

# Volatile data lakes in the cloud

## feat. Apache Iceberg

---

Pawel Adaszewski @ JUG Saxony

14 November 2024

---

# About me



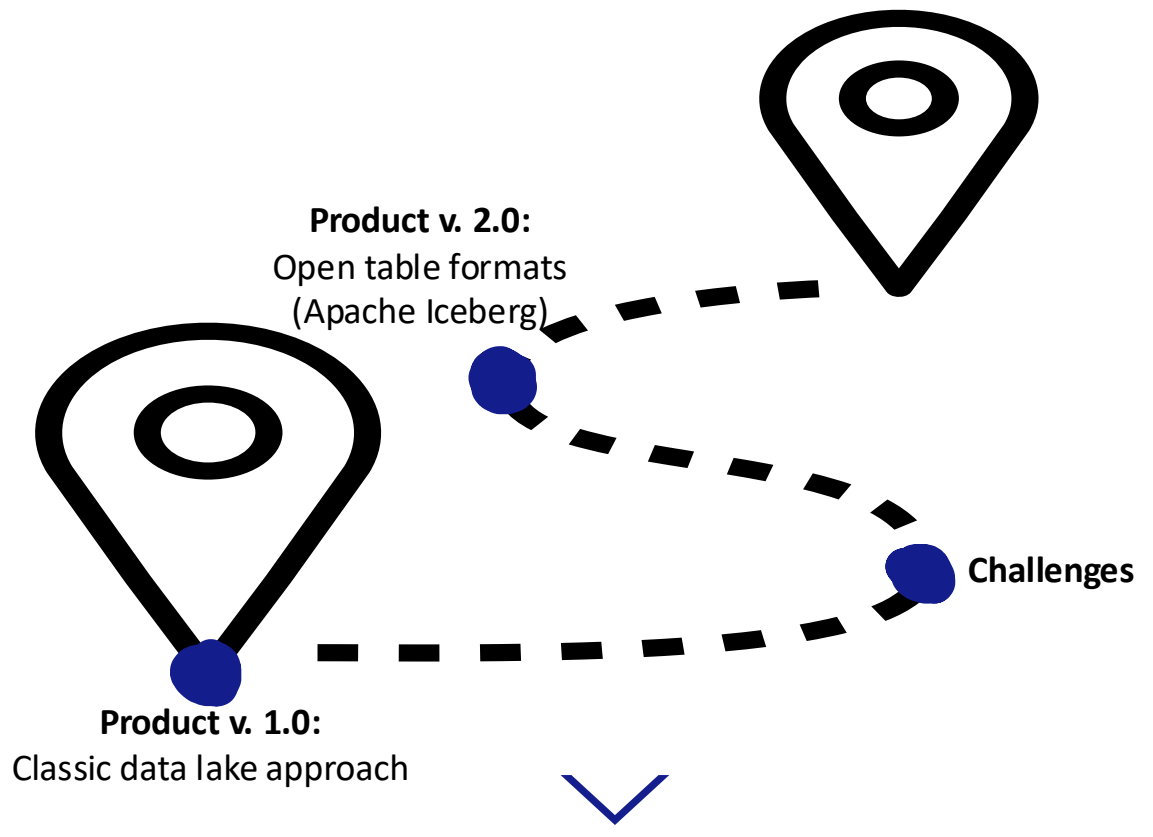
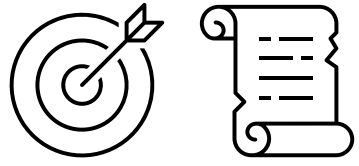
**Pawel Adaszewski**

