



ACID ORC, Iceberg and Delta Lake

an overview of table formats
for large scale storage and analytics

Michal Gancarski
michal.gancarski@zalando.de



17-10-2019





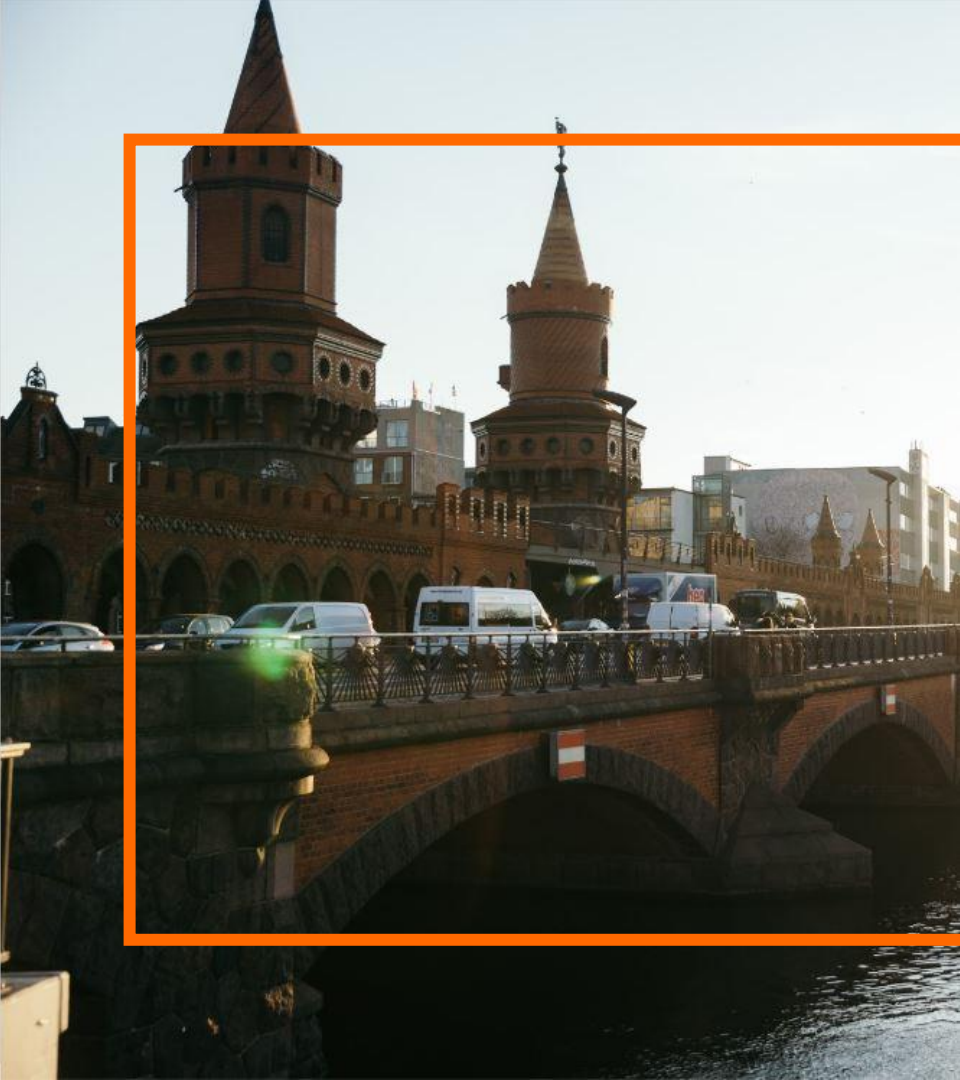
TABLE OF CONTENTS

All Is Not Well In The Land Of Big Data

