Data-Intensive Distributed Computing

CS 451/651 (Fall 2025)

UNIVERSITY OF WATERLOO

Data Warehouses, Data Lakes, and Lakehouses

(v1.00)

Week 2: September 9

Jimmy Lin

David R. Cheriton School of Computer Science

University of Waterloo

These slides are available at `https://lintool.github.io/cs451-2025f/`

Key Questions

What are the main differences between
operational and analytical infrastructure?

What are data warehouses?
What problems did they evolve to solve?

What are data lakes and lakehouses?
What problems did they evolve to solve?

What are the components of modern data platforms?

How do operational and analytical data models differ?

What goes on in ETL/ELT?

How do different physical representations of data affect
storage, compute, and other tradeoffs within data platforms?

This Week

Now: Evolution of Data Platforms

Data Warehouses, Data Lakes, and Lakehouses

Next: Three Deep Dives

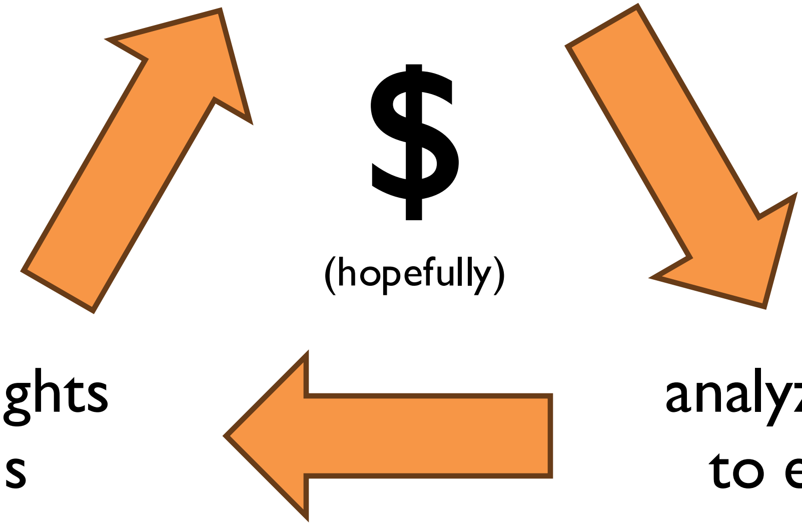
Data Modeling, ELT, Physical Representations

Context…

The Data Flywheel

(a virtuous cycle)

Build a useful product



$

(hopefully)

analyze user behavior to extract insights

transform insights into actions

Google. Facebook. Twitter. Amazon. Uber.

Context…

What's this course about?

The *infrastructure* that supports the data flywheel.

data platforms + data engineering

Context…

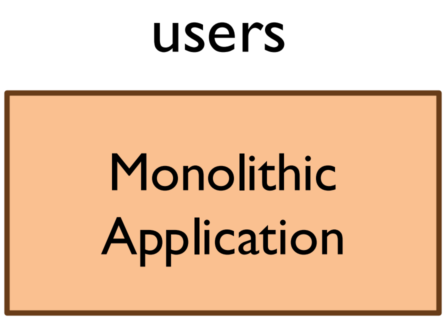
What problems do data platforms solve?

Ingesting, storing, manipulating, maintaining, serving…
the data that supports the data flywheel.

Evolution of Data Platforms

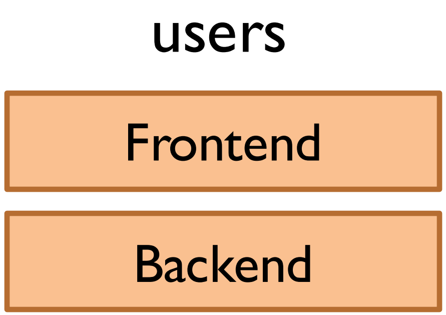
Data Warehouses, Data Lakes, and Lakehouses

In the beginning…



users

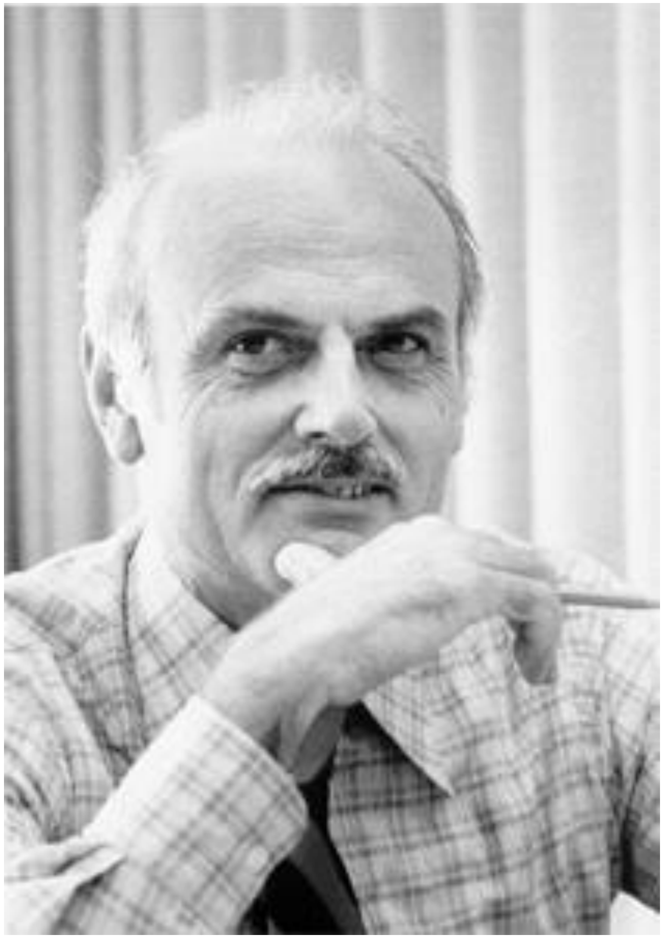
Monolithic Application



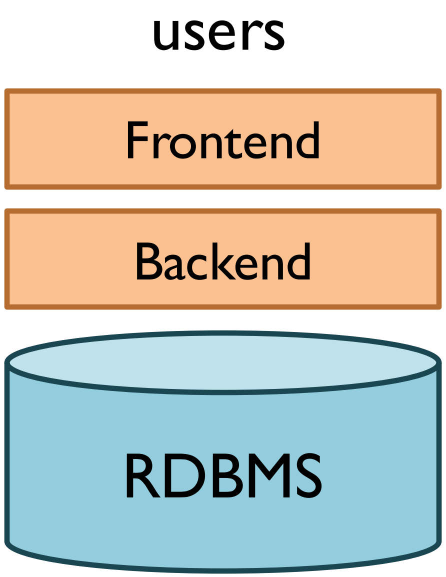
users

Frontend

Backend



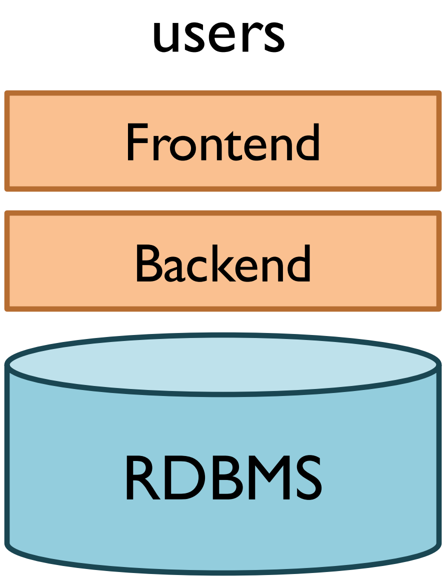
Source: Wikipedia


users
Frontend
Backend
RDBMS

RDBMS = Relational Database Management System

Imposes a relational view of data: tables, rows, columns

Provides a set of relational operators to manipulate data: SQL



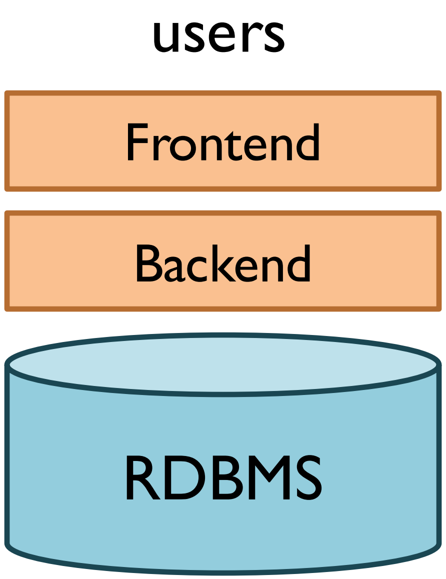
Why is this a good idea?

Offload physical data design

Standardize query processing

Ensure data integrity, manage concurrency

Handle backup and recovery



users

Frontend

Backend

RDBMS

Okay, but why *relational*?

RDBMS = Relational Database Management System

Imposes a relational view of data: tables, rows, columns

Provides a set of relational operators to manipulate data: SQL

But we're constrained by the relational model?

(1) The relational model is surprisingly general.

(2) More later…

Remember from last time?

Business Intelligence: Case Study

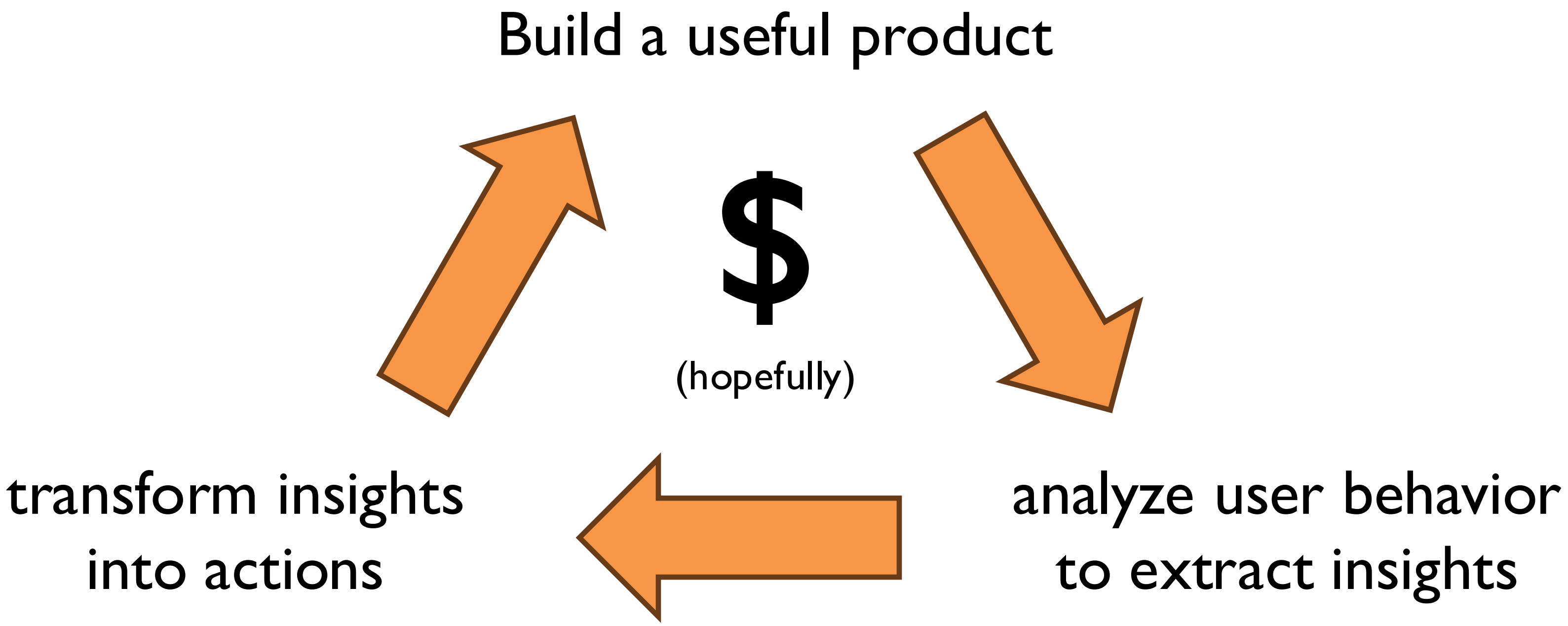
In the 1990s, Wal-Mart found that customers tended to buy diapers and beer together. So they put them next to each other and increased sales of both.*

* BTW, this is completely apocryphal. (But it makes a nice story.)

