UNIVERSITY OF WATERLOO

Data-Intensive Distributed Computing

CS 451/651 431/631 (Winter 2018)

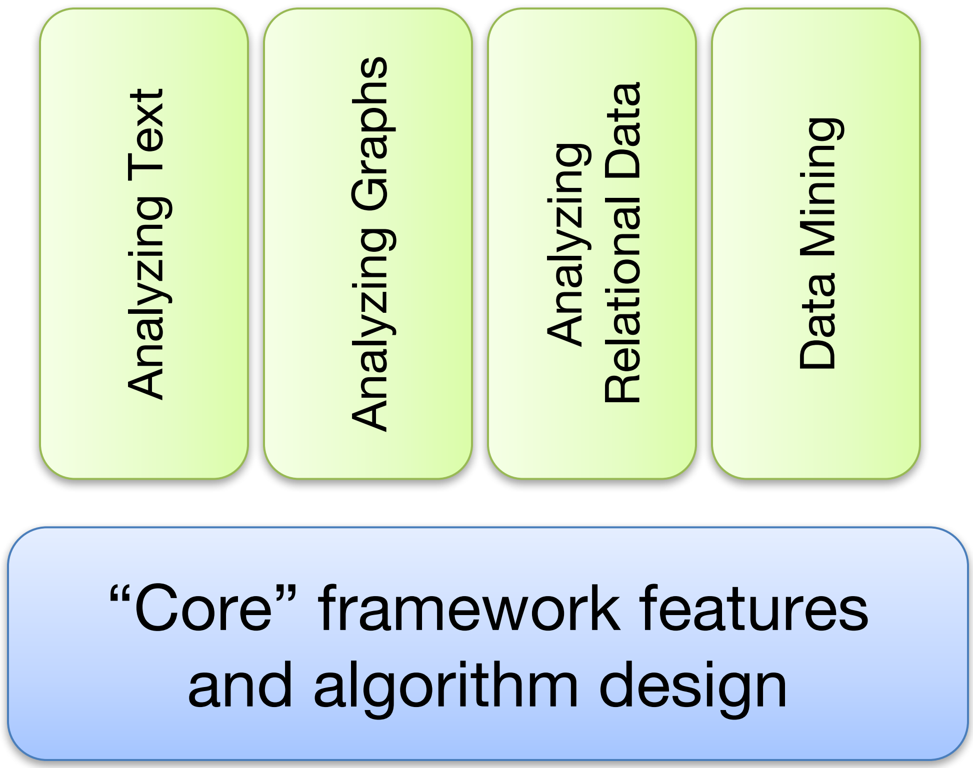
Part 5: Analyzing Relational Data (1/3)

February 8, 2018

Jimmy Lin
David R. Cheriton School of Computer Science
University of Waterloo

These slides are available at http://lintool.github.io/bigdata-2018w/

Structure of the Course



Evolution of Enterprise Architectures

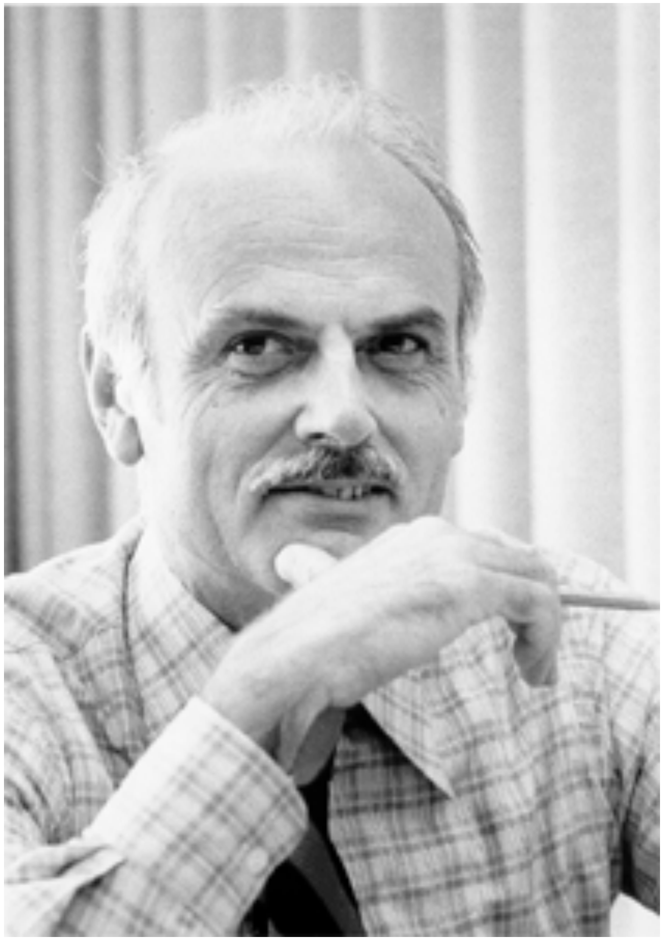
Next two sessions: techniques, algorithms, and optimizations for relational processing users