**Cloud Architect @ ZEISS Digital Innovation**

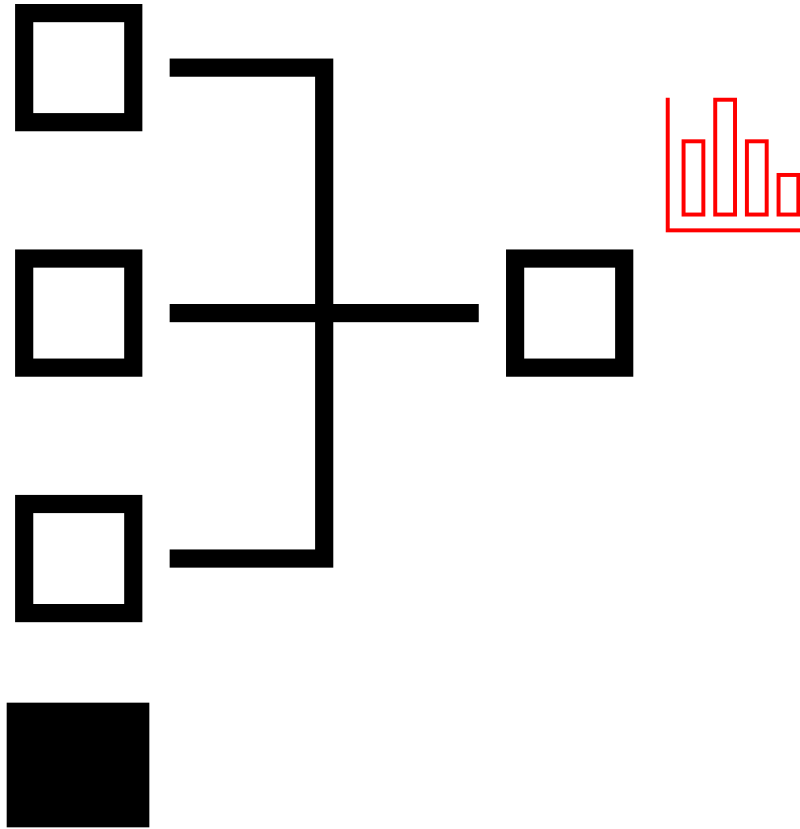
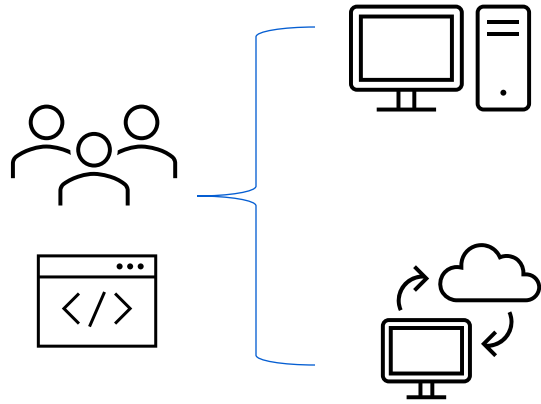
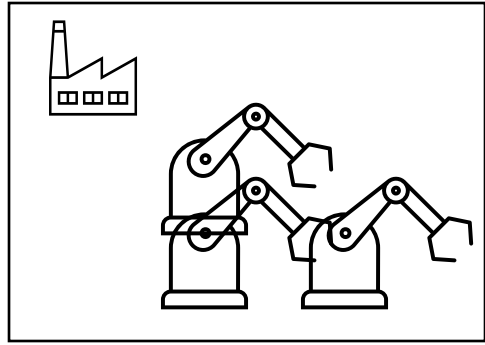
*Background (AWS):*

- Development
- Architecture
- Domain: Automotive, Manufacturing, IIoT
  - ➔ Big data collection, evaluation & analysis

***[linkedin.com/in/pawel-adaszewski](https://www.linkedin.com/in/pawel-adaszewski)***



Basic introduction and inspiration to adopt open table formats based on large manufacturing reporting project learnings