There Is Hope, However

This Is How We Do It

Moving Forward



All Is Not Well In The Land Of Big Data

ACID Properties

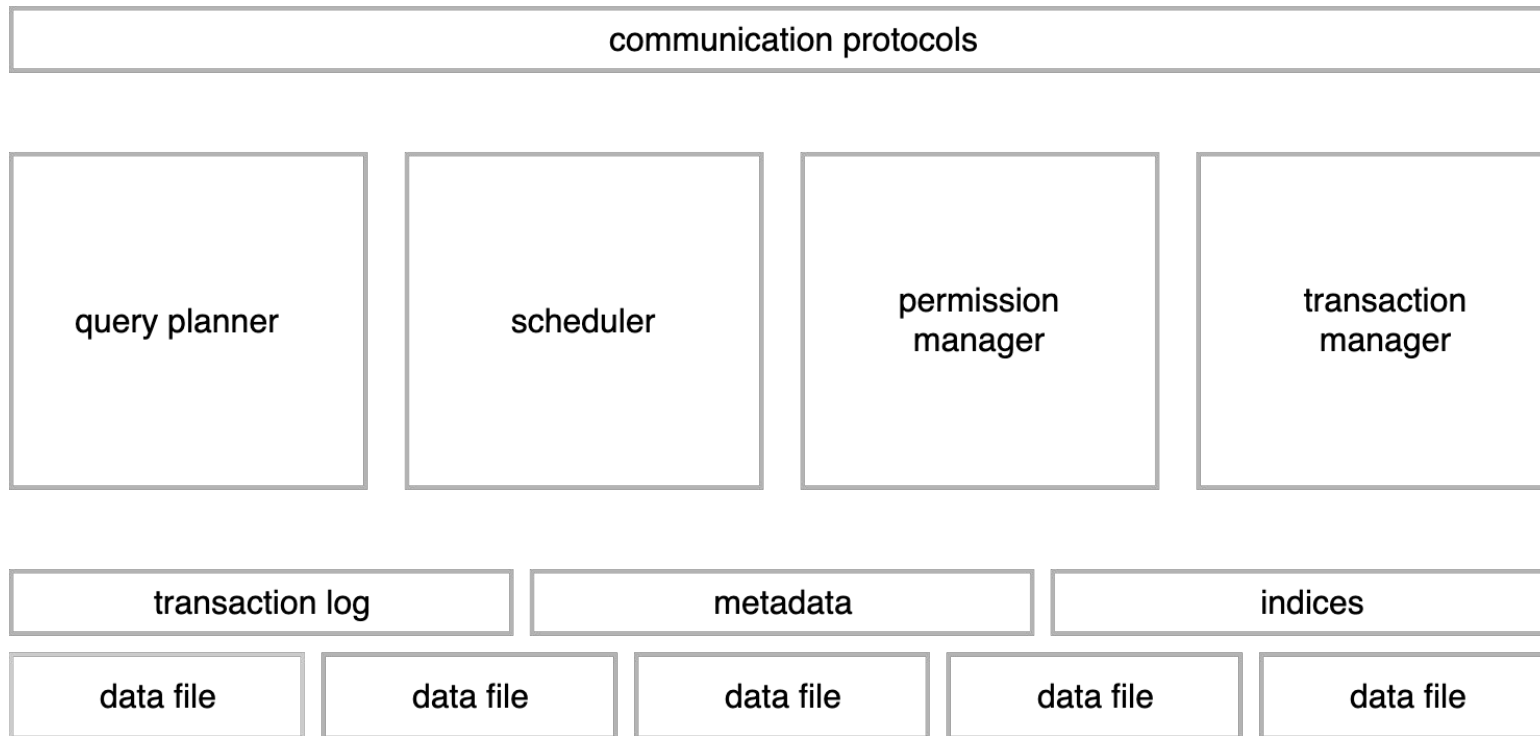
1
A
Atomic

2
C
Consistent

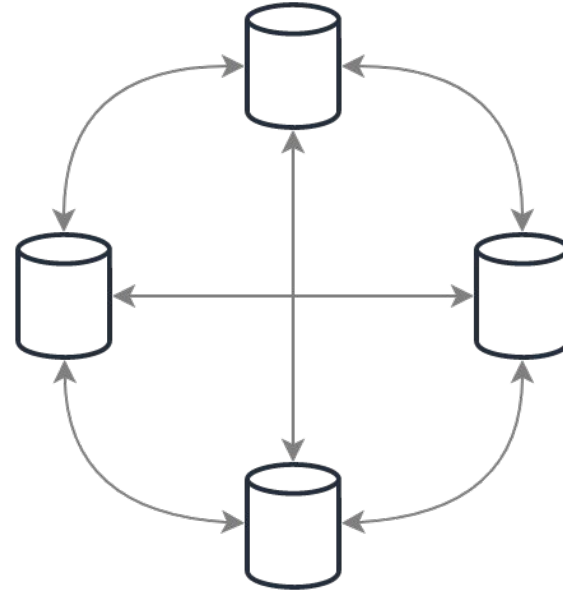
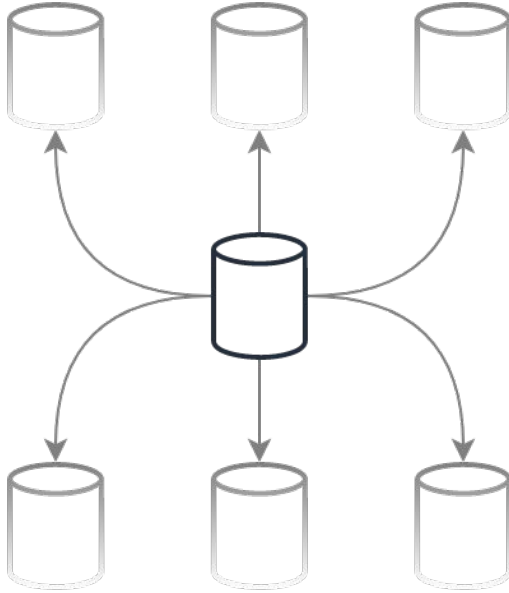
3
I
Isolated