Monolithic Application

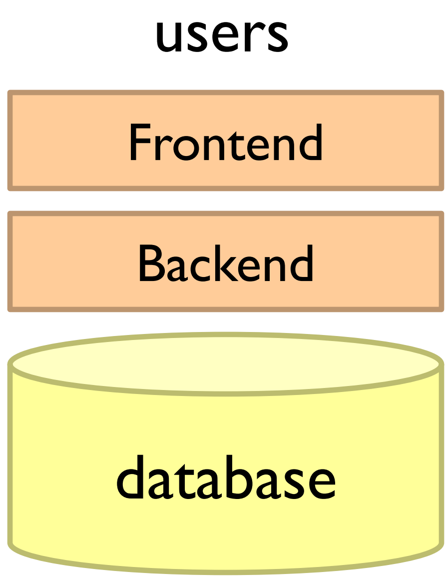
users

Frontend

Backend



Source: Wikipedia

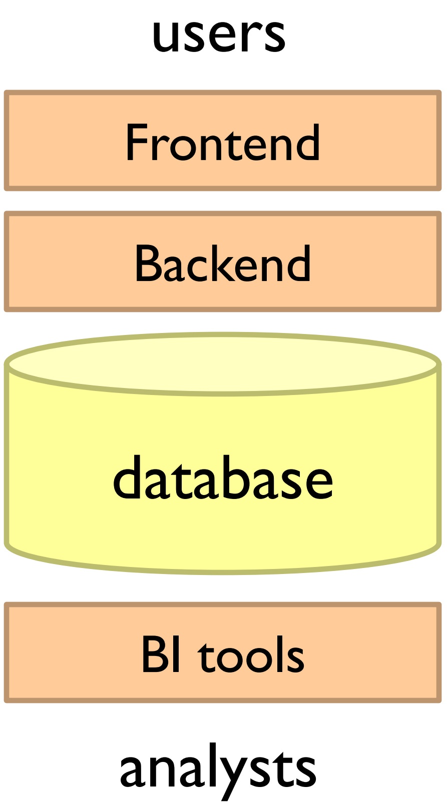


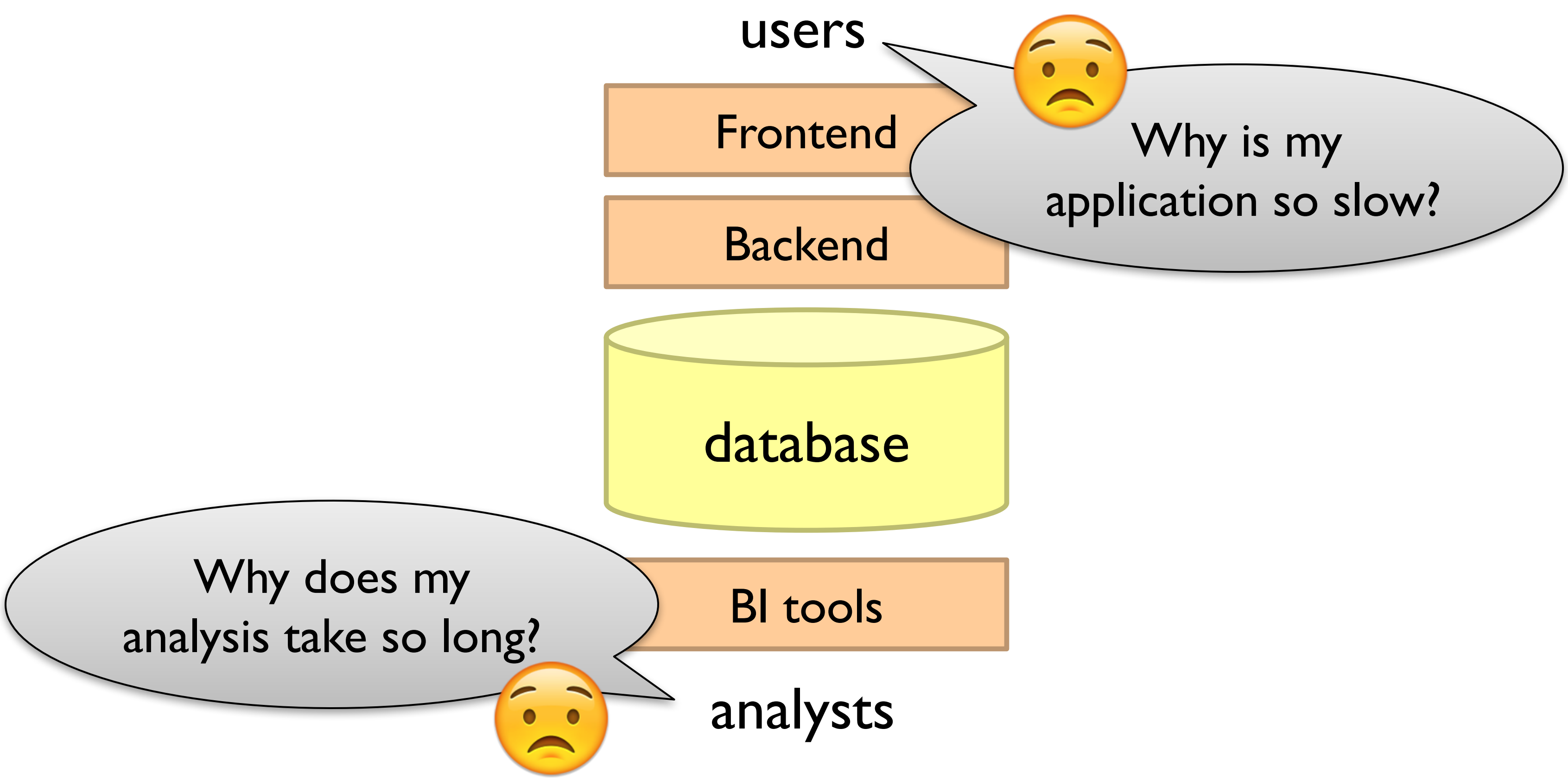
Why is this a good idea?

Business Intelligence

An organization should retain data that result from carrying out its mission and exploit those data to generate insights that benefit the organization, for example, market analysis, strategic planning, decision making, etc.

Duh!?


users
Frontend
Backend
database
BI tools
analysts


users
Frontend
Backend
database
BI tools
analysts
Why is my
application so slow?
Why does my
analysis take so long?