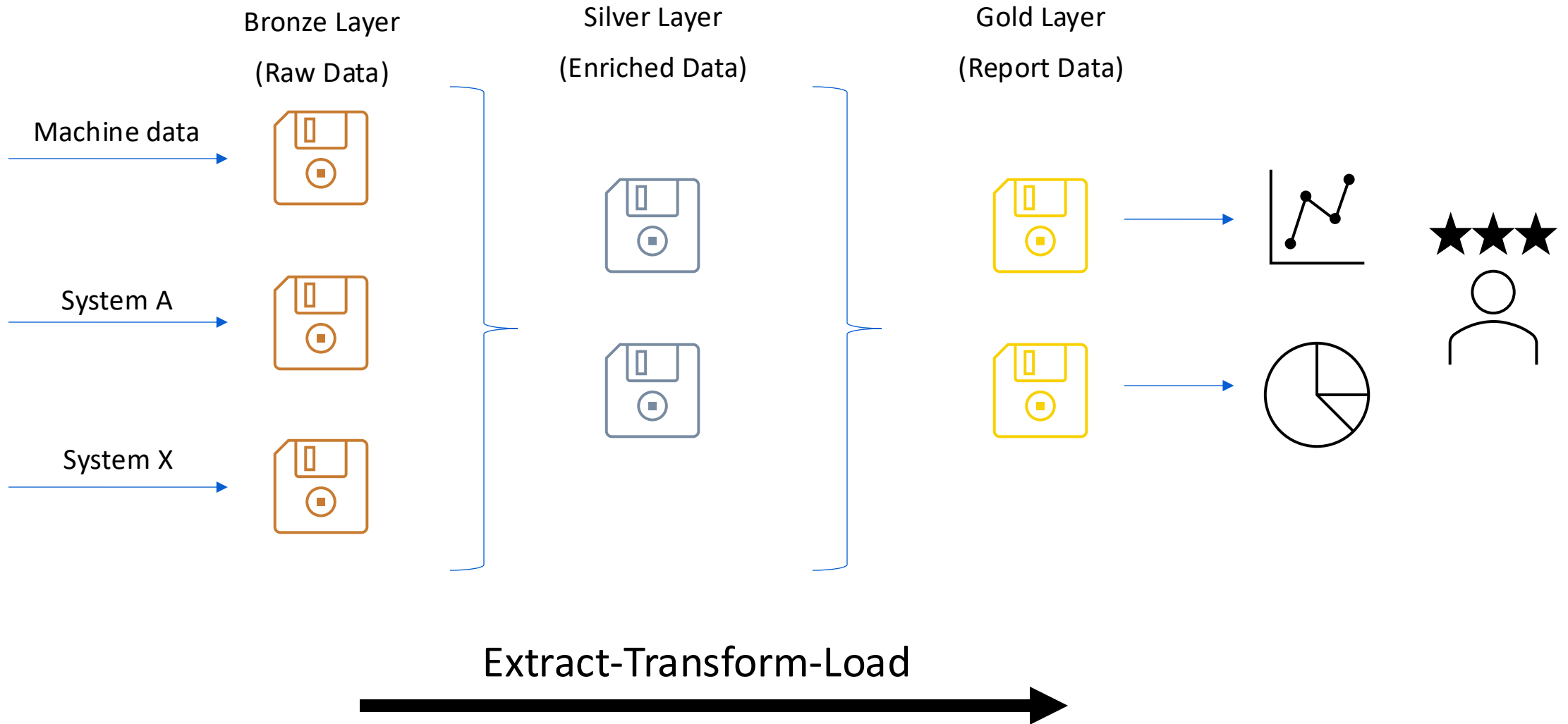


# Data Warehouse vs. Data Lake

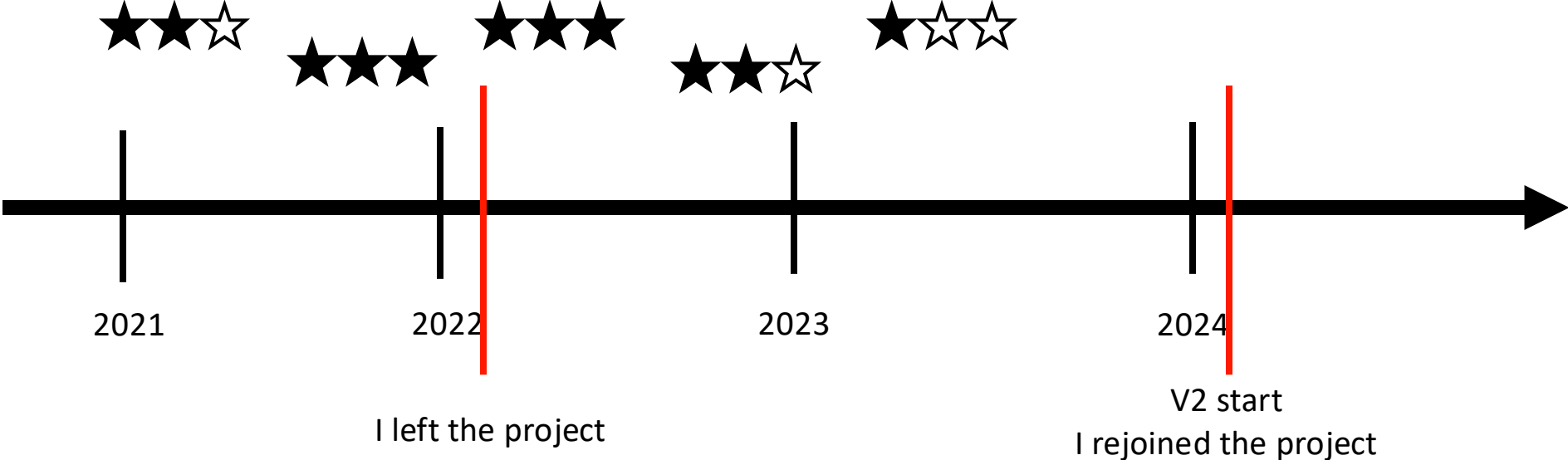




# Process



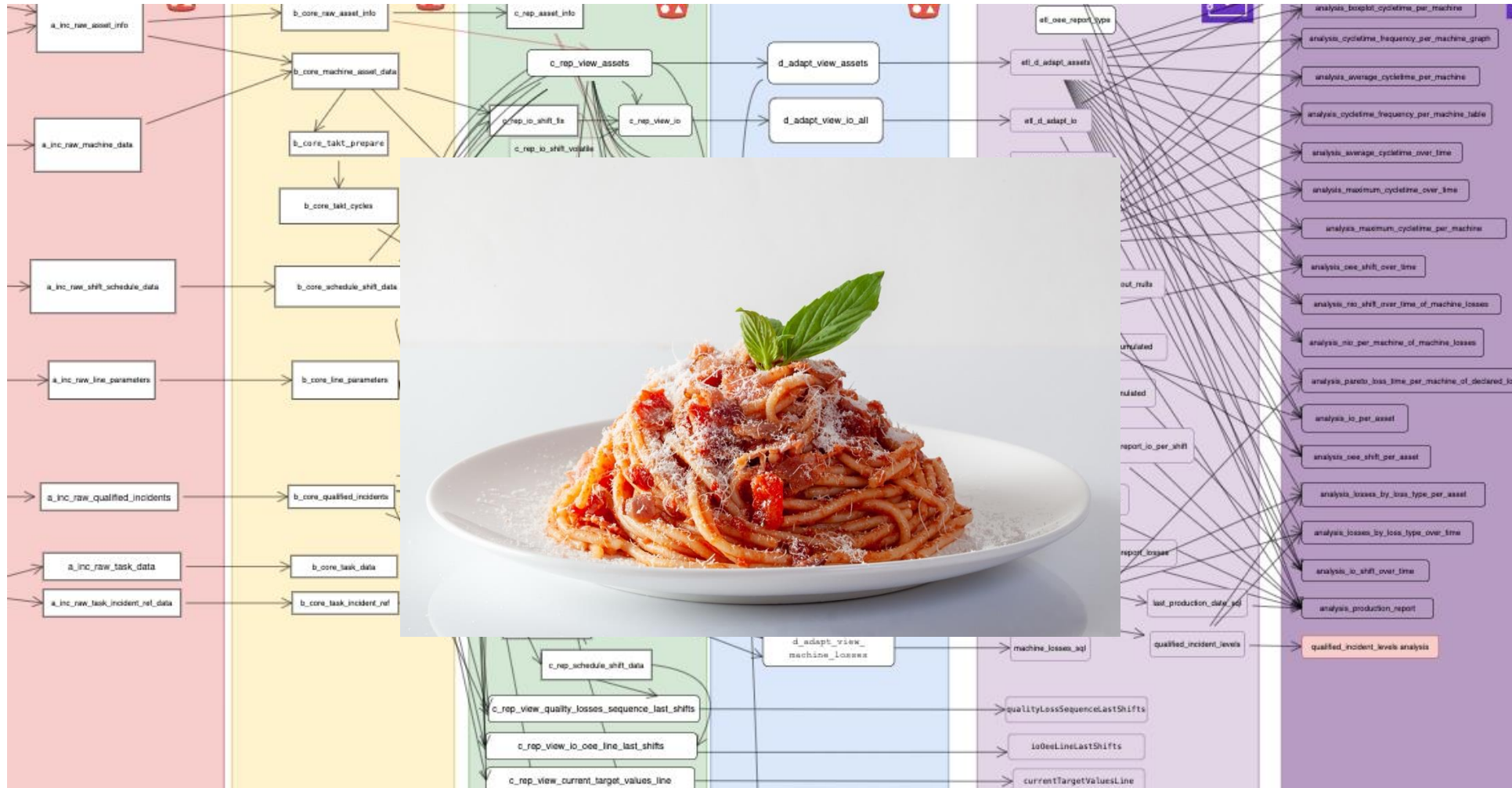
# Customer satisfaction timeline



# Why (have we had to move from v1 to v2)?

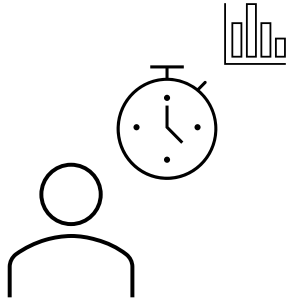
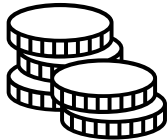


Extract-Transform-Load (ETL):

