Redpanda

STREAMFEST

The Future of Data Streaming: Ubiquitous, Unified, Efficient

Yaroslav Tkachenko

about me…

Yaroslav Tkachenko

SOFTWARE ENGINEER, CONSULTANT, ADVISOR

- I’ve been building data platforms for the past ~7 years, primarily with data streaming tech.
- Founding Engineer at **Goldsky**, Staff Data Engineer at **Shopify**, Software Architect at **Activision**.
- Evangelizing and advocating for streaming systems.

Data streaming is ubiquitous

New requirements

Data products

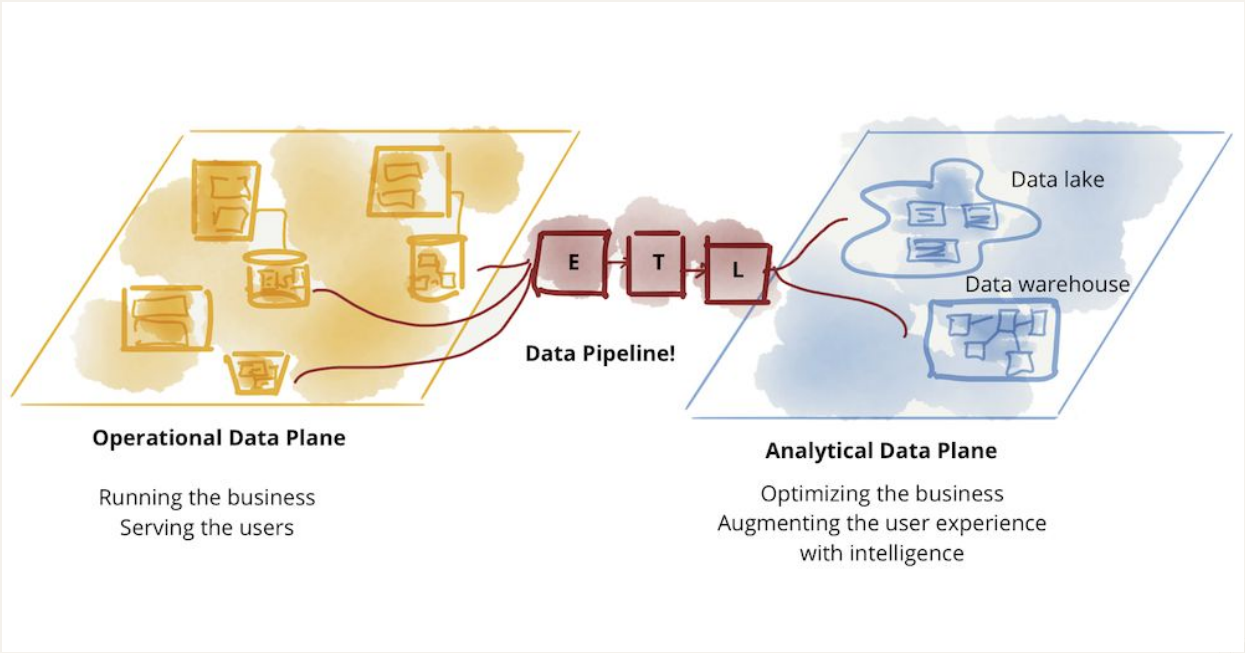
- User facing products leveraging internal data lakes and data warehouses.
- Different reliability expectations.

GenAI

- Fresh, personalized datasets.
- More data integration workflows (vector search, RAG, etc.)