Remember from last time?

The Data Flywheel

(a virtuous cycle)



Build a useful product

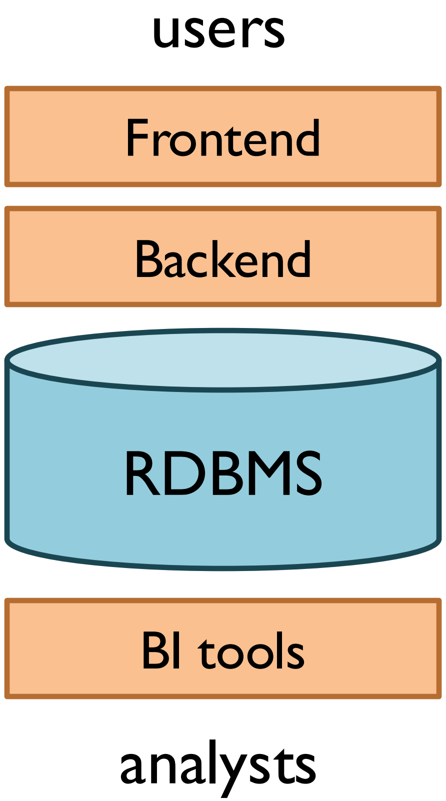
analyze user behavior
to extract insights

transform insights
into actions

$

(hopefully)

Google. Facebook. Twitter. Amazon. Uber.



users

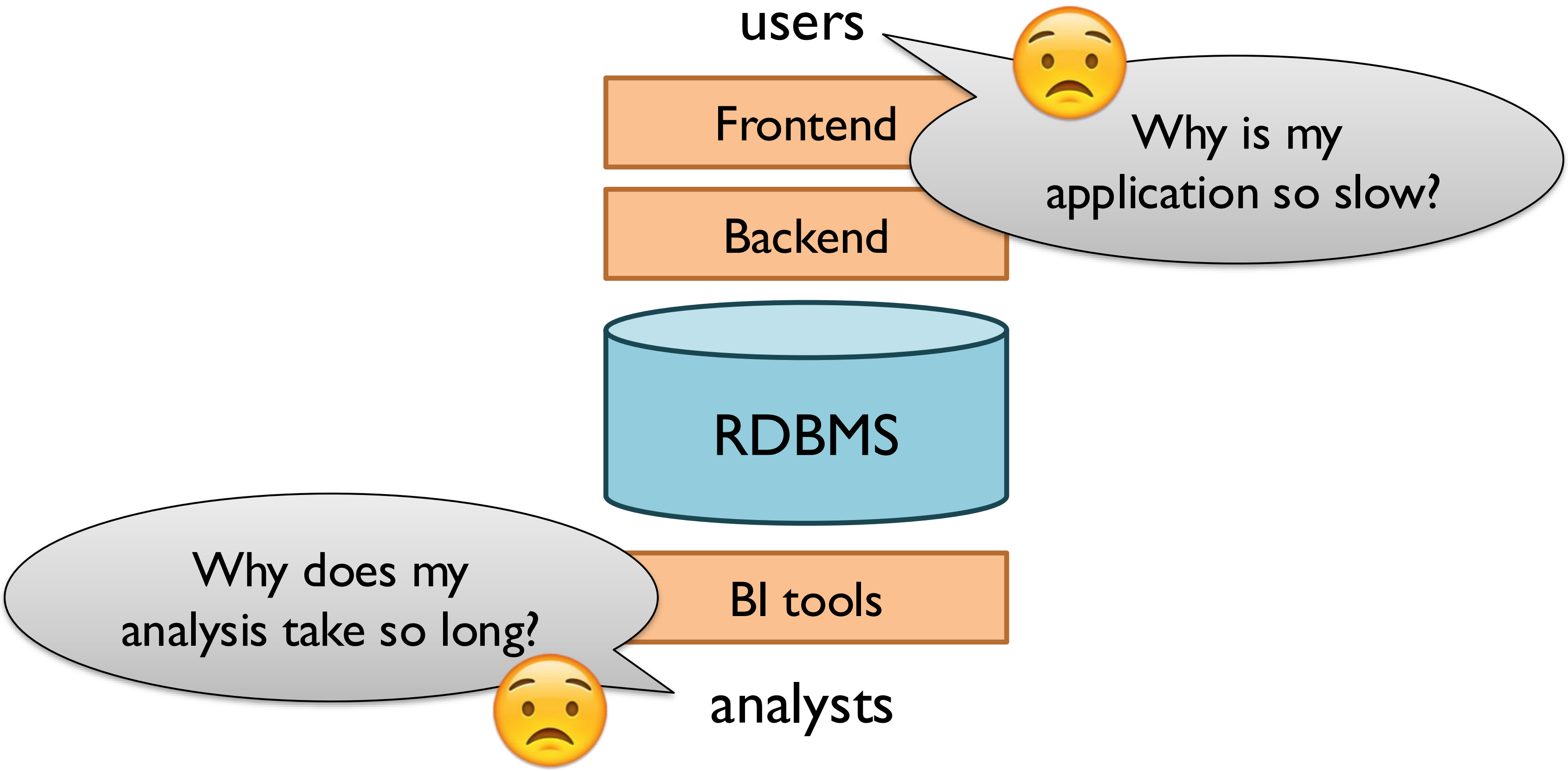
Frontend

Backend

RDBMS

BI tools

analysts



users

Frontend

Backend

RDBMS

BI tools

analysts

Why is my application so slow?

Why does my analysis take so long?

RDBMS Workloads

OLTP (online transaction processing)

Typical applications: e-commerce, banking, airline reservations

Customer-facing: real-time, low latency, highly-concurrent

Tasks: relatively small set of transactional queries; CRUD

Data access pattern: random reads, updates, writes (small amounts of data)

OLAP (online analytical processing)

Typical applications: business intelligence, data mining

Back-end processing: batch workloads, less concurrency

Tasks: complex analytical queries, often ad hoc

Data access pattern: table scans, large amounts of data per query

OLTP and OLAP Together?

Downsides of co-existing OLTP and OLAP workloads

Conflicting queries and data access patterns

Poor memory management

Variable latency

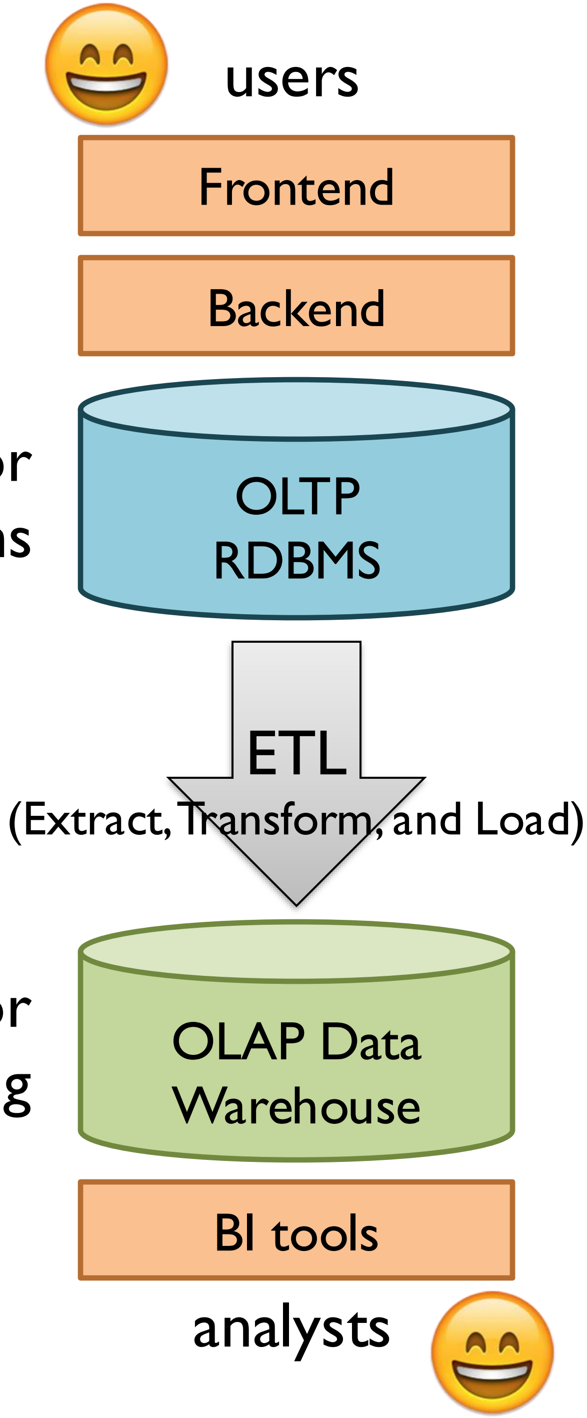
😟 users and analysts

Solution? Separate the two!

Build a data warehouse!

Source: Wikipedia (Warehouse)

OLTP database for user-facing transactions

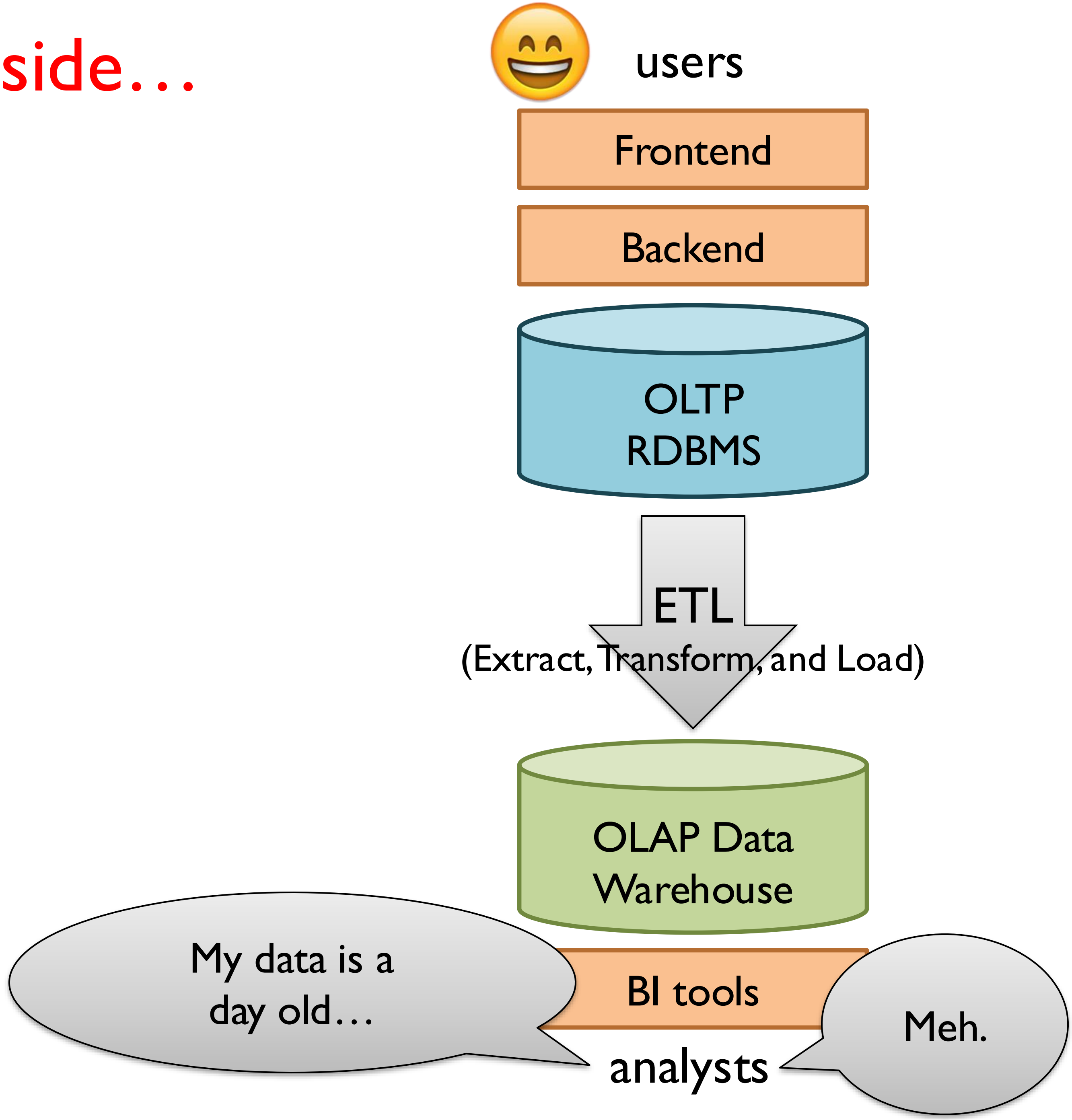


OLAP database for data warehousing

Where and *when* does this actually happen?