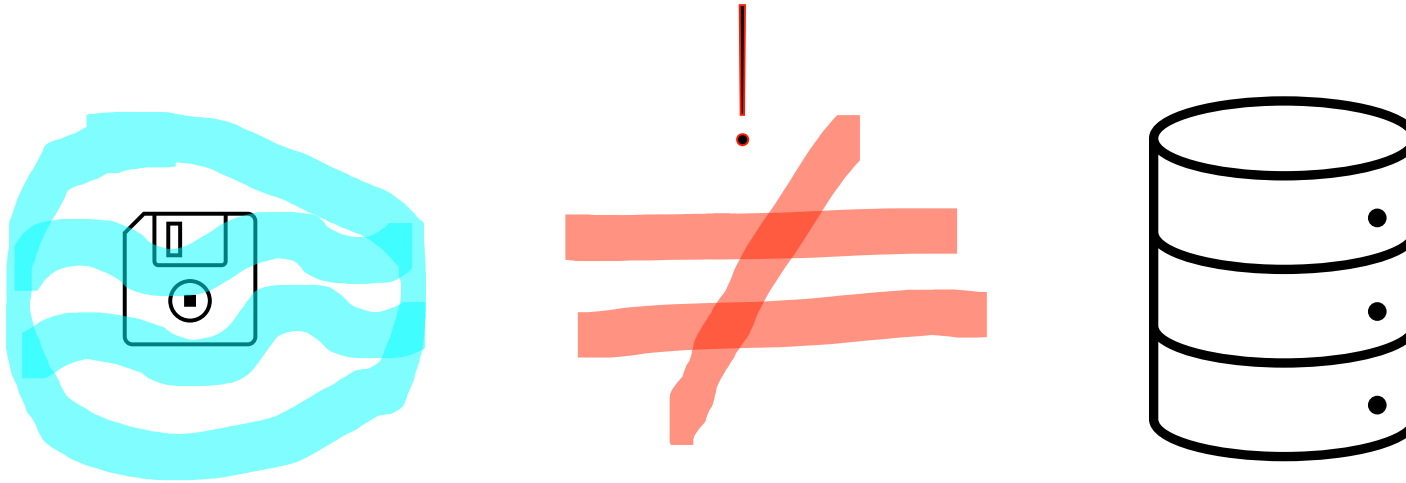




# Consequences?



# Why?

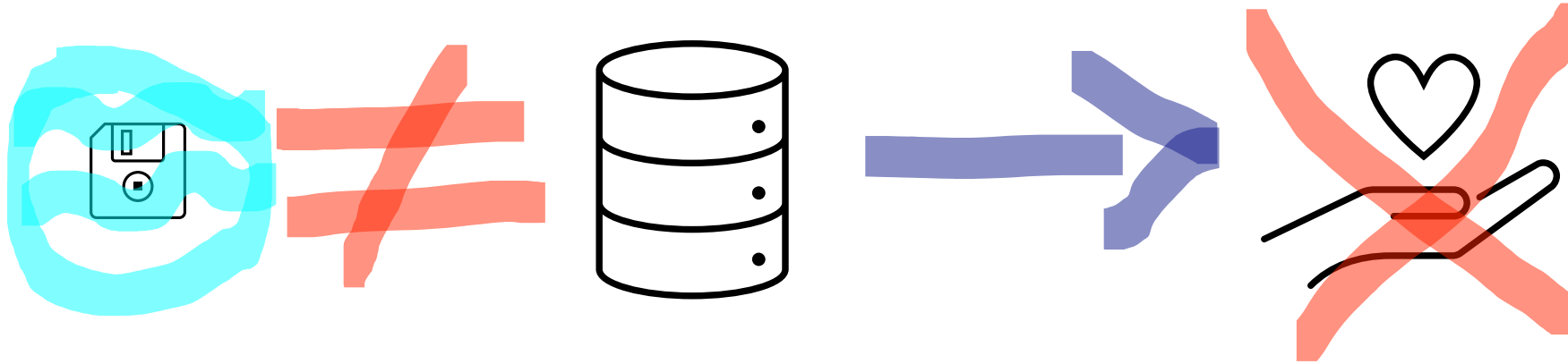


No support for transactions (ACID: atomicity, consistency, isolation, durability)



Append-only by initial design!  
(analytical use-cases)

So?



- ~~X~~ Record level updates or deletes
- ~~X~~ Managing query performance when the table grows
- ~~X~~ Transactions → Data corruption for partway failed operations

**How to effectively handle volatile data in our queries?  
(No complex joins with windowing functions and full table scans)**

# Open table formats for the rescue!



Data lake? Data warehouse?



