)

# Timeseries (2/2)

- Avalailable timeseries databases are too narrow, *i.e. pure* timeseries.
- Examples:
  - InfluxDB
  - OpenTSDB
- There are projects trying to solve this issue, *e.g.*:

$$timescale = relational + timeseries$$

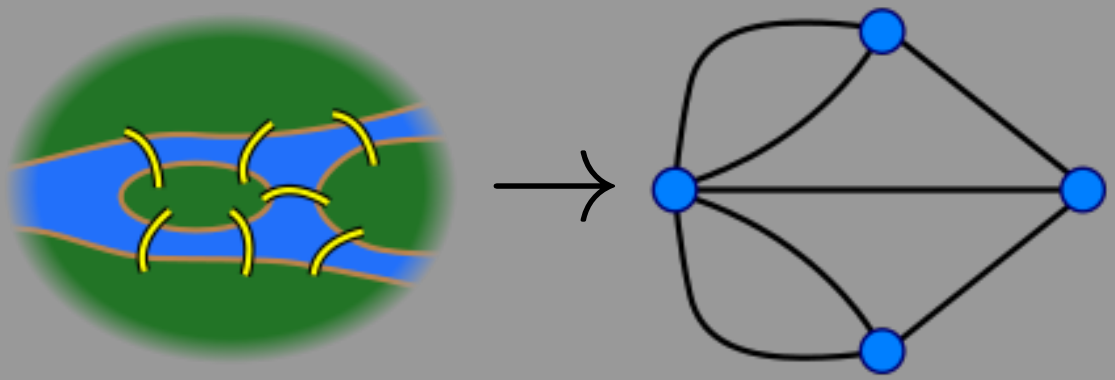
... but it's still Postgres: somewhat slow



# Graph

- Make timeseries more widely applicable!
- Natural candidates: Relational and Graph
- Graph is more flexible:
  we can add new edges
  without changing the structure of entities
- Graph is less rigid:
  duplicated edges are no issue

The Founder of Graph Theory:
Leonhard Euler



Provide full power of data-science tools on client side:

- NumPy
- SciPy
- Pandas
- Matplotlib
- ...

Provide ready-made domain-specific packages:

- Statistics
- Timeseries
- Bayes, Support Vector Machines, Clustering, *etc.*
- Geodata
- Retail (*e.g.* recommendation engine)

SQL + 


High level of integration client & server:

- Server-side support for domain-specific packages,
  *i.e.* support in SQL and server-side scripting (Lua & Python)
- Stream Processing
- Monitoring
- Integration with messaging tools (*e.g.* Kafka)
- External Interfaces:
  - native (Python, C, Go, *etc.*)
  - REST (for, *e.g.*, Grafana)
  - HTML (through, *e.g.*, Go)
  - ODBC/JDBC

# Strategy

- **Complement** existing OLTP infrastructures
- **Compete** with complex big-data and data science platforms
- Provide a **low-cost** alternative to frameworks with high cost of ownership (*e.g.*: Hadoop, large OLAP systems, *etc.*)
- Target industries with **timeserish** data and need of data science (*e.g.* IoT, retail, fintech, "Industry 4.0")
- Provide pre-packaged out-of-the-box solutions (*e.g.* integration with Kafka, Zookeeper, SPARK & Grafana)
- Operate with a mix of open-source and proprietary licenses (*e.g.* generic modules for free, specific modules for pay)