4
D
Durable

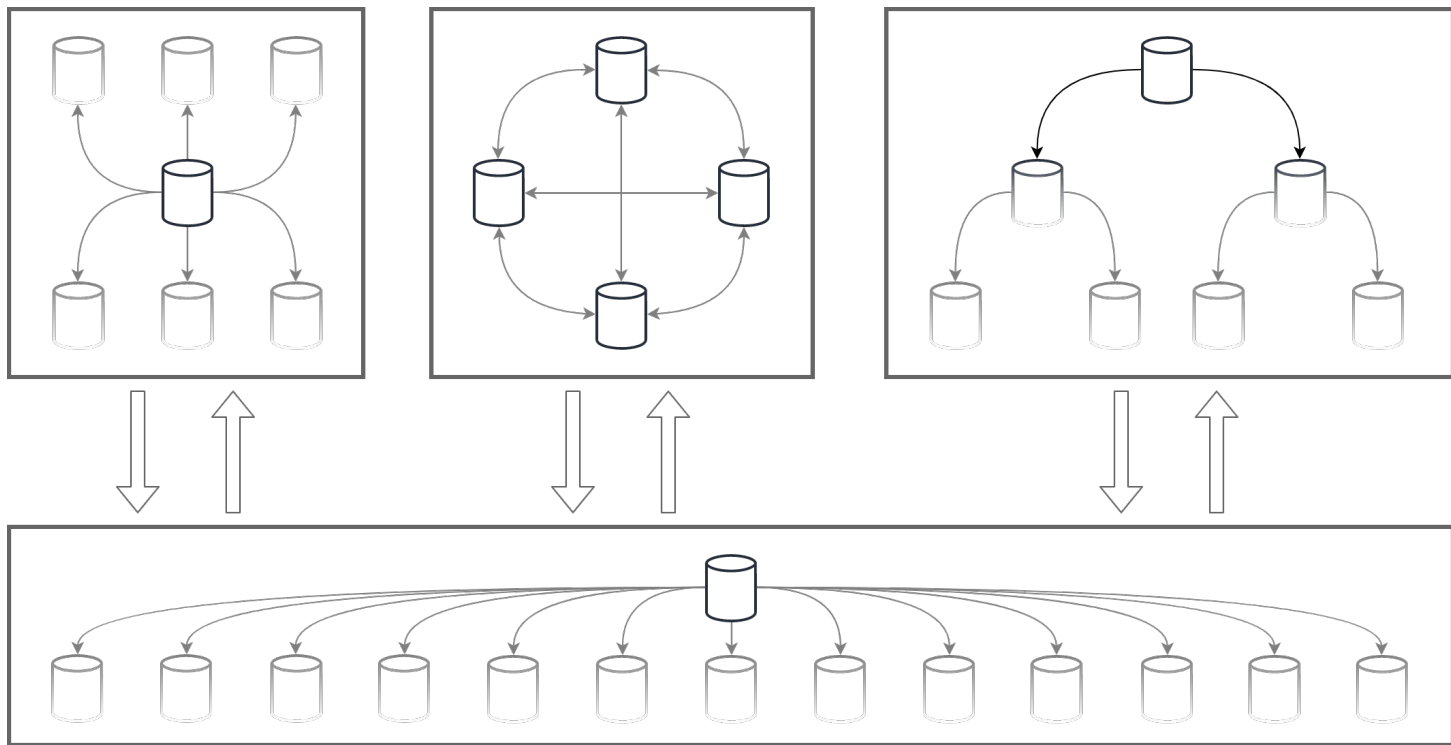
Single Node Database



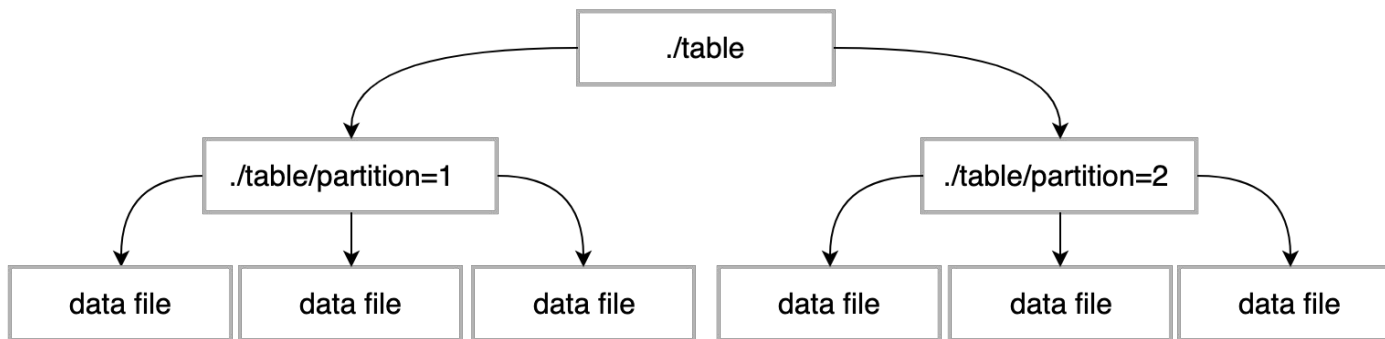
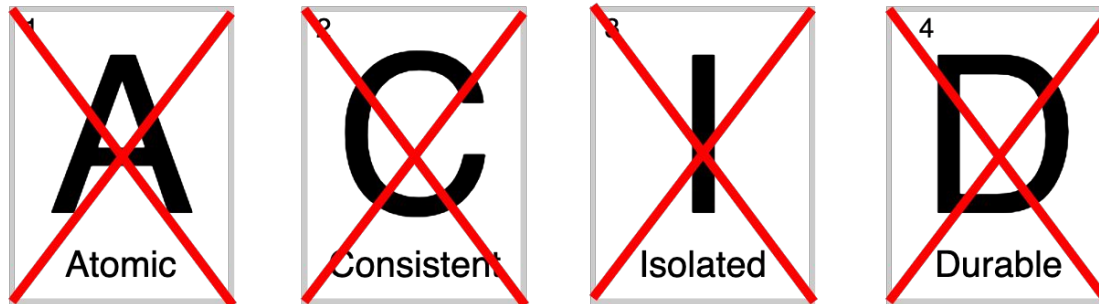
Distributed Database