Database Workloads

OLTP (online transaction processing)

Typical applications: e-commerce, banking, airline reservations
User facing: real-time, low latency, highly-concurrent
Tasks: relatively small set of "standard" transactional queries
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

Poor memory management

Conflicting data access patterns

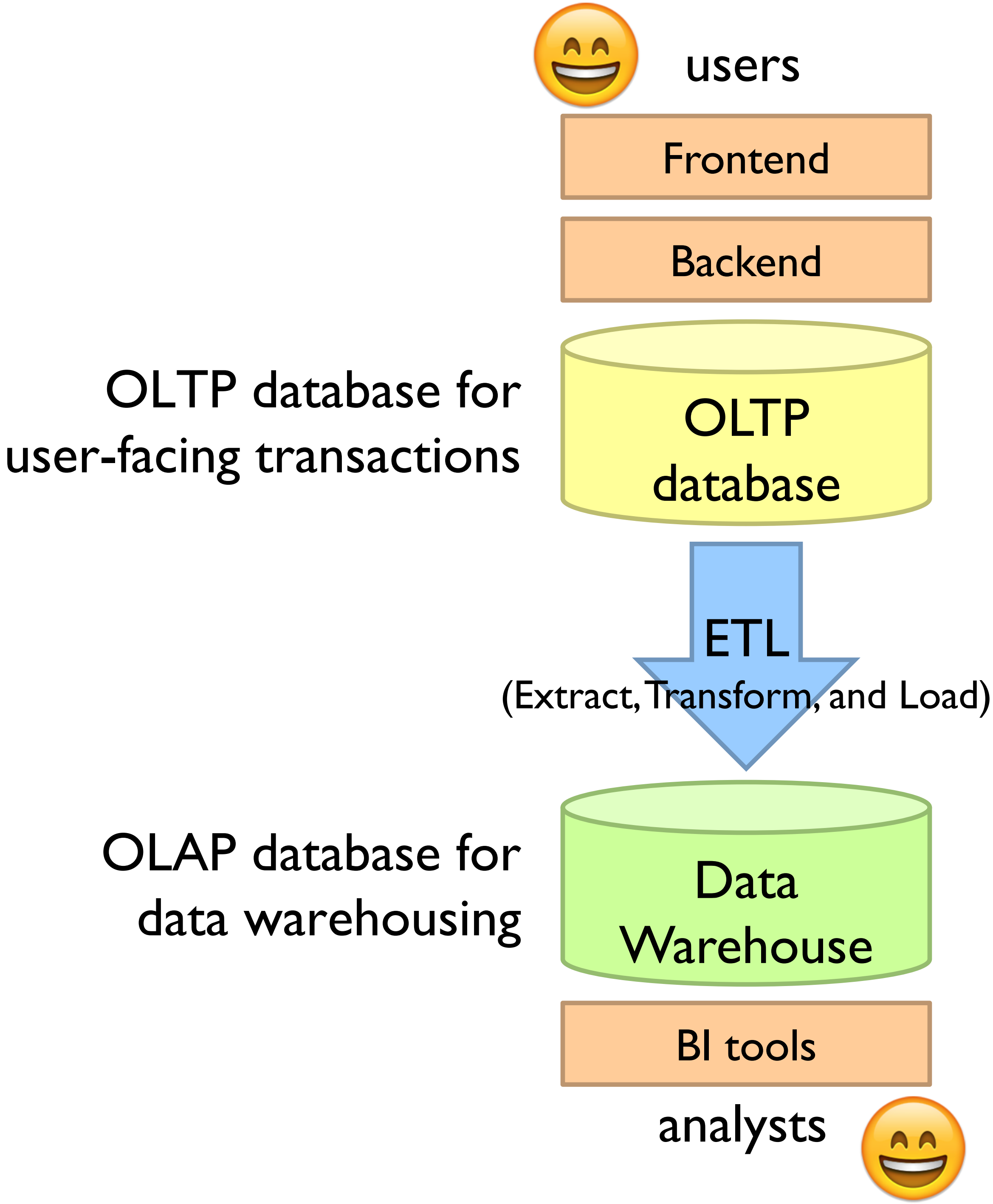
Variable latency

😟 users and analysts

Solution?