**Data lakehouse!**



**DELTA LAKE**

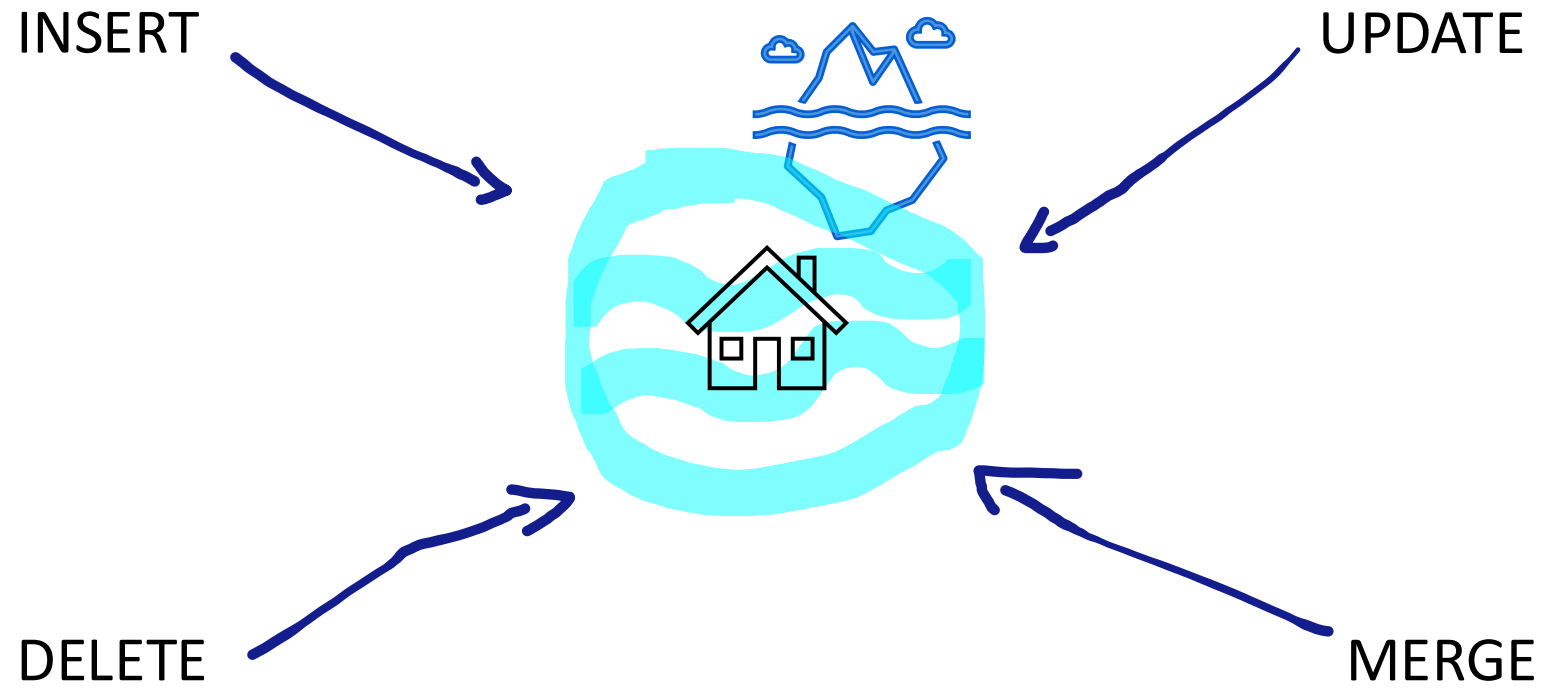


**ACID**

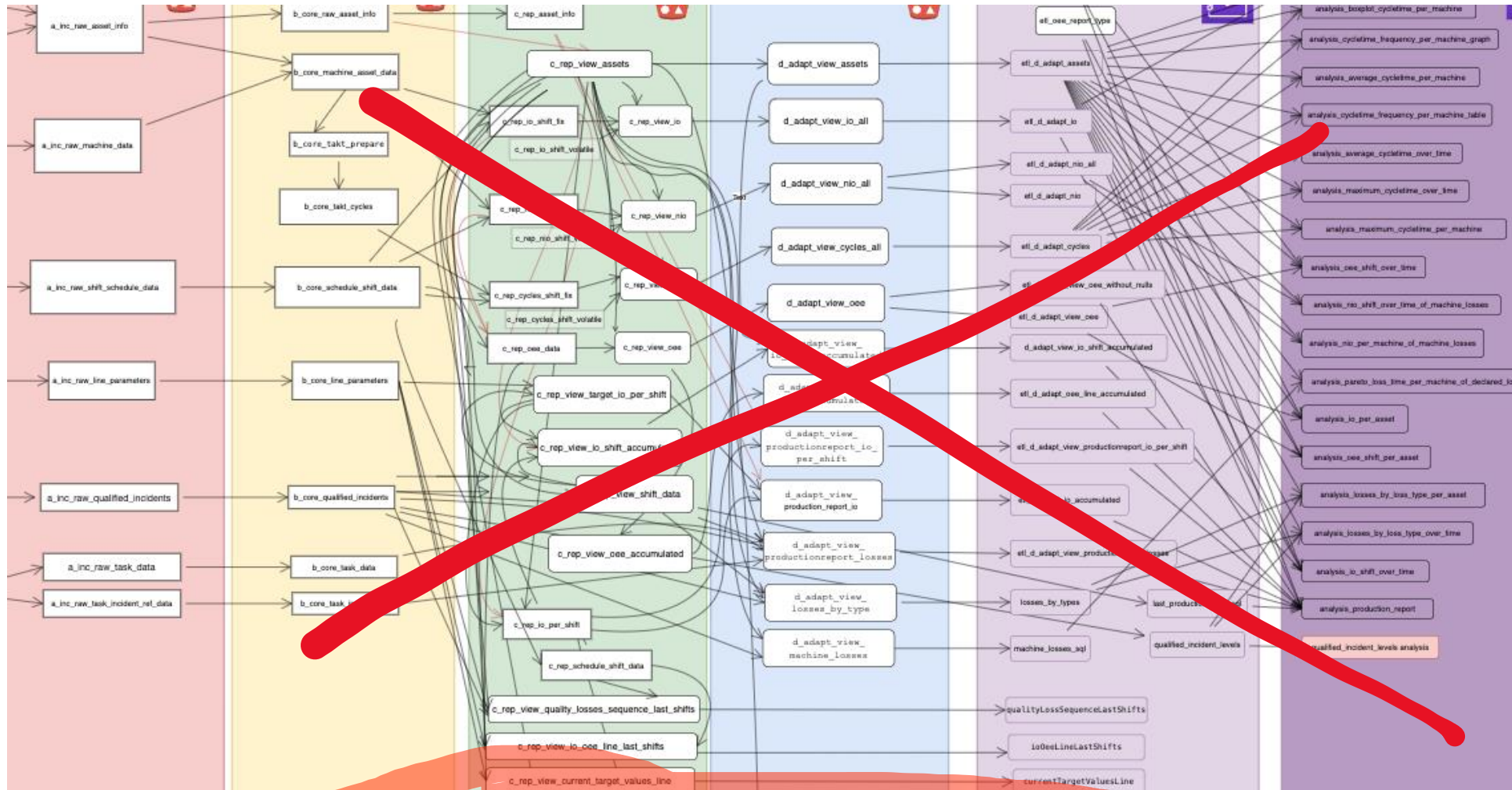
Apache Iceberg is a table format that allows groups of **Parquet files in a data lake to be recognized as database tables**. These tables can be easily queried using SQL with **various engines** or loaded into popular Python dataframe libraries such as Polars, Pandas, Ibis, or SQLFrame. This technology supports an architecture known as the Data Lakehouse, which allows you to use all your favorite tools and platforms against a **single copy of your data**. This approach **eliminates the need to create multiple copies of your data** across several tools, reducing costs, consistency issues, and other headaches.