Deep Dive later…

Aside…



users

Frontend

Backend

OLTP RDBMS

ETL

(Extract, Transform, and Load)

OLAP Data Warehouse

BI tools

analysts

My data is a day old…

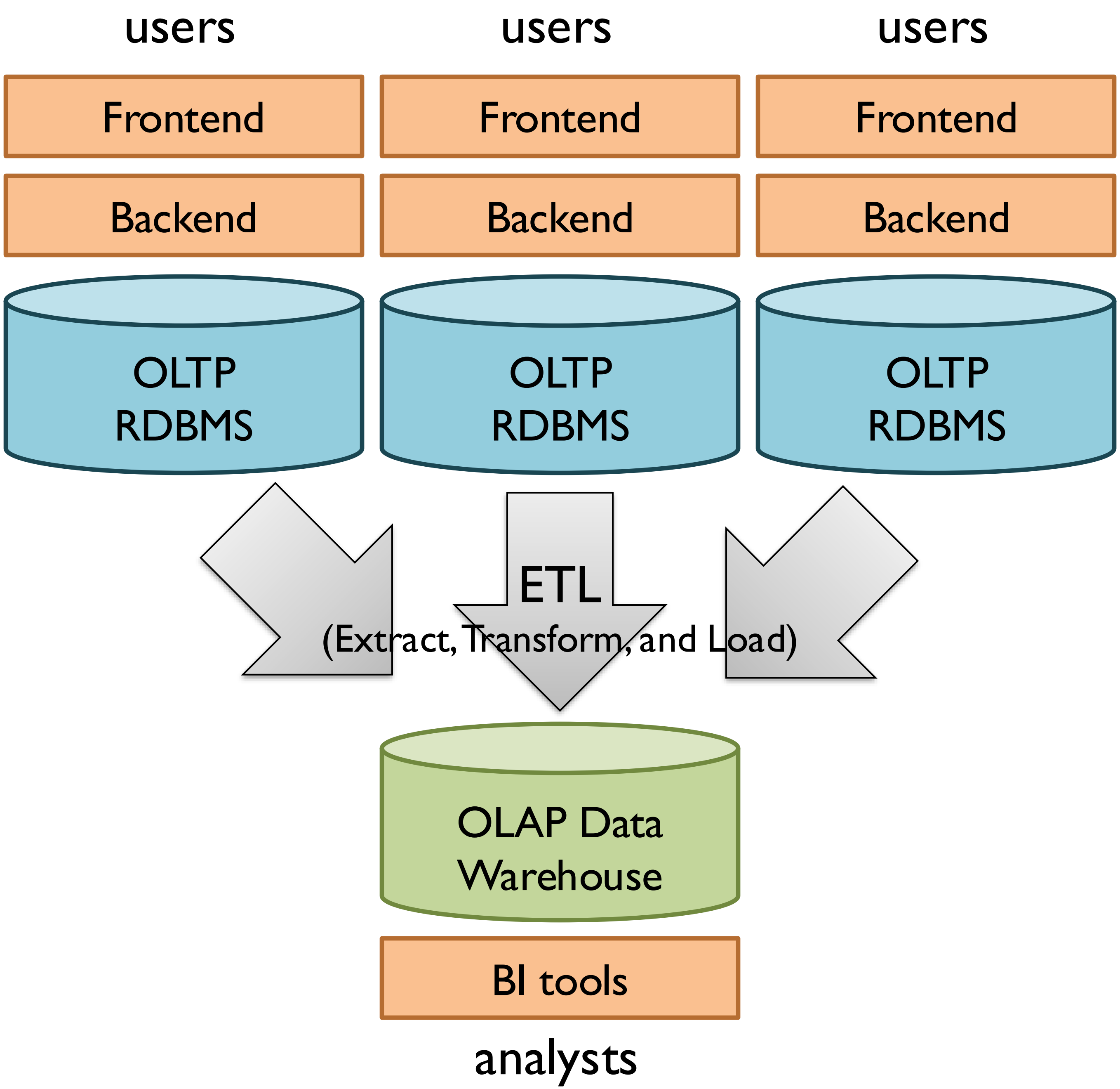
Meh.

Aside…

Why is this a difficult problem?

Characteristics of the data: Volume, Velocity, Variety[*] + Veracity

[*] Coined by Gartner analyst Doug Laney in 2001.


users

users

users

Frontend

Frontend

Frontend

Backend

Backend

Backend

OLTP RDBMS

OLTP RDBMS

OLTP RDBMS

ETL

(Extract, Transform, and Load)

OLAP Data Warehouse

BI tools

analysts

facebook®

Jeff Hammerbacher, Information Platforms and the Rise of the Data Scientist.
In, *Beautiful Data*, O'Reilly, 2009.

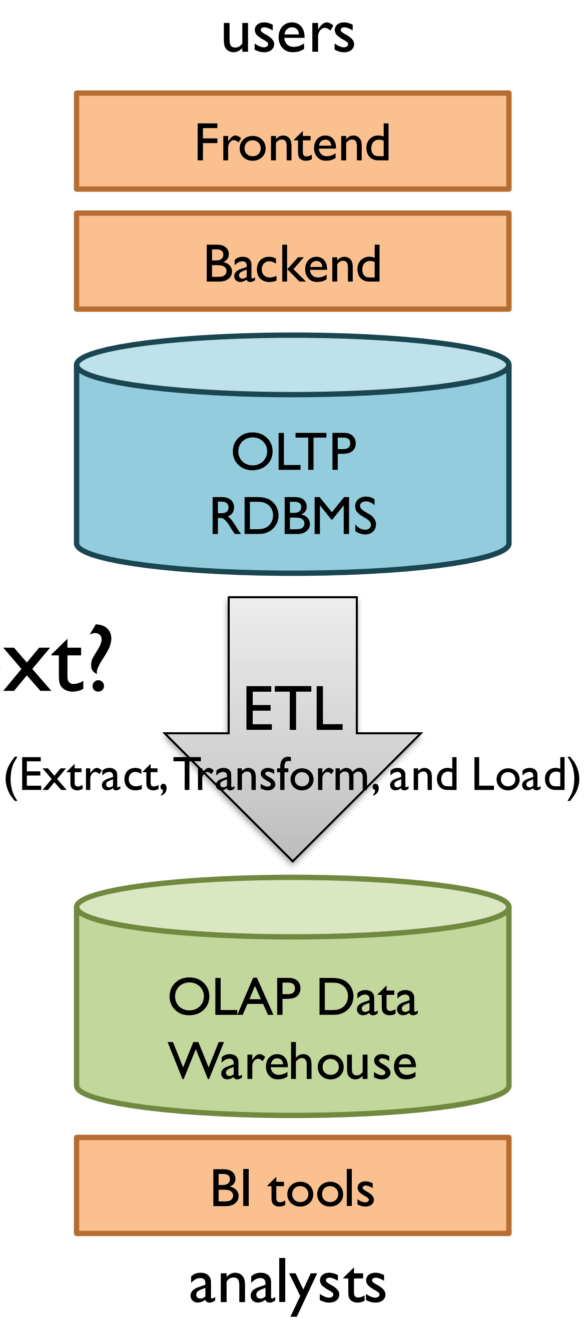
> "On the first day of logging the Facebook clickstream, more than 400 gigabytes of data was collected. The load, index, and aggregation processes for this data set really taxed the Oracle data warehouse. Even after significant tuning, we were unable to aggregate a day of clickstream data in less than 24 hours."

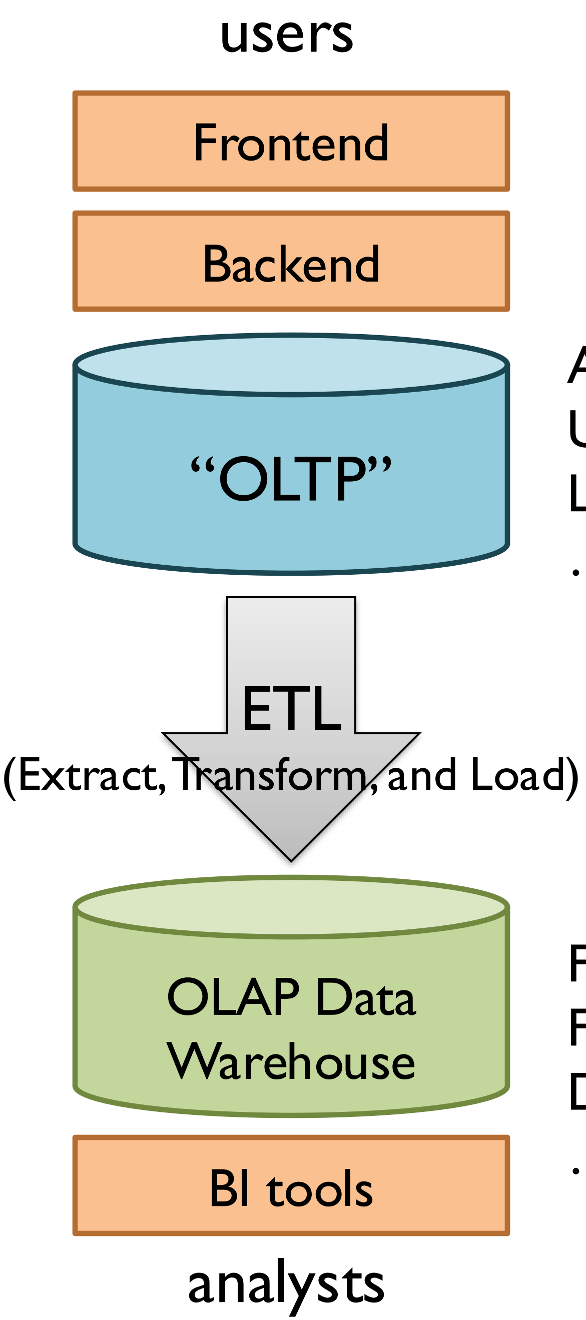
Why is this a difficult problem?

Characteristics of the data: Volume, Velocity, Variety[*] + Veracity

[*] Coined by Gartner analyst Doug Laney in 2001.