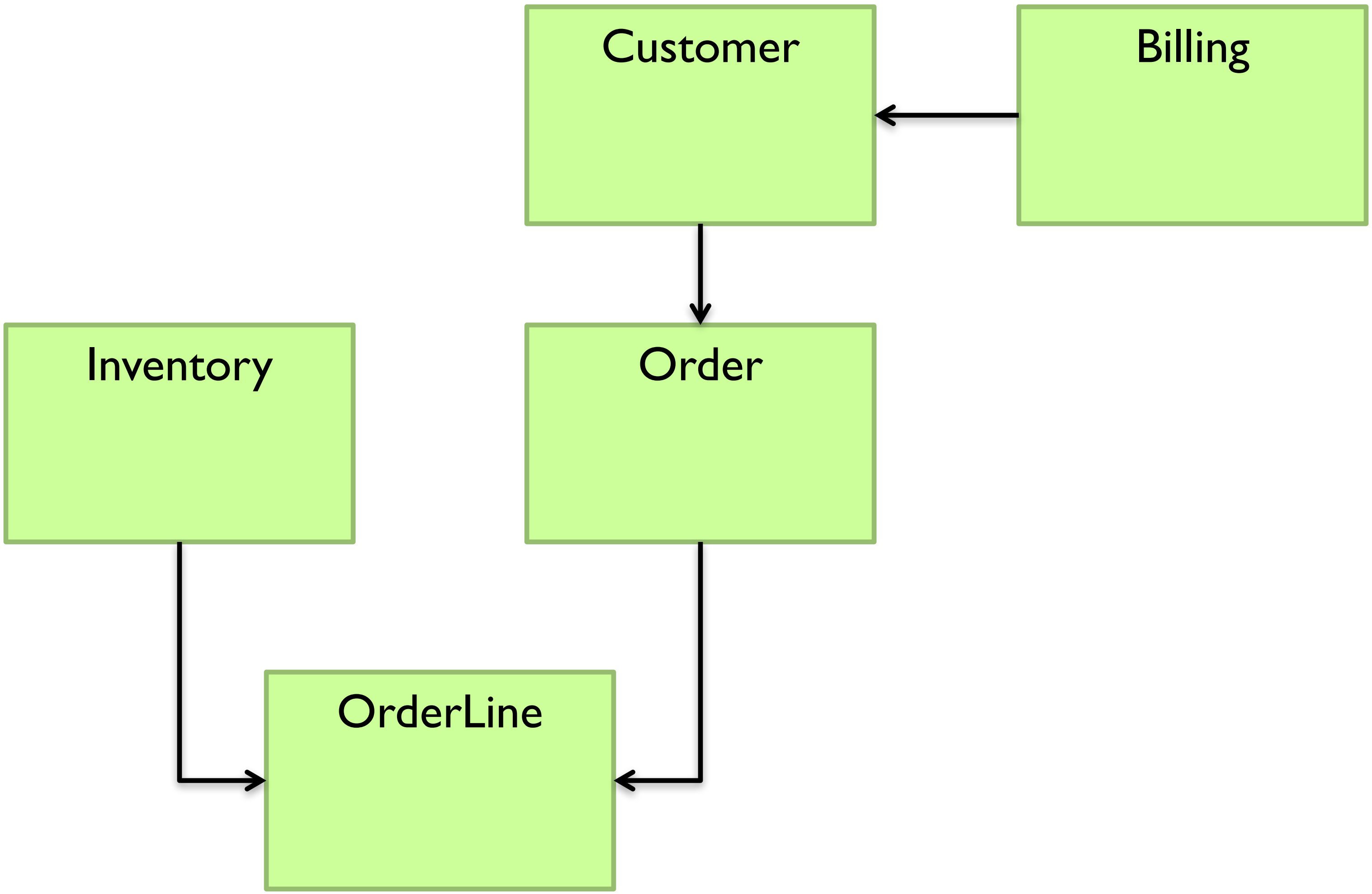
Build a data warehouse!

Source: Wikipedia (Warehouse)

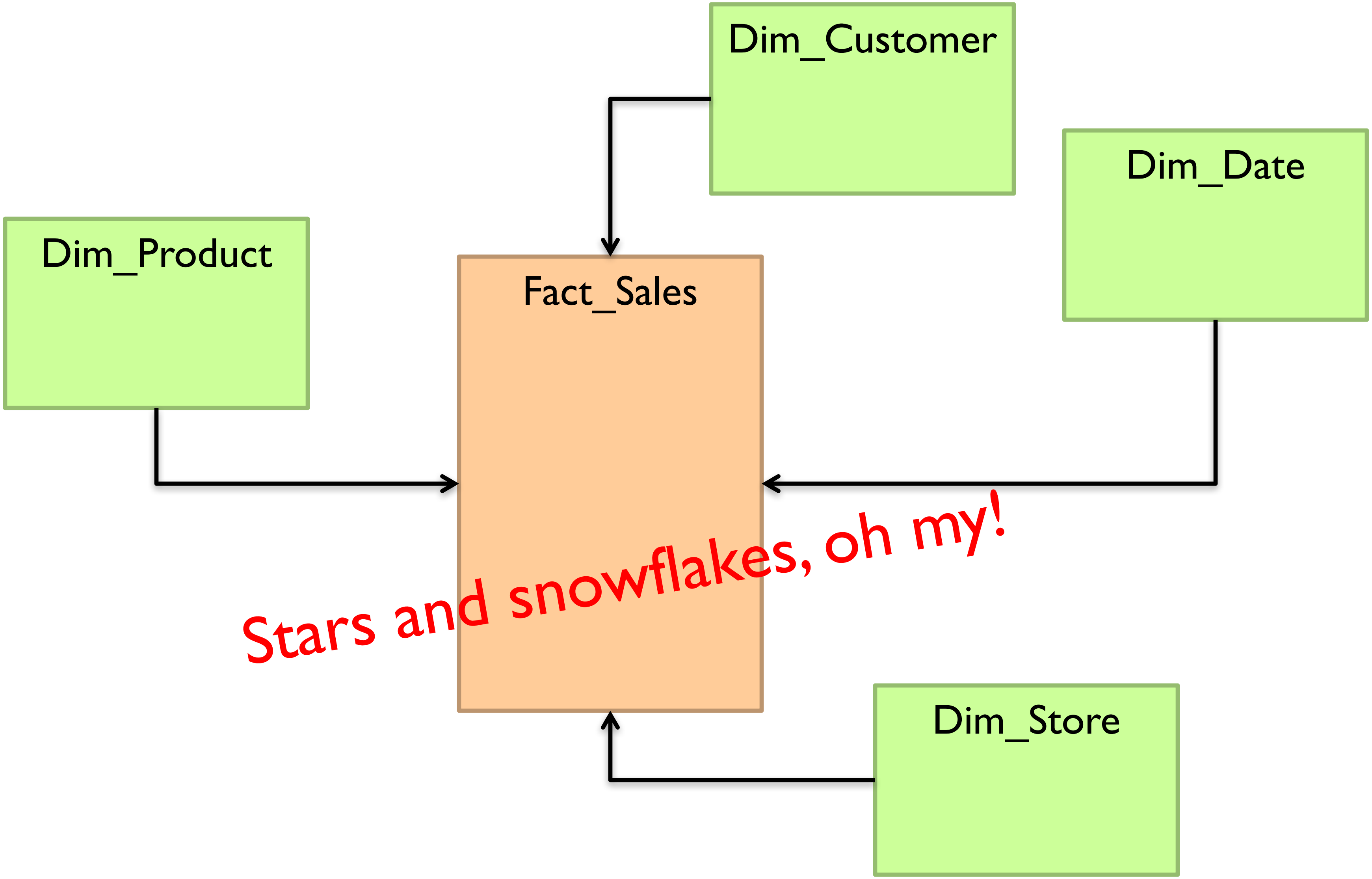


What's special about
OLTP vs. OLAP?