One of the **challenges** that has held back the widespread adoption of Data Lakehouses is the **fear of “picking the wrong format”** among the three main formats: Apache Iceberg, Apache Hudi, and Delta Lake. However, several recent events have solidified **Apache Iceberg as the industry standard** that enterprises can confidently adopt. In this blog, I will discuss some of these significant events to illustrate Apache Iceberg’s current momentum.

What does this mean?



So?



NO NEED FOR COMPLEX ETLs

Wait... how is this possible with append-only mechanism of a data lake?

### DELETE SCENARIO

Metadata: SNAPSHOT 1

ID	Name
1	Nick
2	John

File: 01.parquet



Read file, change row,  
save new file

Metadata: SNAPSHOT 2

ID	Name
1	Nick

File: 02.parquet

### UPDATE SCENARIO

Metadata: SNAPSHOT 1

ID	Name
1	Nick
2	John

File: 01.parquet



Read file, change row,  
save new file

Metadata: SNAPSHOT 2

ID	Name
1	Nick
2	Johny

File: 02.parquet

# Copy-on-Write (CoW)



# CoW too slow for write heavy applications?

## DELETE/UPDATE SCENARIO

Metadata: SNAPSHOT 1

ID	Name
1	Nick
2	John

File: 01.parquet



Metadata: SNAPSHOT 2

ID	Name
1	Nick
2	John

File: 01.parquet

Output:

ID	Name
1	Nicky

file_path	position
01.parquet	2

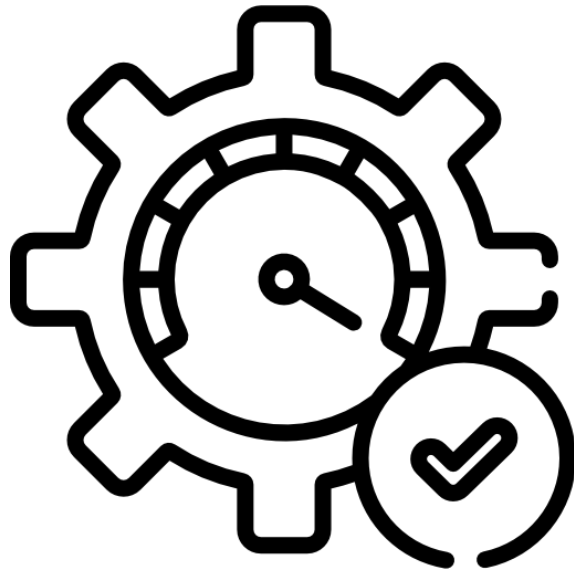
File: delete01.parquet

ID	Name
1	Nicky

File: 02.parquet

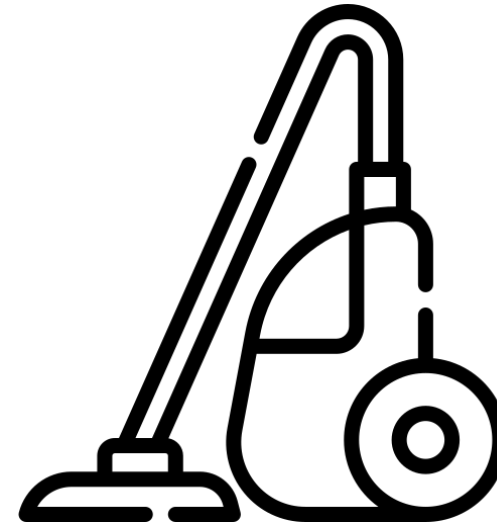
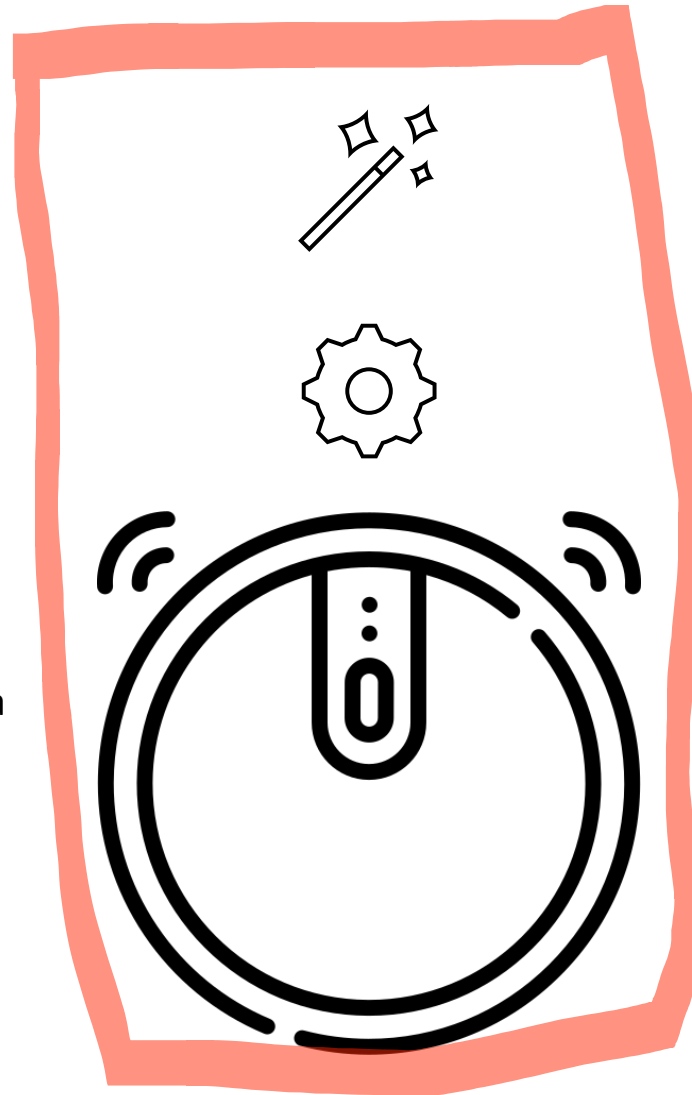
# Merge-on-Read (MoR)