Facebook context?





users

Frontend

Backend

"OLTP"

ETL

(Extract, Transform, and Load)

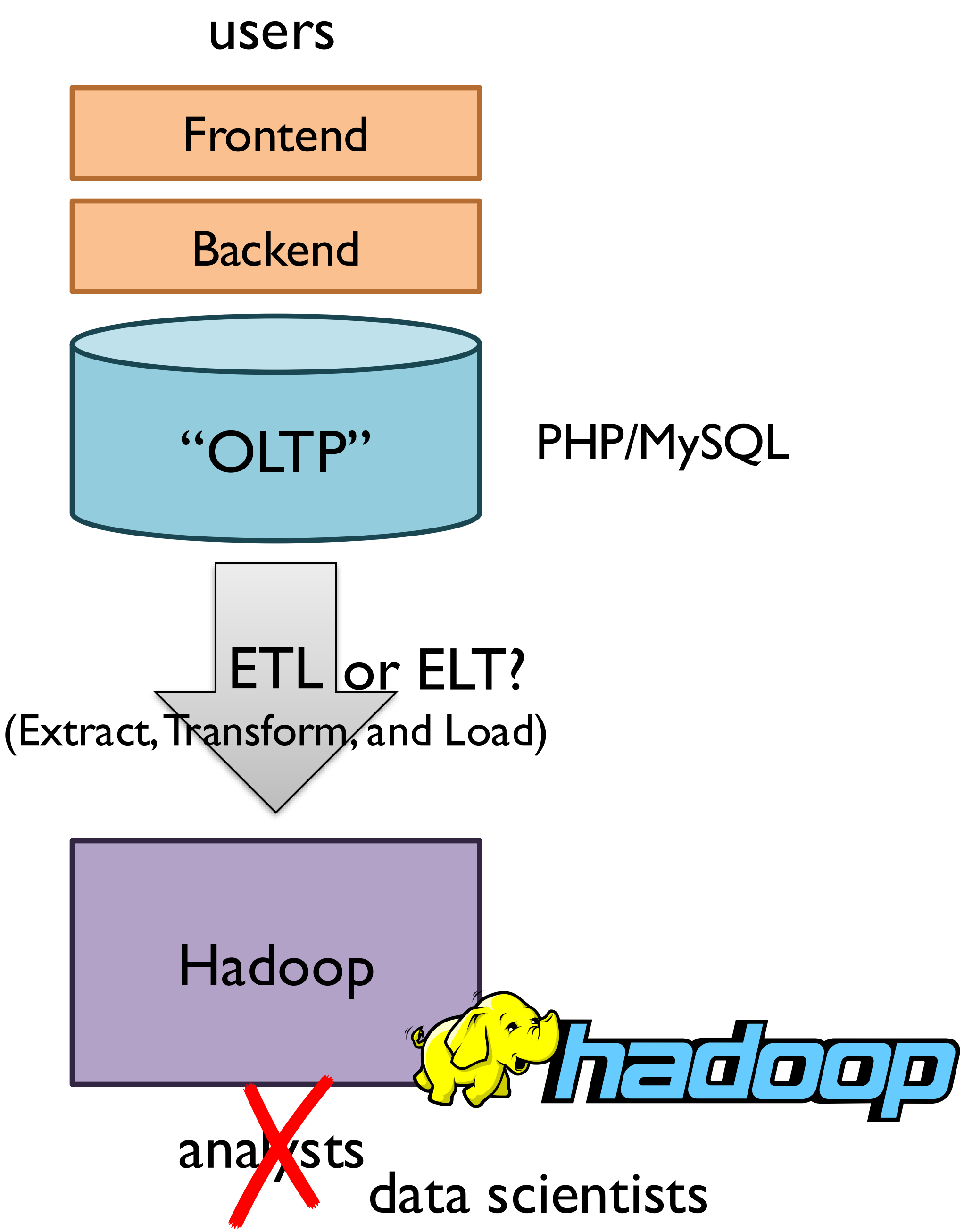
OLAP Data Warehouse

BI tools

analysts

Adding friends
Updating profiles
Likes, comments
…

Feed ranking
Friend recommendation
Demographic analysis
…



users

Frontend

Backend

“OLTP”

PHP/MySQL

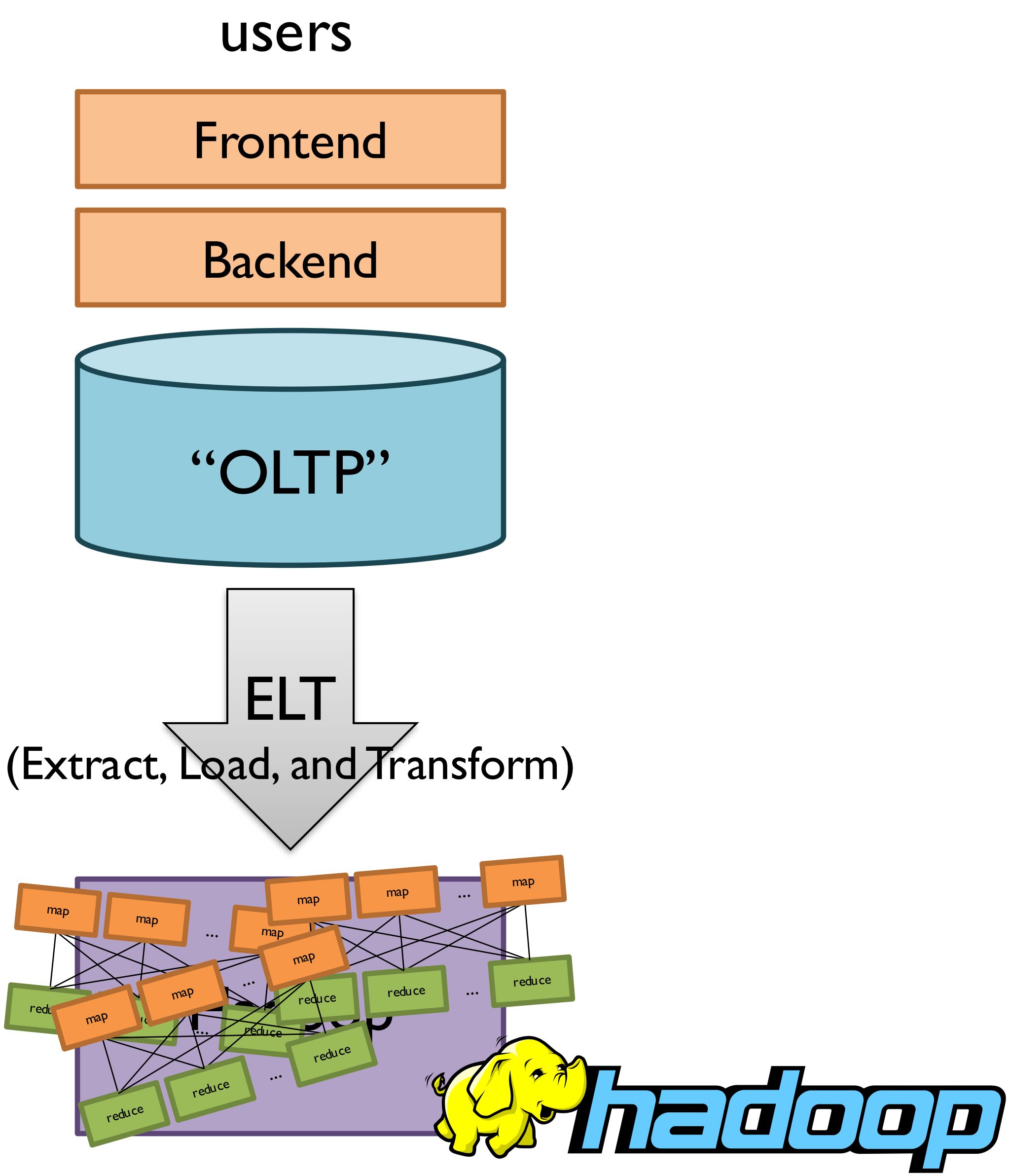
ETL or ELT?

(Extract, Transform, and Load)

Hadoop

~~analysts~~

data scientists



users

Frontend

Backend

“OLTP”

ELT

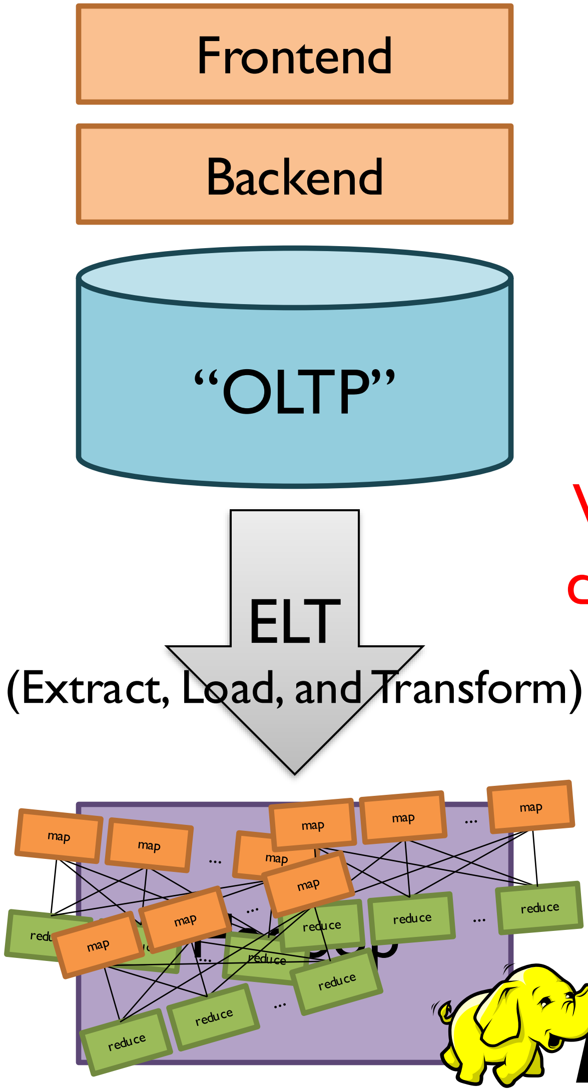
(Extract, Load, and Transform)

data scientists

Data storage on HDFS
Processing with MapReduce

The Irony…

users



Frontend

Backend

"OLTP"

ELT

(Extract, Load, and Transform)

data scientists

Wait, so why not use a database to begin with?

Data storage on HDFS
Processing with MapReduce

Why not just use an existing analytical database?

Scalability. Cost. Flexibility.

Jeff Hammerbacher, Information Platforms and the Rise of the Data Scientist.
In, *Beautiful Data*, O'Reilly, 2009.

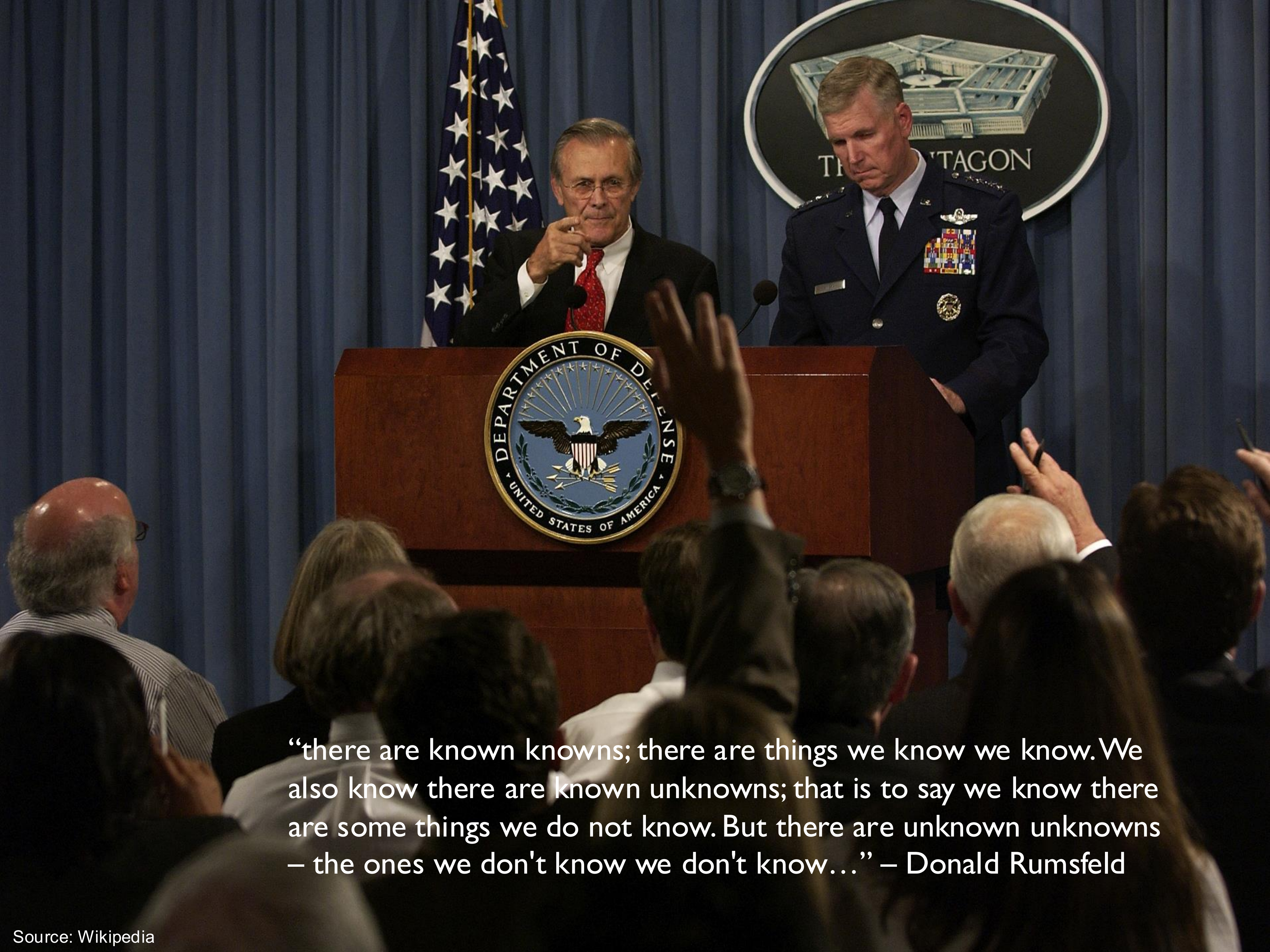
> "On the first day of logging the Facebook clickstream, more than 400 gigabytes of data was collected. The load, index, and aggregation processes for this data set really taxed the Oracle data warehouse. Even after significant tuning, we were unable to aggregate a day of clickstream data in less than 24 hours."