Distributed Data Infrastructure



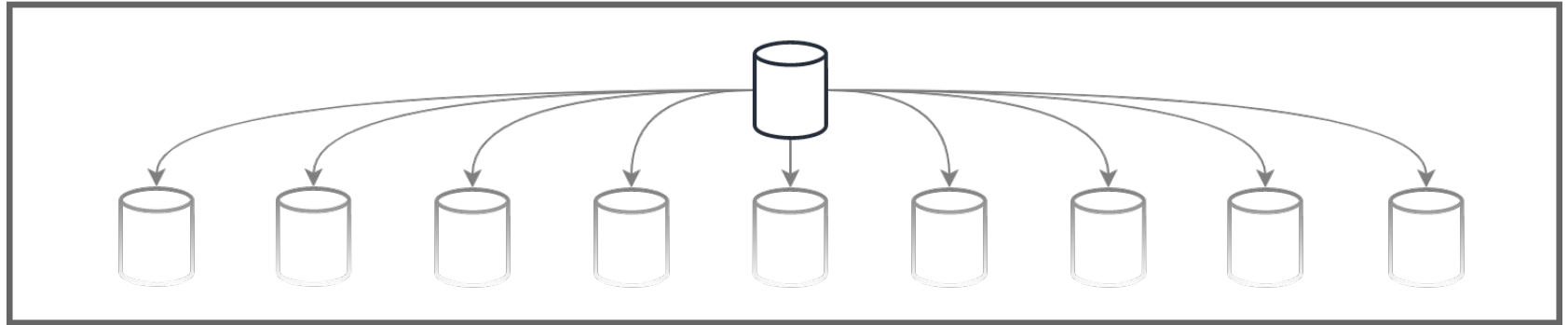
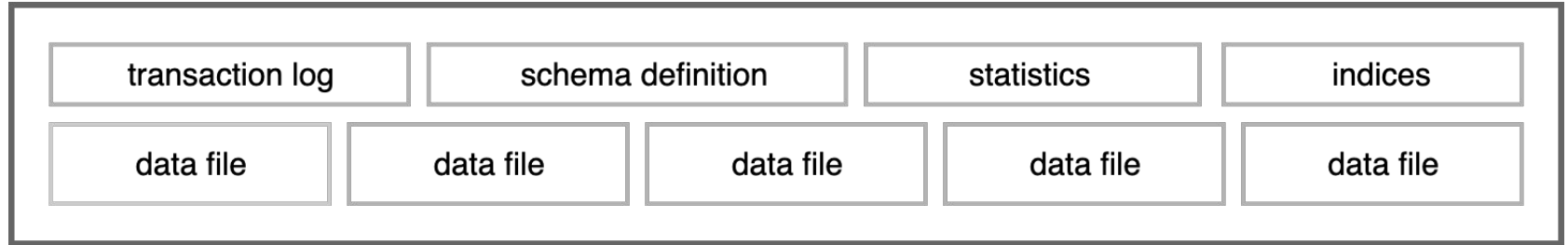
Lost ACID





There Is Hope, However

A Table Format?



ACID ORC



ACID ORC

```
CREATE TABLE d_manufacturers (id int, name string)
PARTITIONED BY (country string)
STORED AS ORC
TBLPROPERTIES ('transactional'='true');
```



```
./d_manufacturers/country=de/base_00000002/
  -- bucket_00000
  -- bucket_00001
./d_manufacturers/country=de/delta_00000003_00000003_0000/
  -- bucket_00000
  -- bucket_00001
./d_manufacturers/country=de/delta_00000004_00000004_0000/