# Can it be further optimized?



OPTIMIZE:

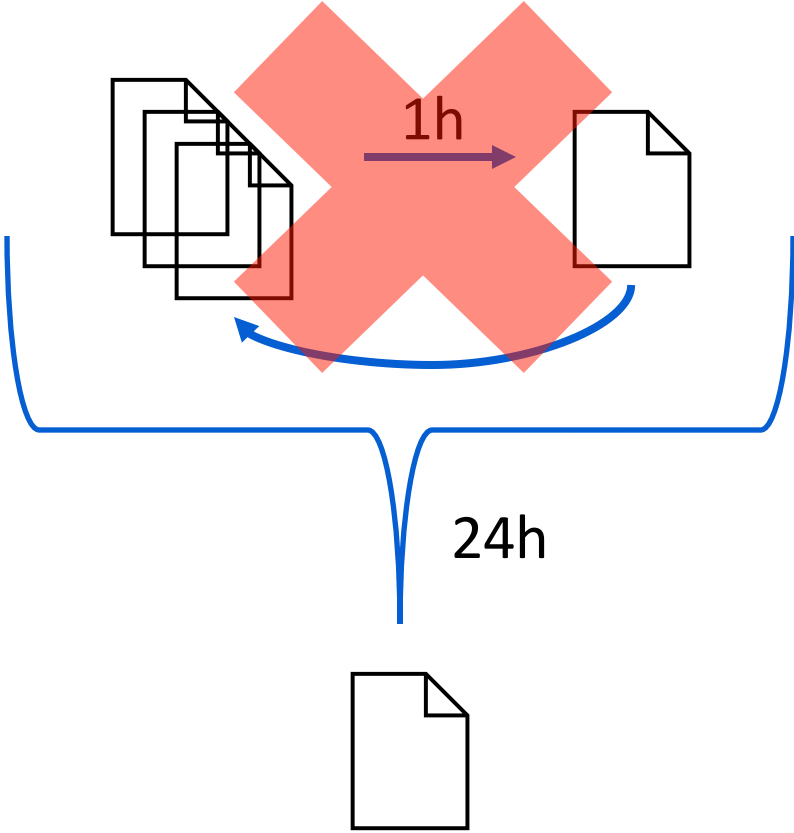
→ Rewrite common data  
(compaction)



VACUUM:

→ Delete unnecessary data

01.01.2021:



# Iceberg + JVM?



- ✓ Implemented in Java → Provides Java API and JDBC integration
  
- ✓ Supported by leading processing engines:
  - Apache Spark
  - Apache Flink
  
- ✓ Integrated into modern serverless tooling:
  - can be implemented with JVM languages

## Other cool goodies



- ✓ Time travel (when you don't vacuum the data 😊 )
- ✓ Easy schema evolution (new schema in a new snapshot)
- ✓ Mechanisms for concurrent writers

# Is it now a match for us? – original answer from June 2024



It depends

Ask me in 6 months

Hard to say

Yea, maybe

We will see

# Is it now a match for us? – August 2024



Yes, but...

AWS needs to  
improve their  
serverless tools for  
handling Apache  
Iceberg tables : )

Cloud provider does not matter!

# Is it now a match for us? – November 2024



Yes, but...

We need to roll out  
the application at  
scale to verify this

AWS quotas still problematic  
(integration with other services)





Seeing beyond