Databases are great…

If your data has structure (and you know what the structure is)
If your data is reasonably clean
If you know what queries you’re going to run ahead of time

Databases are not so great…

If your data has little structure (or you don’t know the structure)
If your data is messy and noisy
If you don’t know what you’re looking for



"there are known knowns; there are things we know we know. We also know there are known unknowns; that is to say we know there are some things we do not know. But there are unknown unknowns – the ones we don't know we don't know…" – Donald Rumsfeld

Source: Wikipedia

Databases are great…

If your data has structure (and you know what the structure is)

If your data is reasonably clean

If you know what queries you're going to run ahead of time

Known unknowns!

Databases are not so great…

If your data has little structure (or you don't know the structure)

If your data is messy and noisy

If you don't know what you're looking for

Unknown unknowns!

It’d be great…

If I could just ingest all my data

(relational, semi-structured, unstructured, graph, multimodal, etc.)

in a multitude of formats

(text, csv, json, etc.)

and figure out what to do with it later.

hadoop Data storage on HDFS

(later, Amazon S3, Google Cloud Storage, Azure Blob Storage, etc.)

If I could then processing / manipulate / transform the data…

(I want to write *ad hoc* scripts ***and*** SQL queries)

hadoop Processing with MapReduce

(later, Spark, Presto, Trino, Flink, etc.)

tl;dr – I want *flexibility*.

What is a data lake?

A data lake is a central location that holds a large amount of data in its native, raw format… a data lake uses a flat architecture and object storage to store the data… By leveraging inexpensive object storage and open formats, data lakes enable many applications to take advantage of the data.

https://www.databricks.com/discover/data-lakes

A data lake is a centralized repository that ingests and stores large volumes of data in its original form. The data can then be processed and used as a basis for a variety of analytic needs. Due to its open, scalable architecture, a data lake can accommodate all types of data from any source, from structured… to semi-structured… to unstructured… The data files are typically stored in staged zones—raw, cleansed, and curated—so that different types of users may use the data in its various forms to meet their needs. Data lakes provide core data consistency across a variety of applications, powering big data analytics, machine learning, predictive analytics, and other forms of intelligent action.

https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-a-data-lake

What is a data lake?

A data lake is a central location that holds a large amount of data in its native, raw format… a data lake uses a flat architecture and object storage to store the data… By leveraging inexpensive object storage and open formats, data lakes enable many applications to take advantage of the data.

https://www.databricks.com/discover/data-lakes

A data lake is a centralized repository that ingests and stores large volumes of data in its original form. The data can then be processed and used as a basis for a variety of analytic needs. Due to its open, scalable architecture, a data lake can accommodate all types of data from any source, from structured… to semi-structured… to unstructured… The data files are typically stored in staged zones—raw, cleansed, and curated—so that different types of users may use the data in its various forms to meet their needs. Data lakes provide core data consistency across a variety of applications, powering big data analytics, machine learning, predictive analytics, and other forms of intelligent action.

https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-a-data-lake

Check?

It’d be great…

If I could just ingest all my data

(relational, semi-structured, unstructured, graph, multimodal, etc.)

in a multitude of formats

(text, csv, json, etc.)

and figure out what to do with it later.

hadoop Data storage on HDFS

(later, Amazon S3, Google Cloud Storage, Azure Blob Storage, etc.)

If I could then processing / manipulate / transform the data…

(I want to write *ad hoc* scripts ***and*** SQL queries)

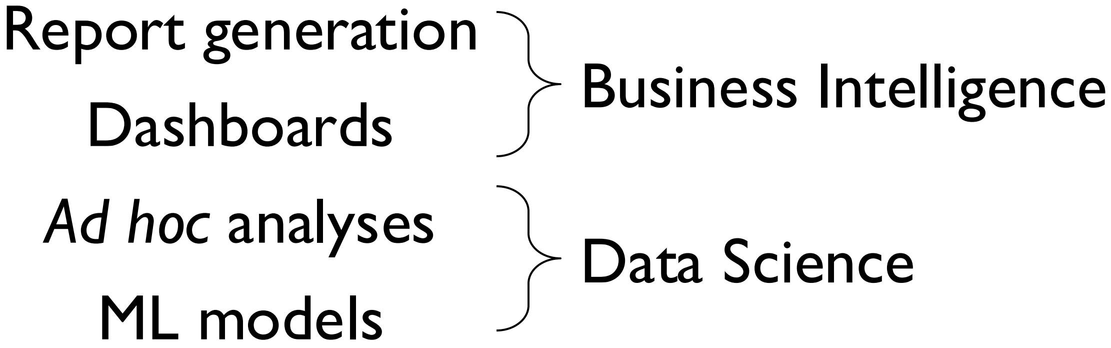
hadoop Processing with MapReduce

(later, Spark, Presto, Trino, Flink, etc.)

Reminder…

Transform Insights into Actions

What does that really mean?



Report generation

Dashboards

Business Intelligence

Ad hoc analyses

ML models

Data Science

known unknowns and unknown unknowns?

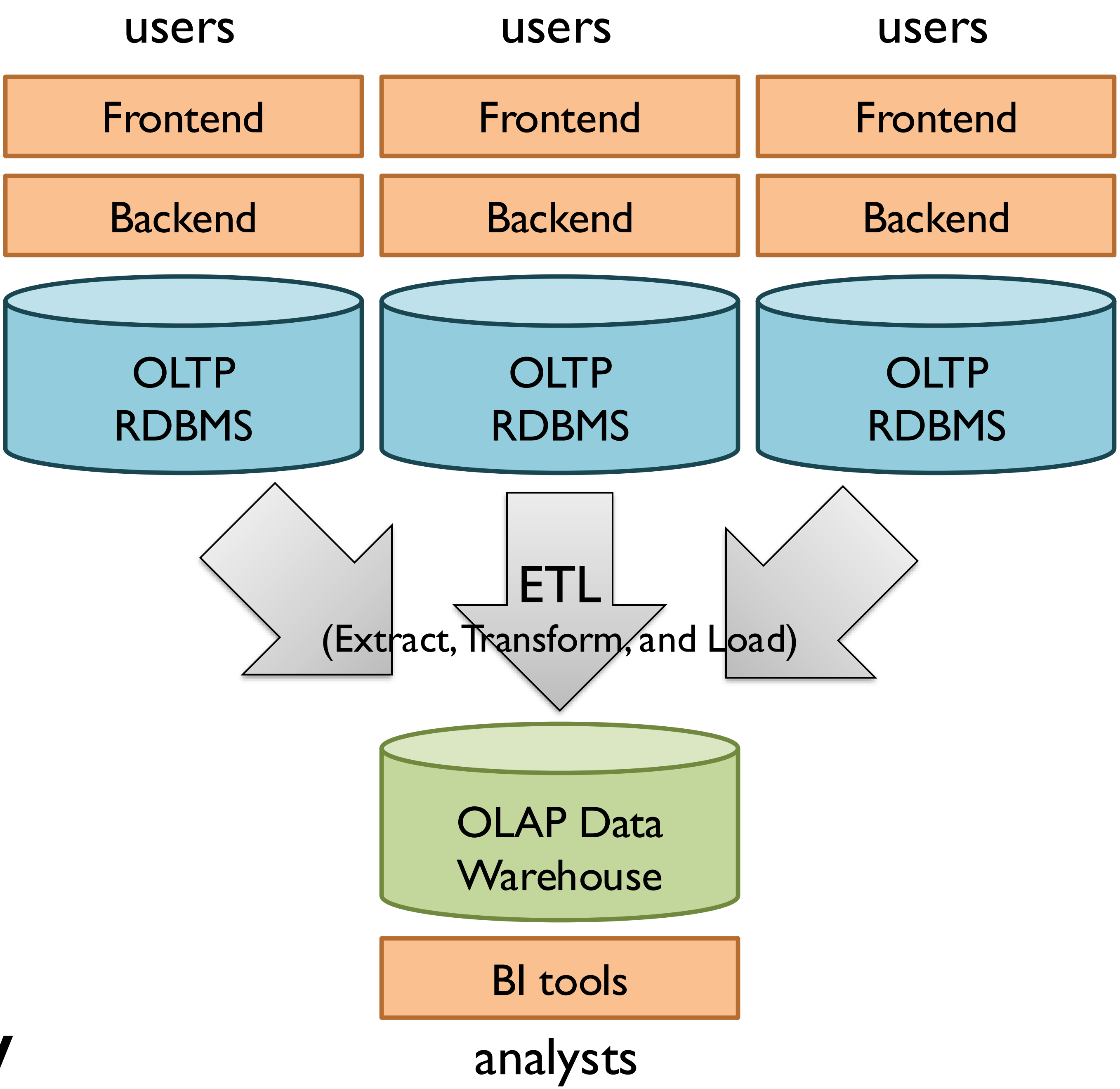
EDL vs. EDW

All types of data
vs. only relational data

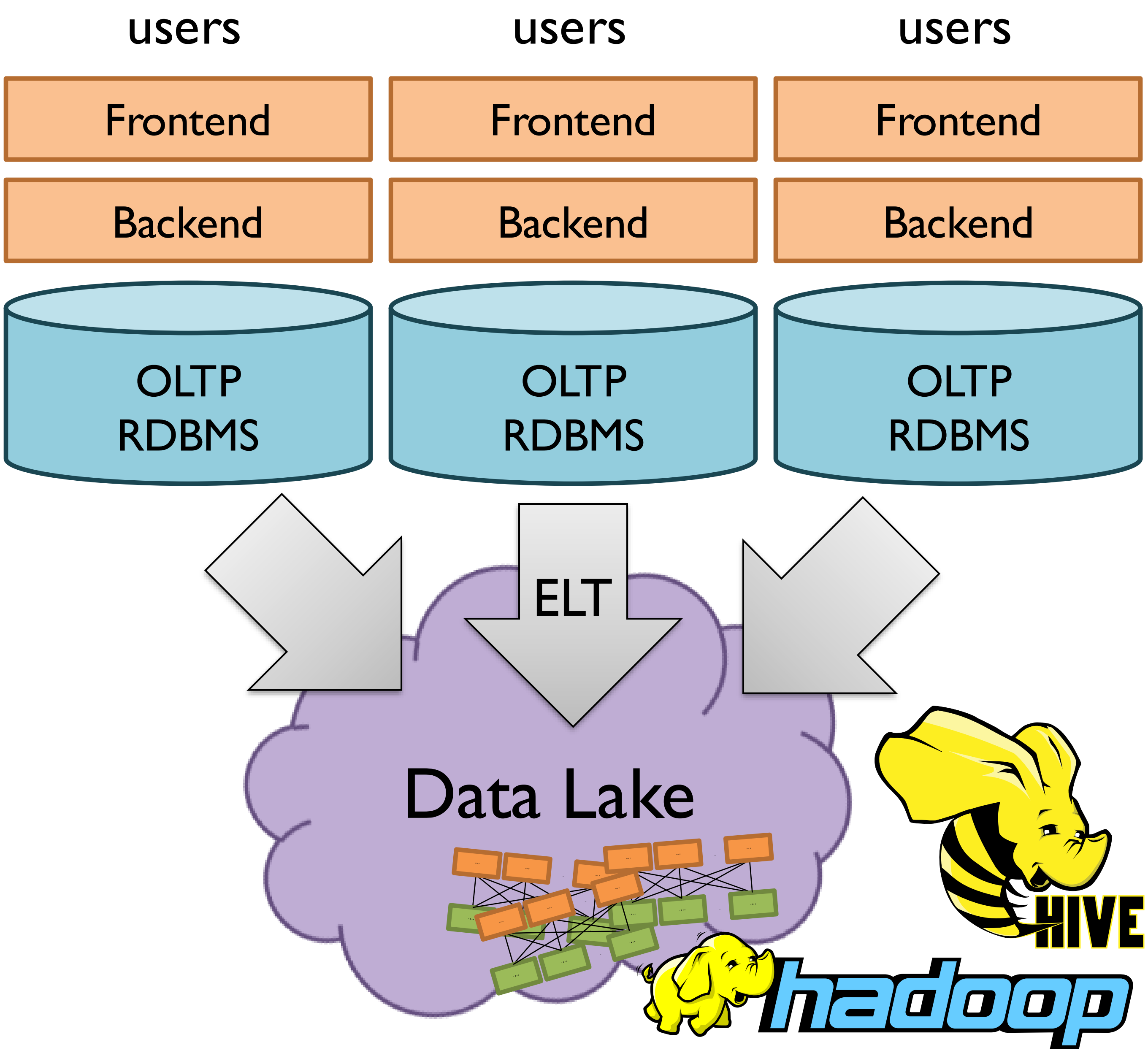
Open data formats
vs. proprietary formats

Cheap (object) storage
vs. expensive file systems

Flexible query processing
vs. only SQL


users
users
users
Frontend
Frontend
Frontend
Backend
Backend
Backend
OLTP RDBMS
OLTP RDBMS
OLTP RDBMS
ETL
(Extract, Transform, and Load)
OLAP Data Warehouse
BI tools
analysts

EDW



users | users | users

Frontend | Frontend | Frontend

Backend | Backend | Backend

OLTP RDBMS | OLTP RDBMS | OLTP RDBMS

ELT

Data Lake

data scientists

EDL (1st gen)

Data Lakes to Data Swamps

tl;dr – I want *flexibility*.
Be careful what you wish for…

Why can't I find anything?