A Simple OLTP Schema



A Simple OLAP Schema



ETL

Extract

Transform

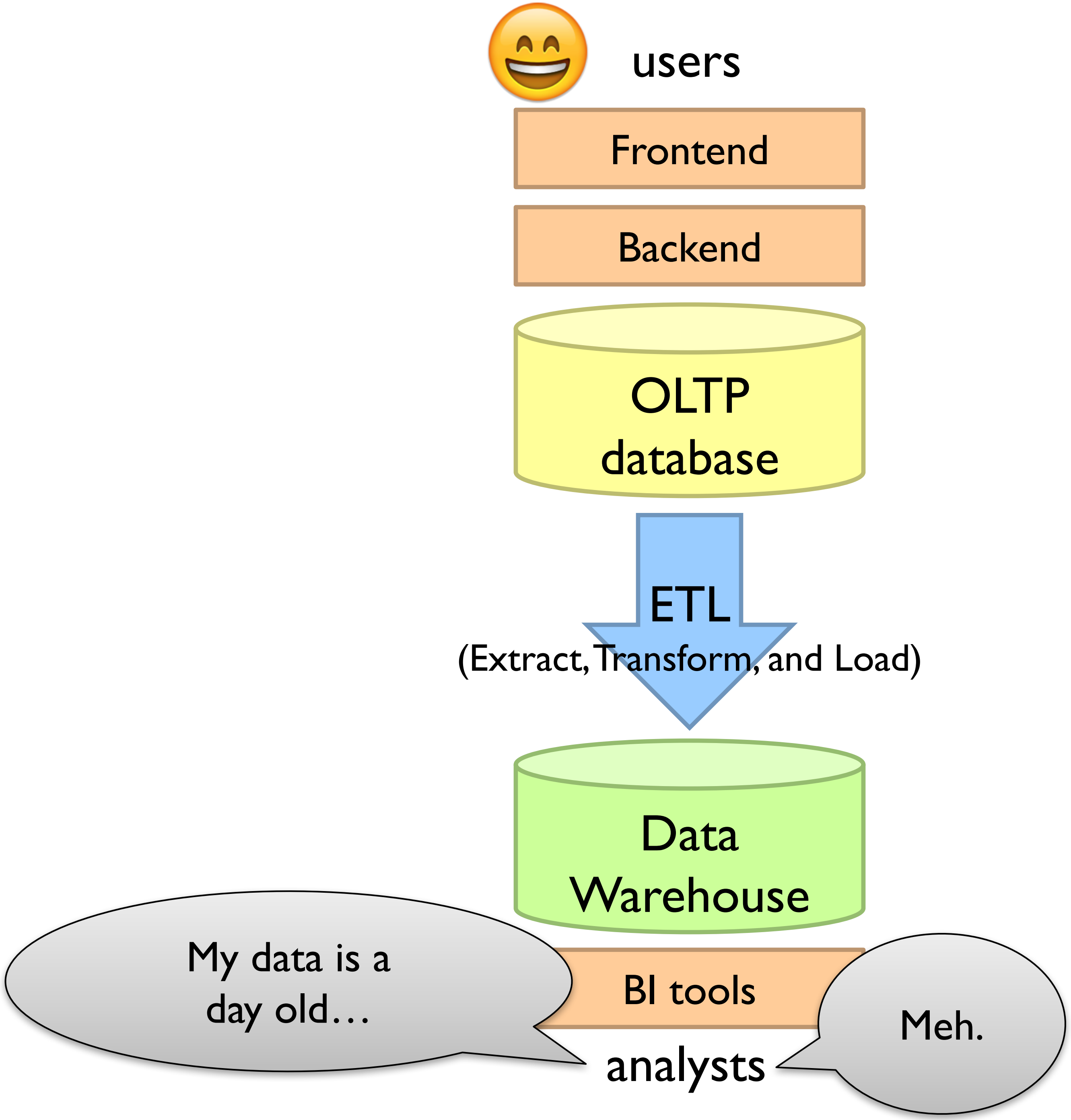
Data cleaning and integrity checking

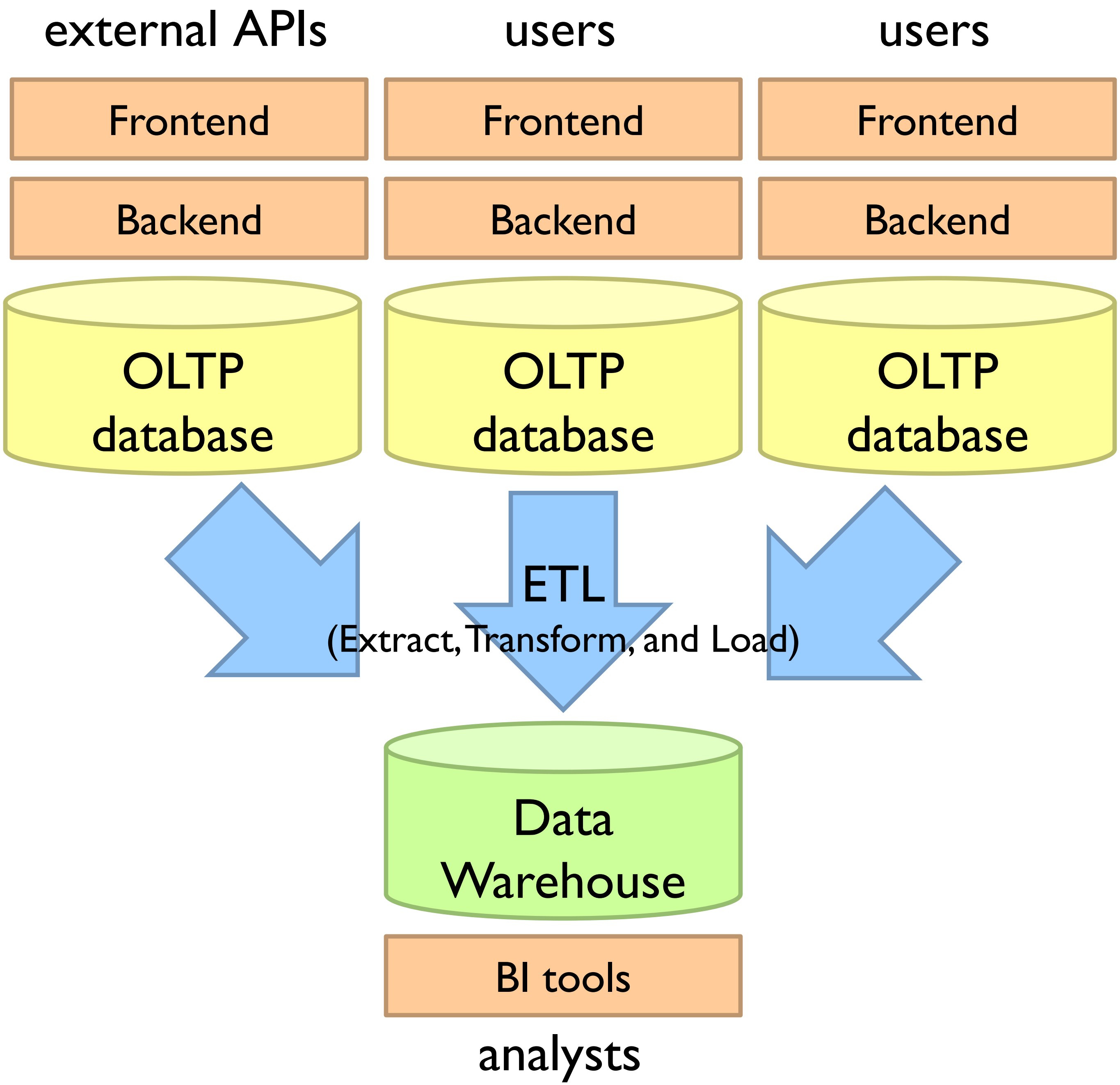
Schema conversion

Field transformations

Load

When does ETL happen?


users
Frontend
Backend
OLTP database
ETL
(Extract, Transform, and Load)