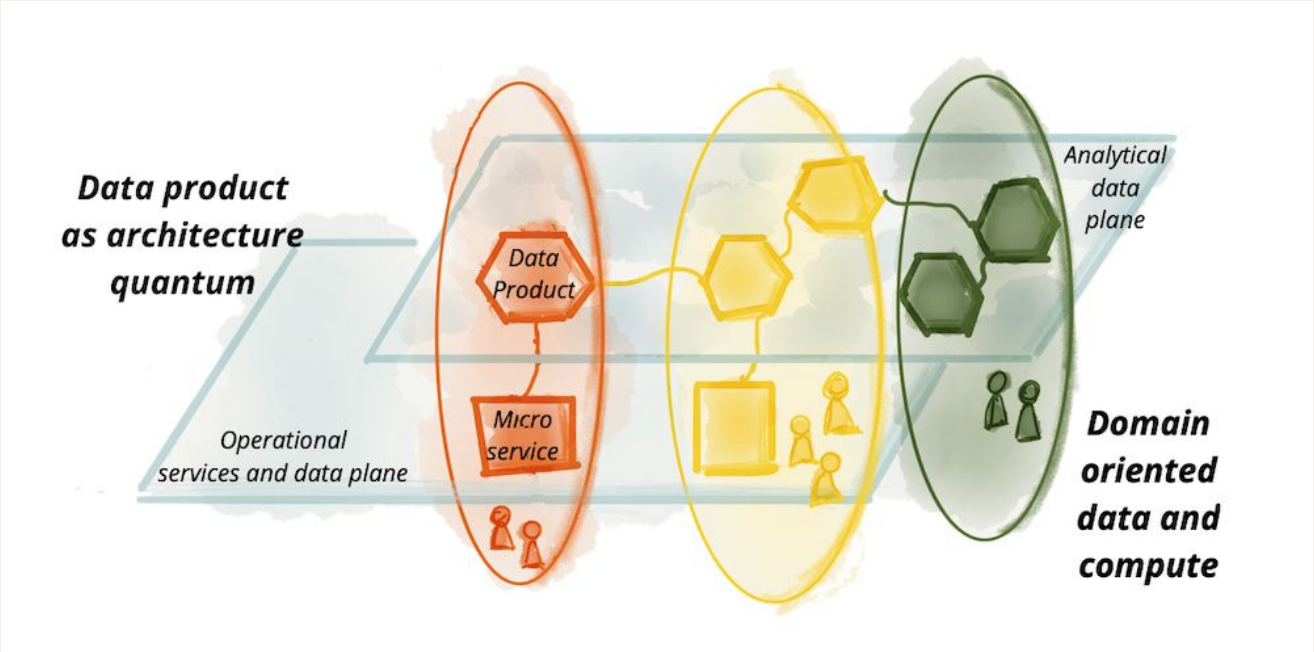
Cost efficiency

- End of ZIRP.
- Reduced headcount.



Data Mesh Principles and Logical Architecture by Zhamak Dehghani

“Shifting left to make it right”

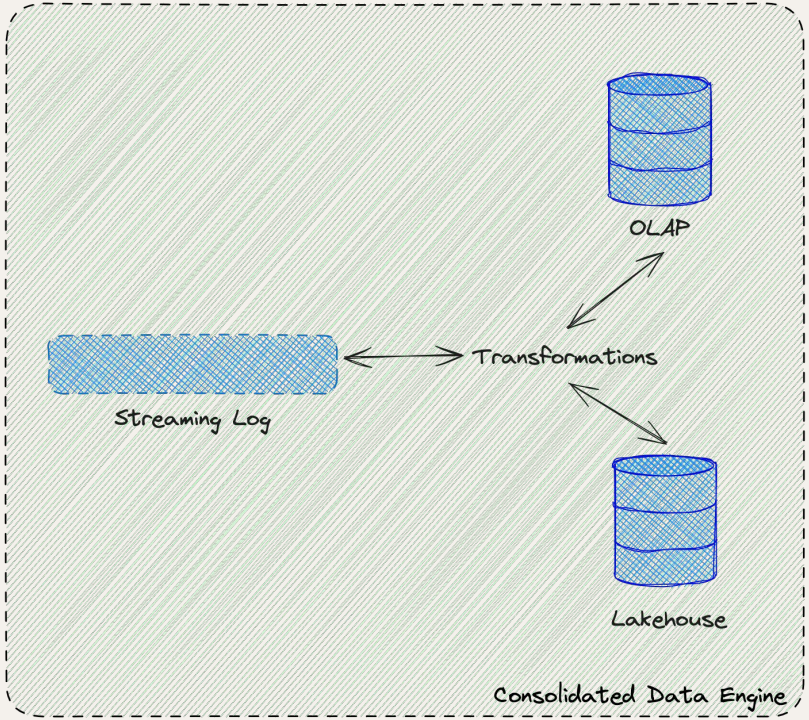


Data Mesh Principles and Logical Architecture by Zhamak Dehghani

Why are we not there yet?