Where's the schema?

Why are there all these multiple near-duplicate copies?

Where did this come from?

Did this job actually finish?

Am I supposed to have access to this?

"jank"

The “Lakehouse” Vision

A unified metadata and governance layer over data lakes

Transaction support

Schema enforcement and governance

Storage is decoupled from compute

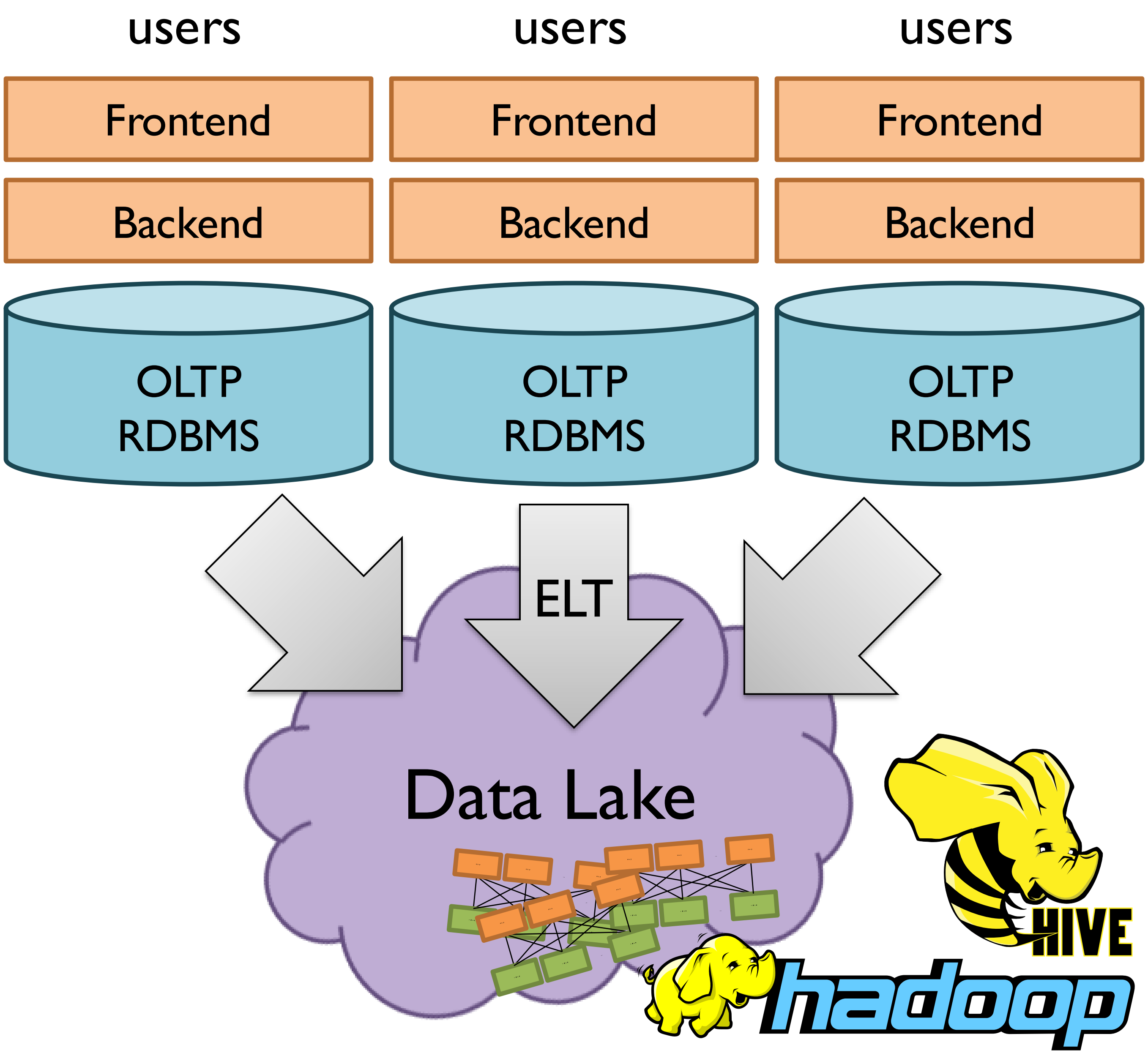
Openness

Support for diverse data types

Support for diverse workloads

https://www.databricks.com/blog/2020/01/30/what-is-a-data-lakehouse.html

EDL (1st gen)