  -- bucket_00001
./d_manufacturers/country=de/delete_delta_00000004_00000004_0000/
  -- bucket_00001
```

ACID ORC



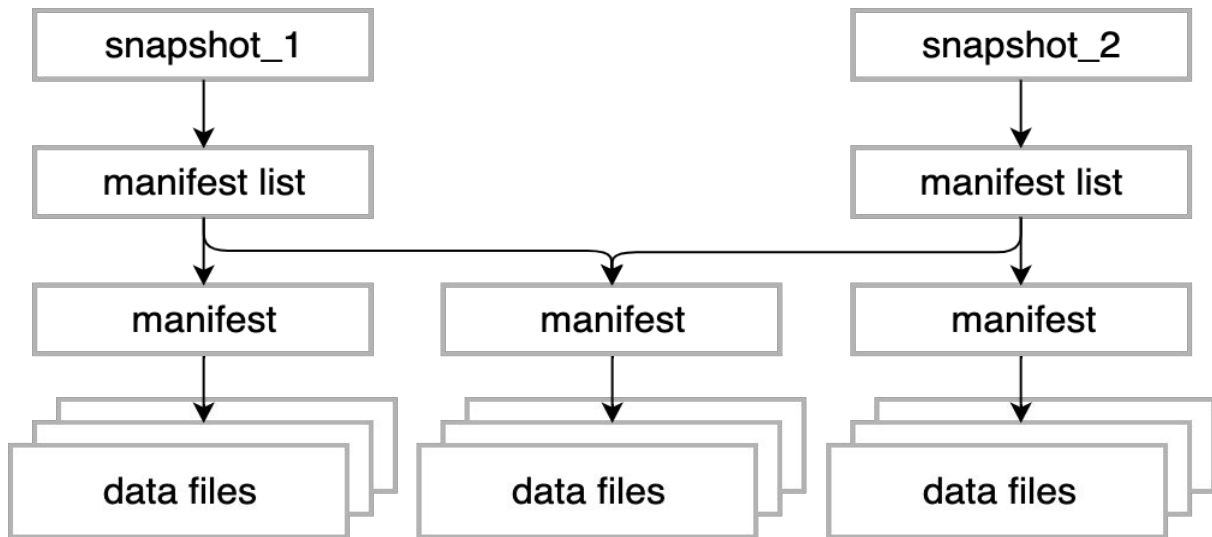
- ❖ Native compatibility with Hive
- ❖ Fast updates / upserts (no file rewrite)
- ❖ Hive 2.x ACID ORC tables can be converted to Hive 3.x ACID ORC tables
- ❖ Commercial Support (Cloudera)
- ❖ Limited support for Spark (being worked on by Qubole)
- ❖ Slow listing and metadata discovery
- ❖ Potentially slower read due to ad-hoc compaction
- ❖ ORC only
- ❖ Mandatory S3Guard or EMR with consistent view enabled