Data Warehouse
BI tools
analysts
My data is a day old…
Meh.


external APIs
users
users
Frontend
Frontend
Frontend
Backend
Backend
Backend
OLTP database
OLTP database
OLTP database
ETL
(Extract, Transform, and Load)
Data Warehouse
BI tools
analysts

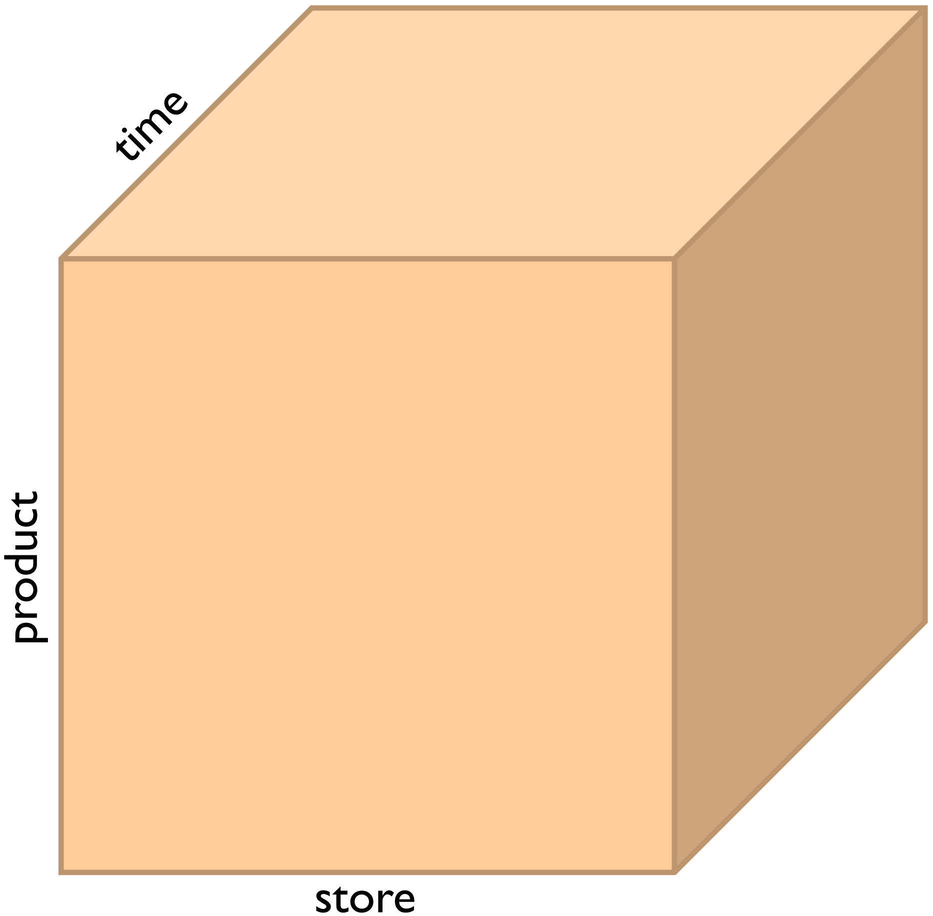
What do you actually do?