users

Frontend

Backend

OLTP RDBMS

users

Frontend

Backend

OLTP RDBMS

users

Frontend

Backend

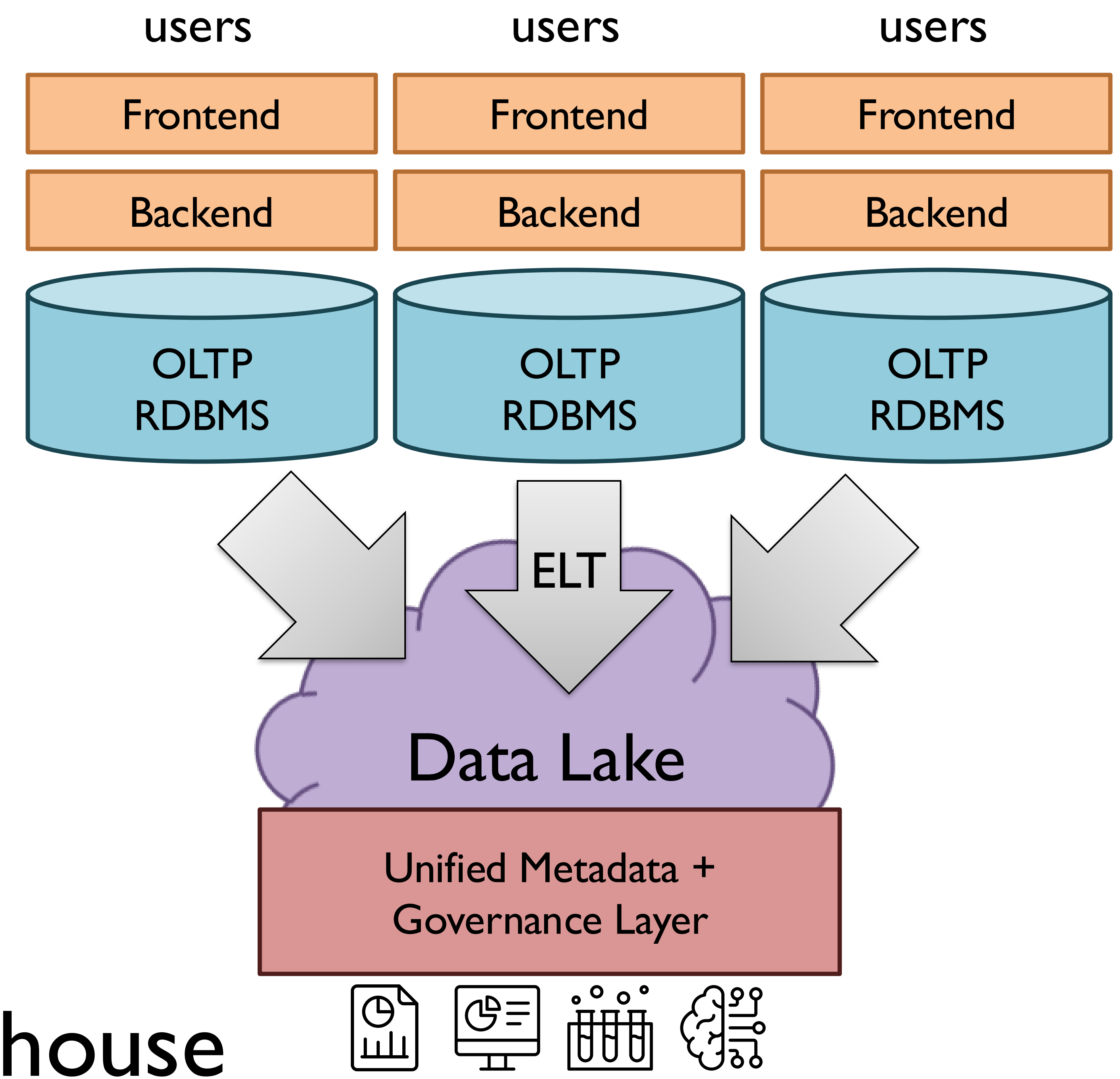
OLTP RDBMS

ELT

Data Lake

data scientists

Lakehouse



users users users

Frontend Frontend Frontend

Backend Backend Backend

OLTP RDBMS OLTP RDBMS OLTP RDBMS

ELT

Data Lake

Unified Metadata + Governance Layer

EDL vs. EDW

Revenge of the DBs

All types of data ✓

vs. only relational data

Open data formats ✓

vs. proprietary formats

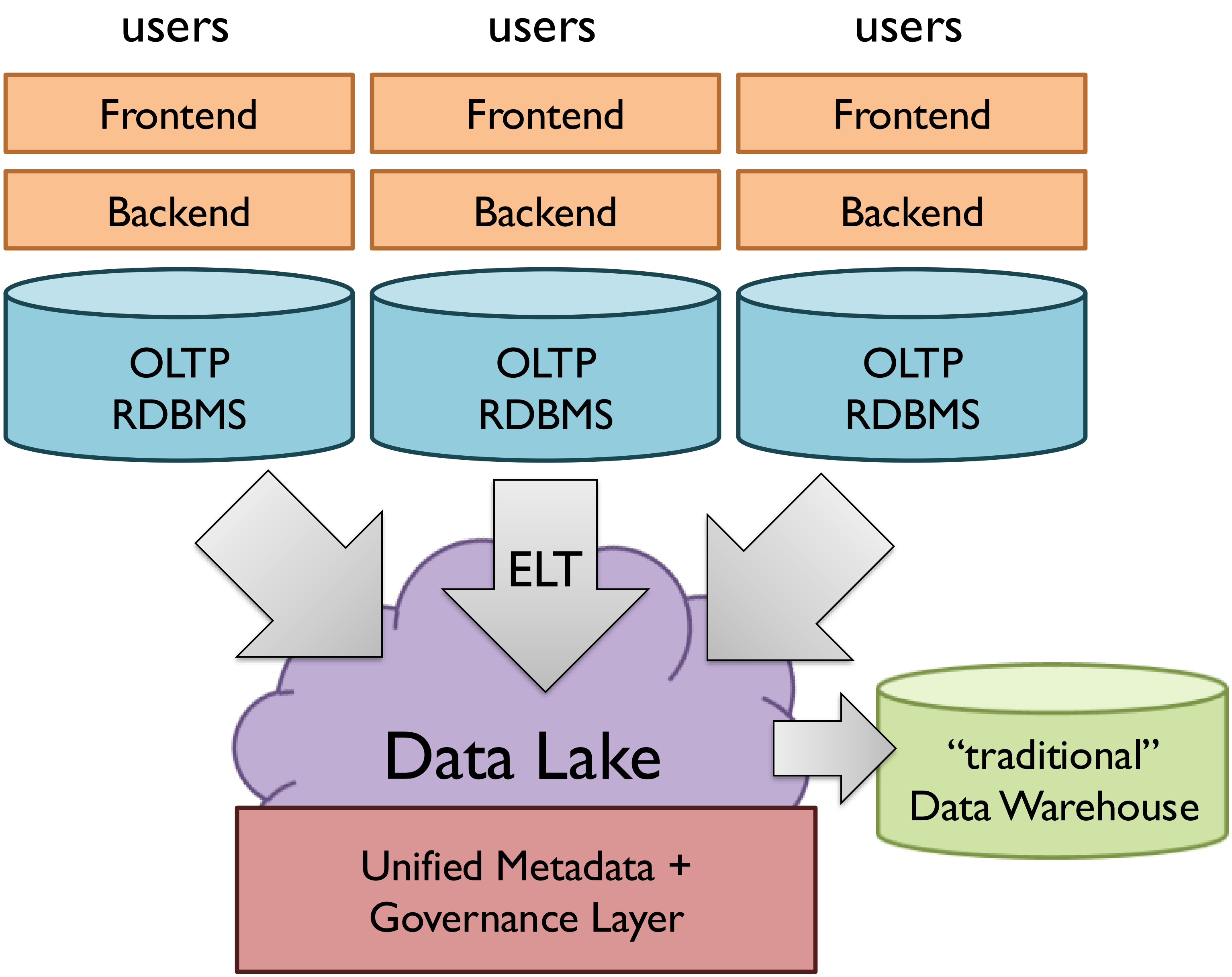
Cheap (object) storage ✓

vs. expensive file systems

Flexible query processing ✓

vs. only SQL

Sometimes, I really just want a RDBMS!



users

Frontend

Backend

OLTP
RDBMS

users

Frontend

Backend

OLTP
RDBMS

users

Frontend

Backend

OLTP
RDBMS

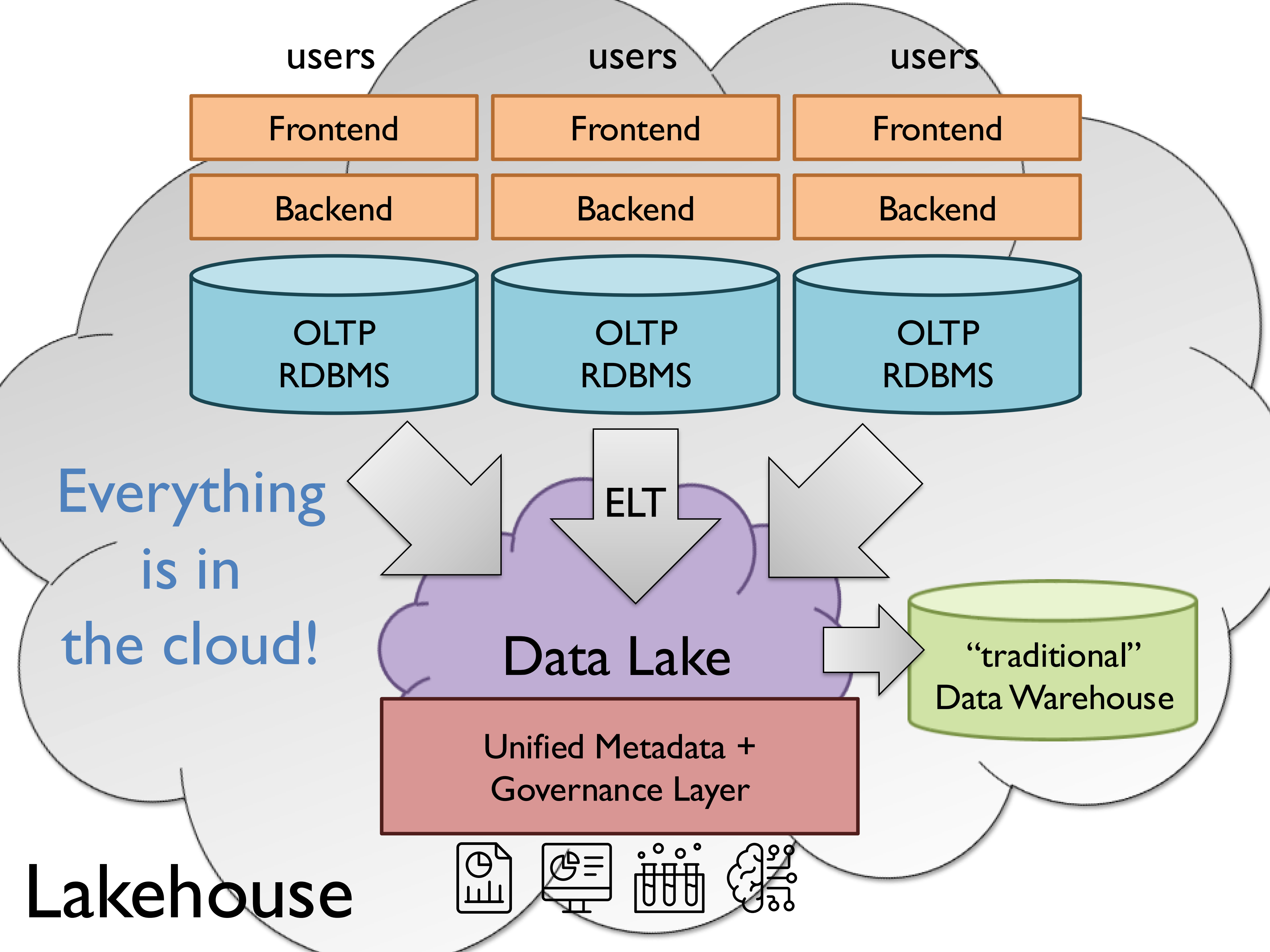
ELT

Data Lake

“traditional”
Data Warehouse

Unified Metadata +
Governance Layer

Lakehouse



users

users

users

Frontend

Frontend

Frontend

Backend

Backend

Backend

OLTP RDBMS

OLTP RDBMS

OLTP RDBMS

ELT

Data Lake

Unified Metadata + Governance Layer

“traditional” Data Warehouse

Everything is in the cloud!

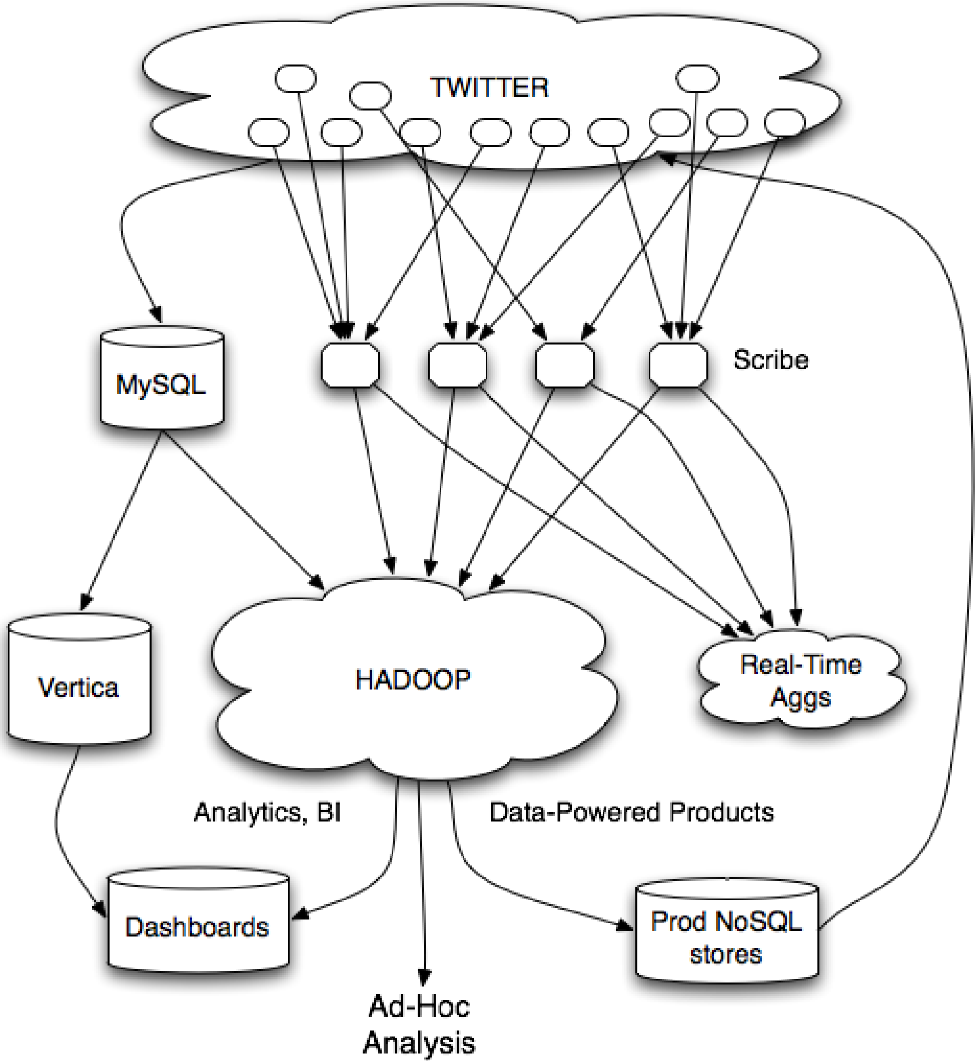
Lakehouse

The Cloud

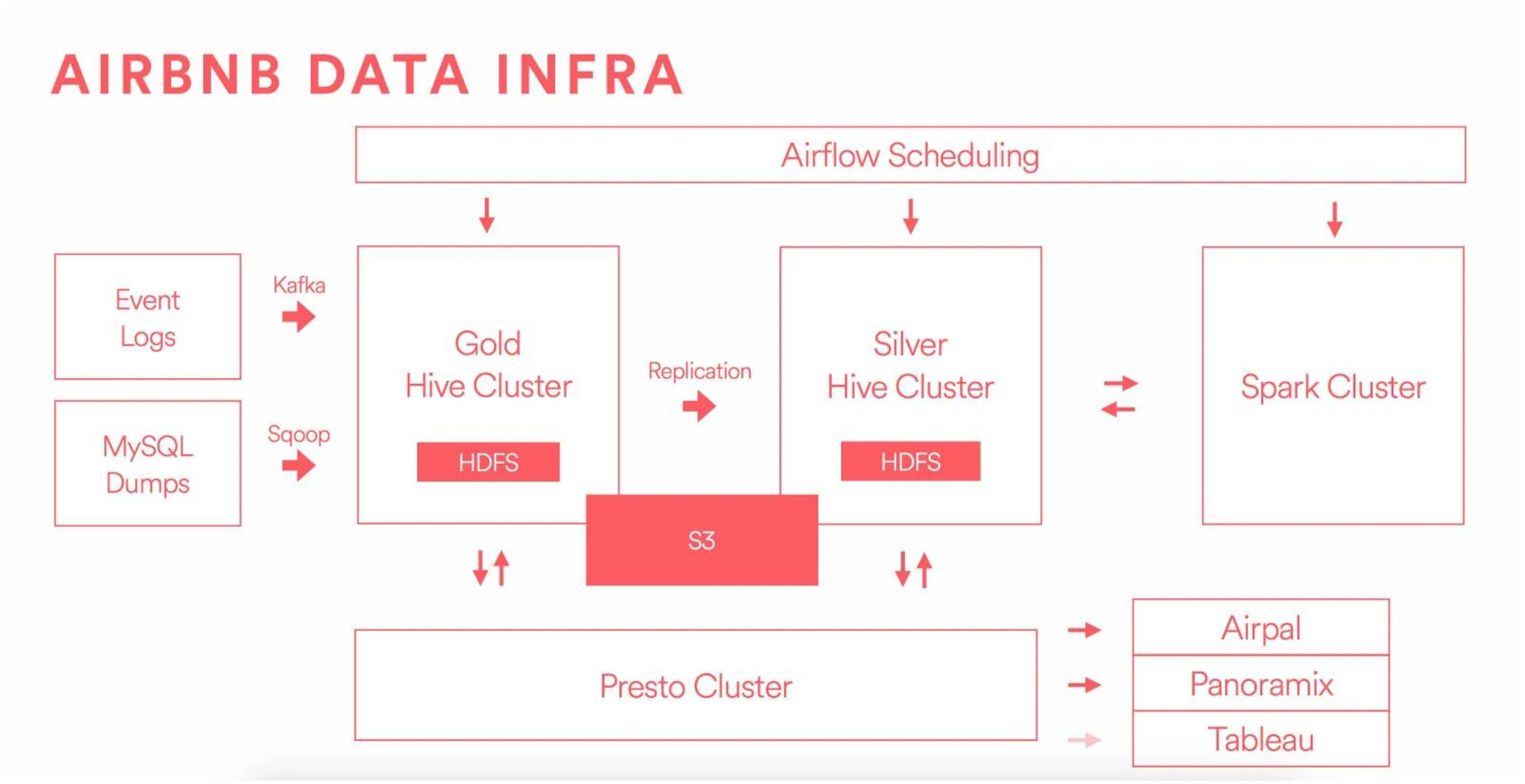
✓ Pro – you don’t have to worry about it.