Apache Iceberg



Apache Iceberg

```
val df = spark.read  
  .format("iceberg")  
  .load("s3://datalake/d_manufacturers")
```



Apache Iceberg



- ❖ Parquet, Avro, ORC supported as file formats
 - ❖ Robust schema and partitioning changes
 - ❖ Fast query planning
 - ❖ Presto connector
 - ❖ Time travel with snapshot id listing
 - ❖ No dependency on Spark
- ```
public List<Snapshot> snapshots() {
 return snapshots;
}
```



- ❖ Spark support
- ❖ Sparse documentation
- ❖ No commercial support
- ❖ Not as mature as other formats



## Delta Lake



# Delta Lake


```
val df = spark.read
 .format("delta")
 .load("s3://datalake/d_manufacturers")
CONVERT TO DELTA parquet.`s3://datalake/d_manufacturers`
```



```
./d_manufacturers/_delta_log/
-- 000000.json
-- ...
-- 000010.checkpoint.parquet
-- _latest_checkpoint
./d_manufacturers/country=de/
-- file_1.parquet
-- file_2.parquet
./d_manufacturers/country=fr/
-- file_3.parquet
```

# Delta Lake

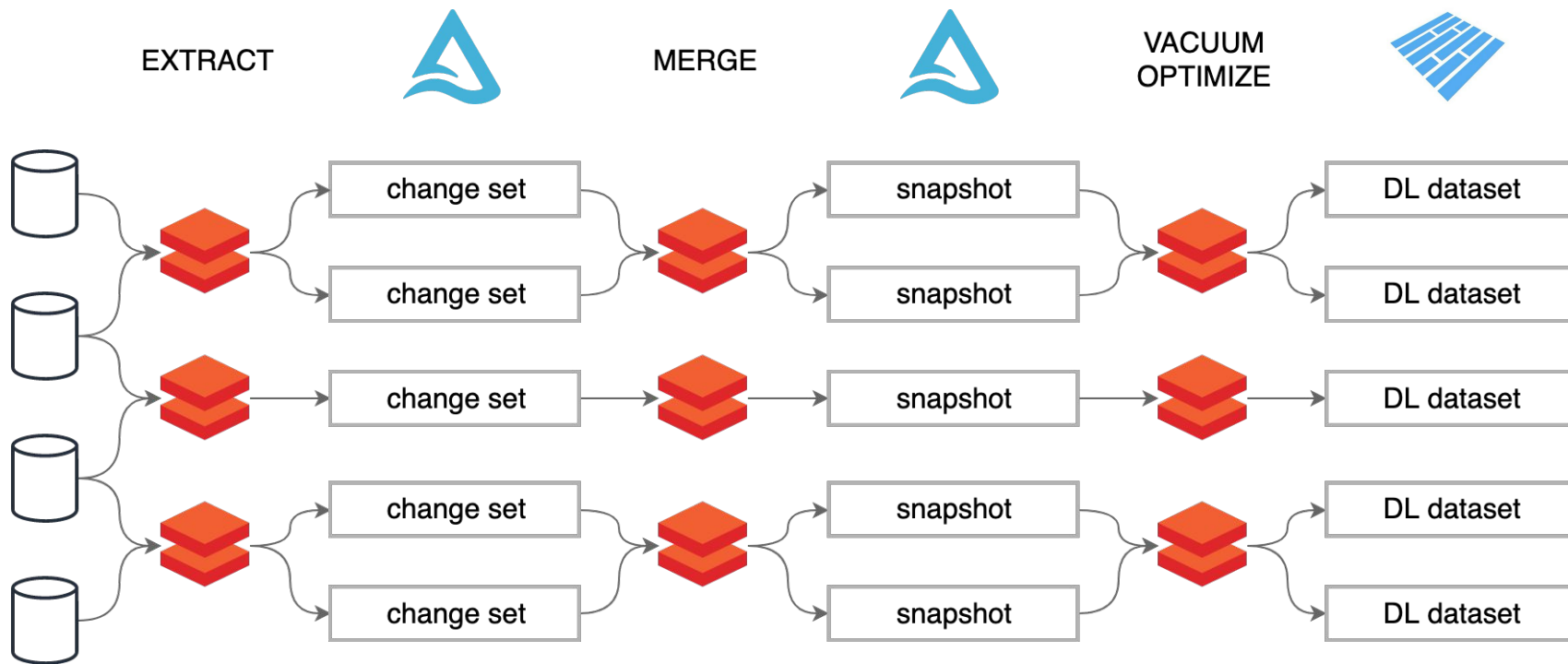


- ❖ Great integration with Spark, including Structured Streaming
  - ❖ Merge syntax in Spark SQL
  - ❖ Time travel
  - ❖ Comprehensive, well written documentation
  - ❖ Fast development backed by a commercial entity
  - ❖ VACUUM + OPTIMIZE
  - ❖ Incoming Presto reader (Starburst)
- 
- 
- ❖ Parquet only
  - ❖ Multicluster writes outside of Databricks only on HDFS



**This Is How We Do It**

# Delta Lake @Zalando





## Moving Forward

# The Future is Bright



# Further Reading

## ACID ORC

<https://orc.apache.org/docs/acid.html>

<https://cwiki.apache.org/confluence/display/Hive/Hive+Transactions>

<http://shzhangji.com/blog/2019/06/10/understanding-hive-acid-transactional-table/>

[https://docs.cloudera.com/HDPDocuments/HDP3/HDP-3.0.0/using-hiveql/content/hive\\_3\\_internals.html](https://docs.cloudera.com/HDPDocuments/HDP3/HDP-3.0.0/using-hiveql/content/hive_3_internals.html)