Report generation

Dashboards

Ad hoc analyses

OLAP Cubes



Common operations

- slice and dice
- roll up/drill down
- pivot

OLAP Cubes: Challenges

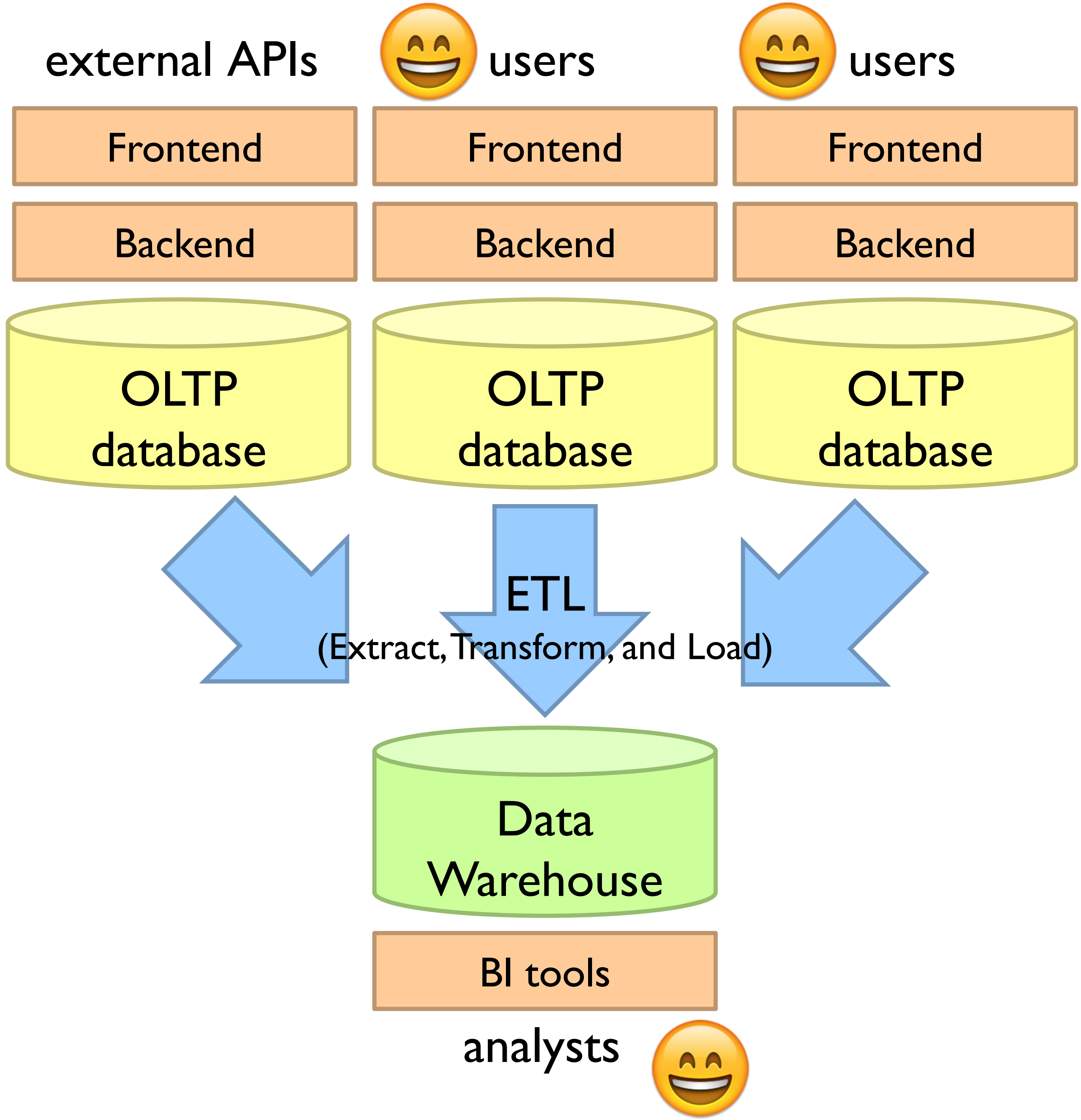
Fundamentally, lots of joins, group-bys and aggregations

How to take advantage of schema structure to avoid repeated work?

Cube materialization

Realistic to materialize the entire cube?

If not, how/when/what to materialize?


external APIs
users
users
Frontend
Frontend
Frontend
Backend
Backend
Backend
OLTP database
OLTP database
OLTP database
ETL
(Extract, Transform, and Load)
Data Warehouse
BI tools
analysts