Con – you don’t have to worry about it. ✗

(More on this the coming weeks…)

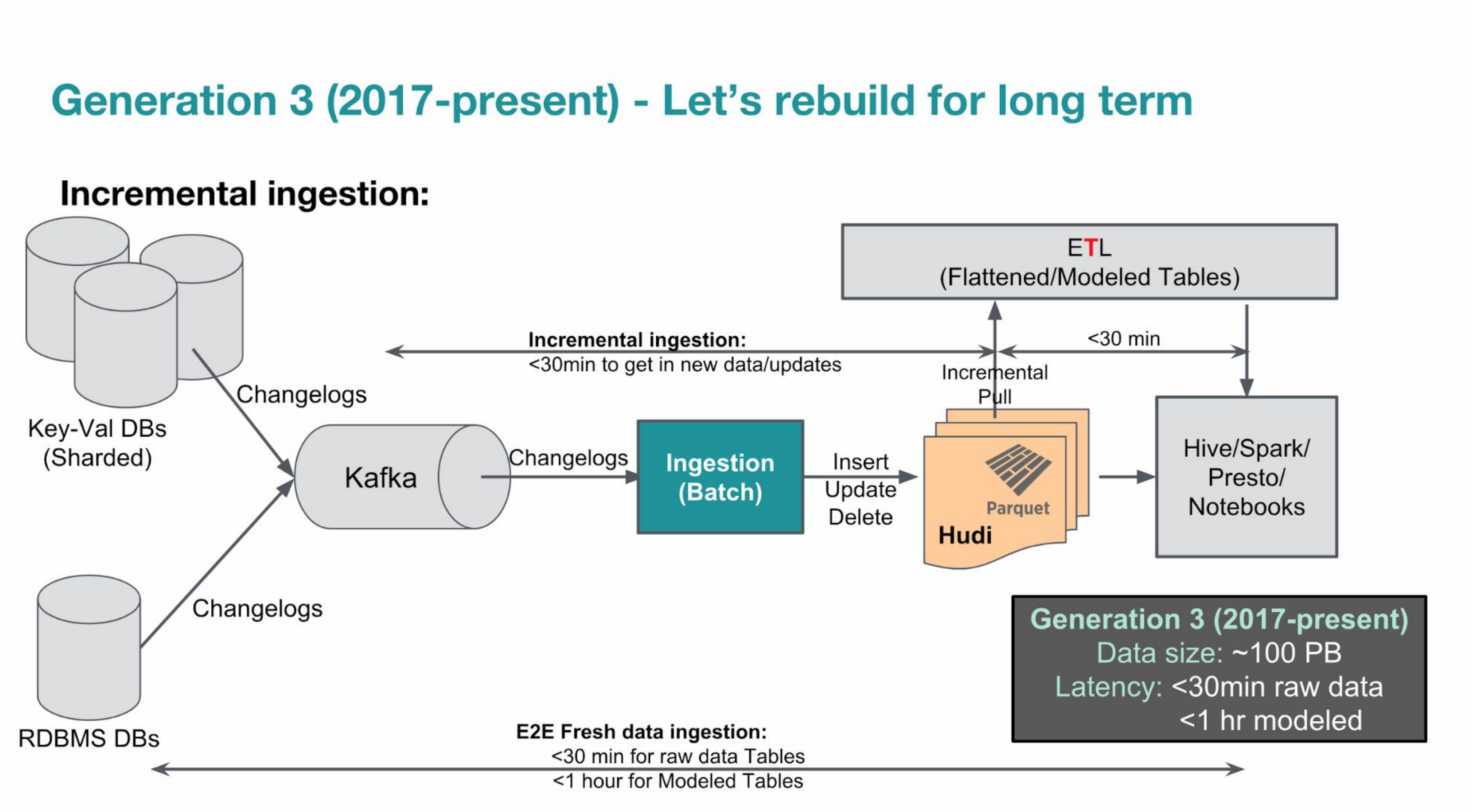


Twitter's data platform (circa 2012)



AirBnB's data platform (circa 2016)

https://medium.com/airbnb-engineering/data-infrastructure-at-airbnb-8adfb34f169c



Generation 3 (2017-present) - Let's rebuild for long term

Incremental ingestion:

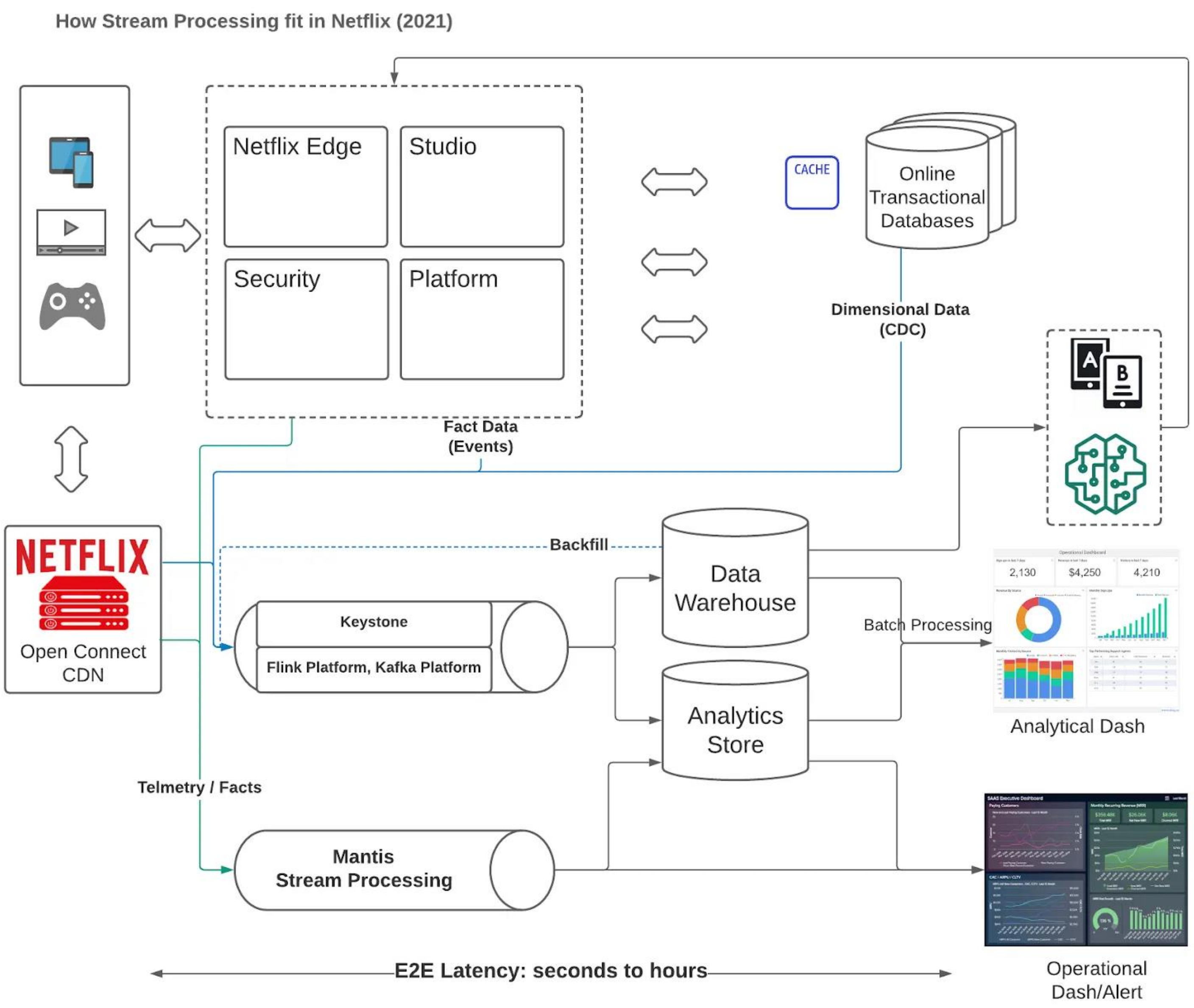
Incremental ingestion:
<30min to get in new data/updates

Generation 3 (2017-present)
Data size: ~100 PB
Latency: <30min raw data
<1 hr modeled

E2E Fresh data ingestion:
<30 min for raw data Tables
<1 hour for Modeled Tables

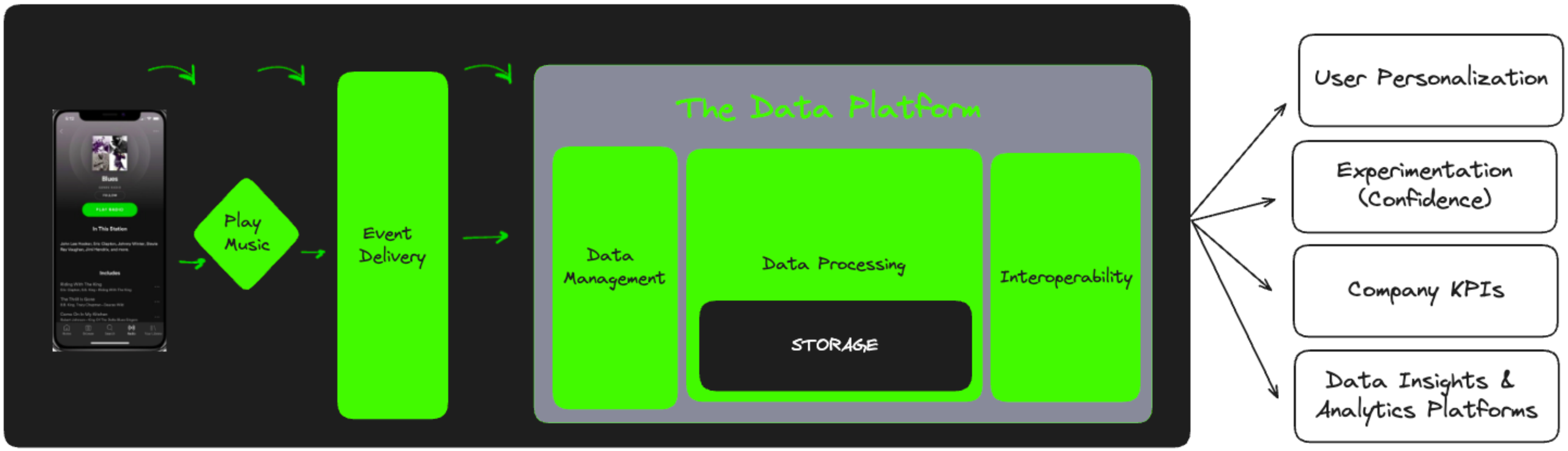
Uber's data platform (circa 2018)

`https://www.uber.com/en-CA/blog/uber-big-data-platform/`



Netflix's data platform (circa 2021)

https://zhenzhongxu.com/the-four-innovation-phases-of-netflixs-trillions-scale-real-time-data-infrastructure-2370938d7f01



Spotify’s data platform (circa 2024)

https://engineering.atspotify.com/2024/5/data-platform-explained-part-ii

冨嶽三十六景
神奈川沖
浪裏