## Iceberg

<https://iceberg.apache.org/>

<https://iceberg.apache.org/spec/>

<https://github.com/apache/incubator-iceberg>

[https://www.youtube.com/watch?v=z7p\\_m17BXs8](https://www.youtube.com/watch?v=z7p_m17BXs8)

<https://www.youtube.com/watch?v=nWwQMIrjhy0>

## Delta Lake

<https://delta.io/>

<https://github.com/delta-io>

<https://github.com/delta-io/delta/blob/master/PROTOCOL.md>

<https://databricks.com/blog/2019/08/21/diving-into-delta-lake-unpacking-the-transaction-log.html>

<https://databricks.com/blog/2019/09/24/diving-into-delta-lake-schema-enforcement-evolution.html>



# Further Reading

## Engine Support

<https://github.com/prestosql/presto/issues/576>

<https://github.com/prestosql/presto/issues/1324>

<https://github.com/prestosql/presto/pull/1067>

<https://docs.databricks.com/delta/presto-compatibility.html>

<https://www.starburstdata.com/technical-blog/starburst-presto-databricks-delta-lake-support/>

<https://www.qubole.com/blog/qubole-open-sources-multi-engine-support-for-updates-and-deletes-in-data-lakes/>

<https://github.com/qubole/spark-acid>

## S3 Consistency

<https://issues.apache.org/jira/browse/HADOOP-13345>

<https://hadoop.apache.org/docs/r3.0.3/hadoop-aws/tools/hadoop-aws/s3guard.html>

## Other

<https://www.postgresql.org/docs/current/storage.html>

<https://www.postgresql.org/docs/current/routine-vacuuming.html>

<https://dev.mysql.com/doc/refman/8.0/en/optimize-table.html>

<https://medium.com/@brunocrt/the-distributed-architecture-behind-cassandra-database-fba8b5cc4785>

<https://github.com/delta-io/delta/issues/41>



THANK  
YOU

## ACID ORC, Iceberg and Delta Lake

---

Michał Gancarski

[michal.gancarski@zalando.de](mailto:michal.gancarski@zalando.de)

 wssbck