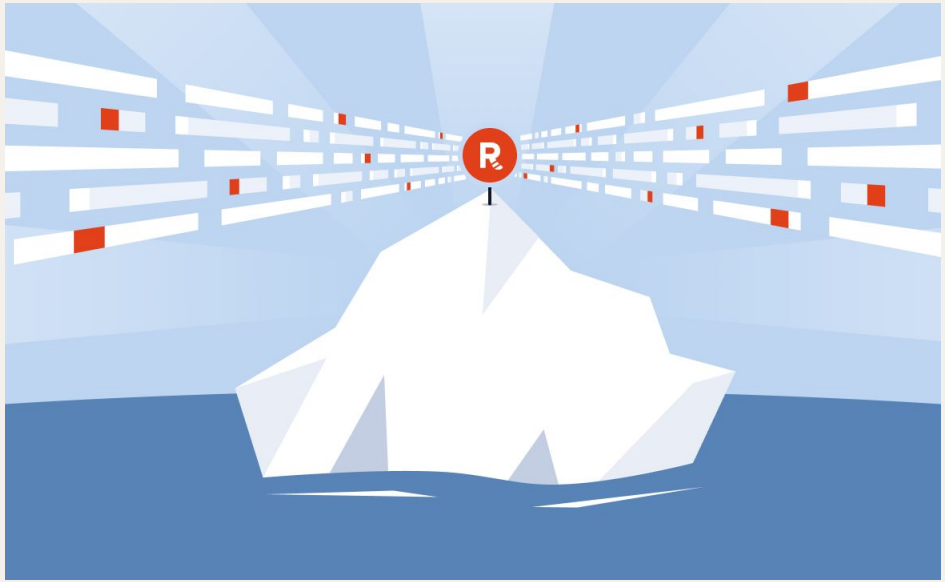
- Change Data Capture (CDC) tools took a while to become widely used.
- This is still mostly organizational problem. Data products must be recognized and given enough importance.
- Data stack maturity. Building end-to-end data products is hard. But it shouldn’t be.

Data streaming is unified



Data Platforms in 2030

Kafka API + Iceberg API



Apache Iceberg Topics: Stream directly into your data lake

~~Batch~~ ~~stream~~ processing

- Use **lakehouse** (e.g. Iceberg) for **backfills and reprocessing** (as well as **historical queries**). Use **streaming platform** (e.g. Redpanda) for **everything else**.
- Stateful stream processing could be changing.
  - FLIP-486: DeltaJoin
- You won't need to choose between batch and stream processing. We'll focus have **data processing**. With streaming or incremental semantics by default.

SQL is not going anywhere

```
SELECT user_id, count
FROM
  CountWithTimeout(
    input => TABLE(data) PARTITION BY user_id,
    on_time => DESCRIPTOR(rowtime),
    uid => 'main-counting'
  )
```

Polymorphic Table Functions in Apache Flink

Data streaming is efficient

Cloud storage is everywhere

- OLTP and OLAP databases, vector databases, embedded key-value databases!
- Disaggregated storage in streaming frameworks.
- It’s not just about the cost. Cloud storage enabled compute/storage separation and comes with useful primitives like conditional writes that can simplify architecture.
- Streaming platforms took notice as well: Redpanda One Cloud Topics, WarpStream, Confluent Cloud Freight, etc.

New wave of data infra tools

- Arrow, DataFusion, Comet, Velox, Substrait, DuckDB, etc.
  - Mostly Rust and C++.
- Focus on **modularity**, **composability** and **efficiency**.
  - The Composable Data Management System Manifesto is worth reading.

Thanks for joining!

Yaroslav Tkachenko

SOFTWARE ENGINEER, CONSULTANT, ADVISOR

- **@sap1ens**
- https://sap1ens.com
- Newsletter: https://streamingdata.tech