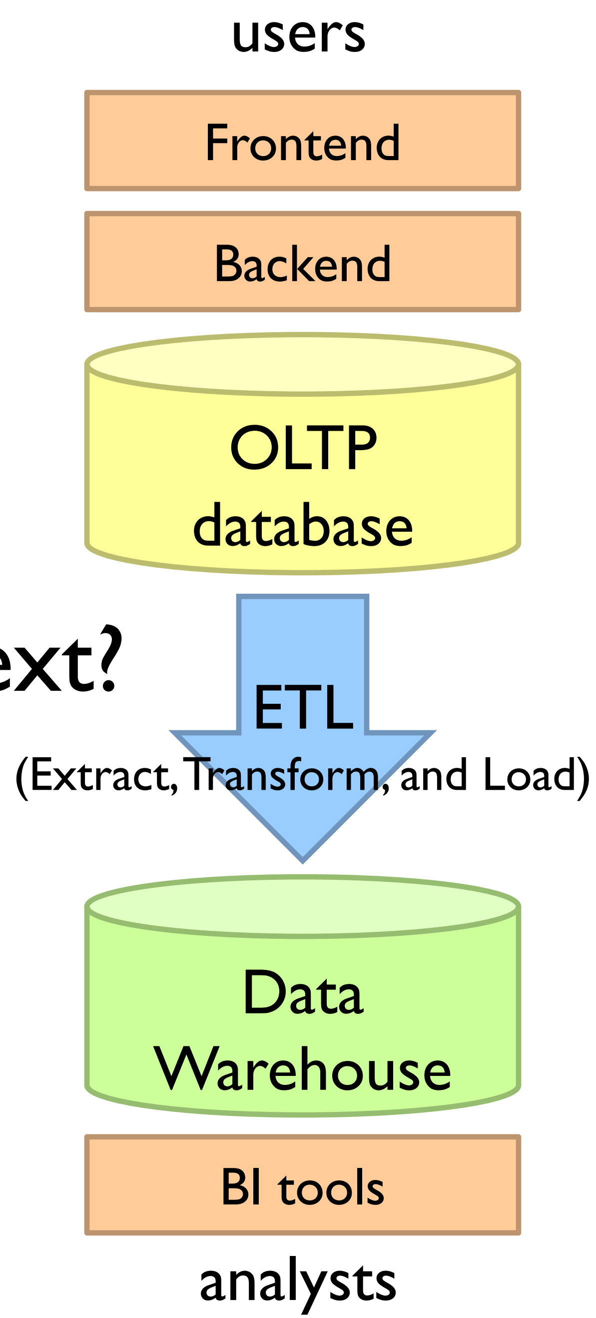
Fast forward…

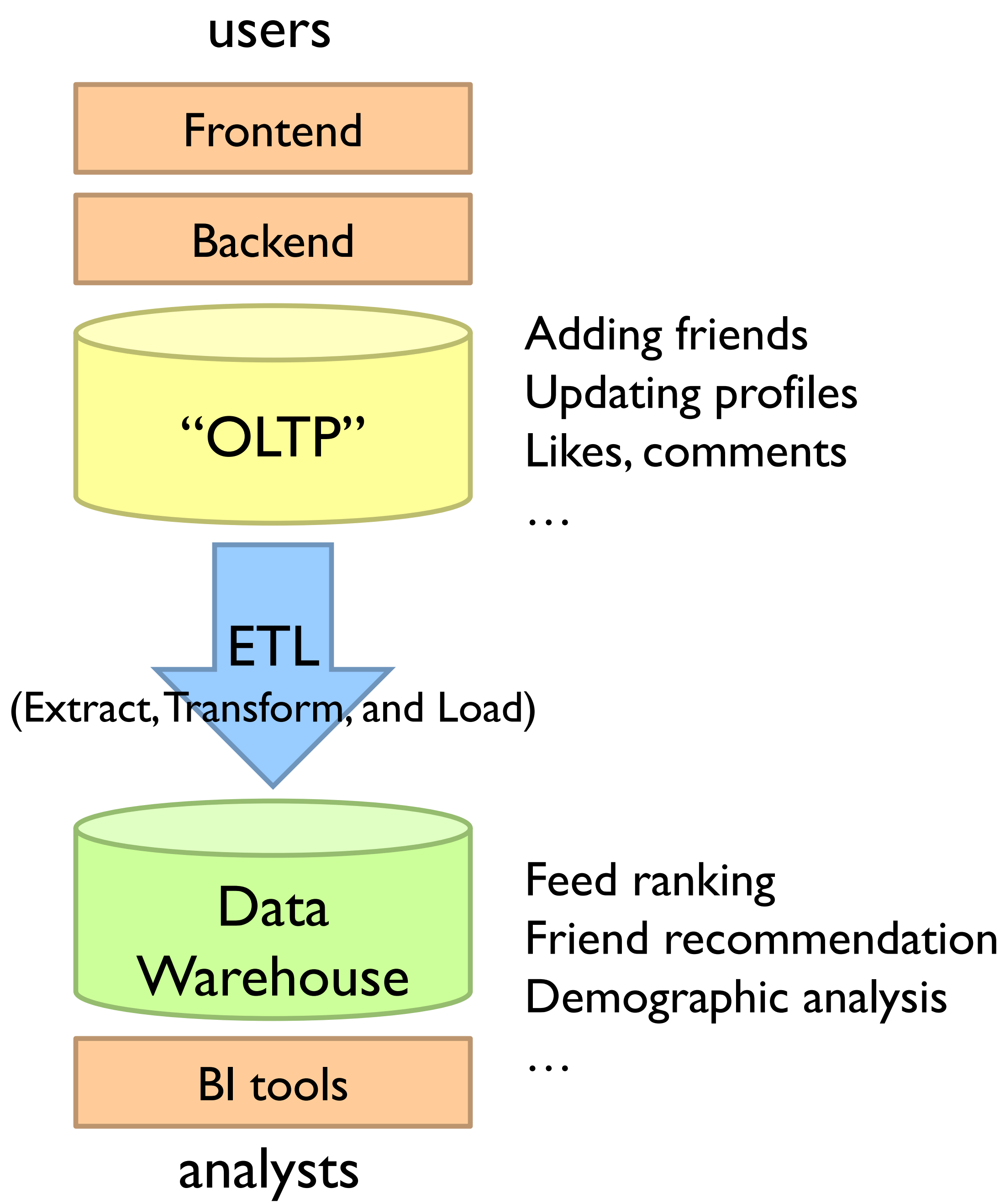
facebook®

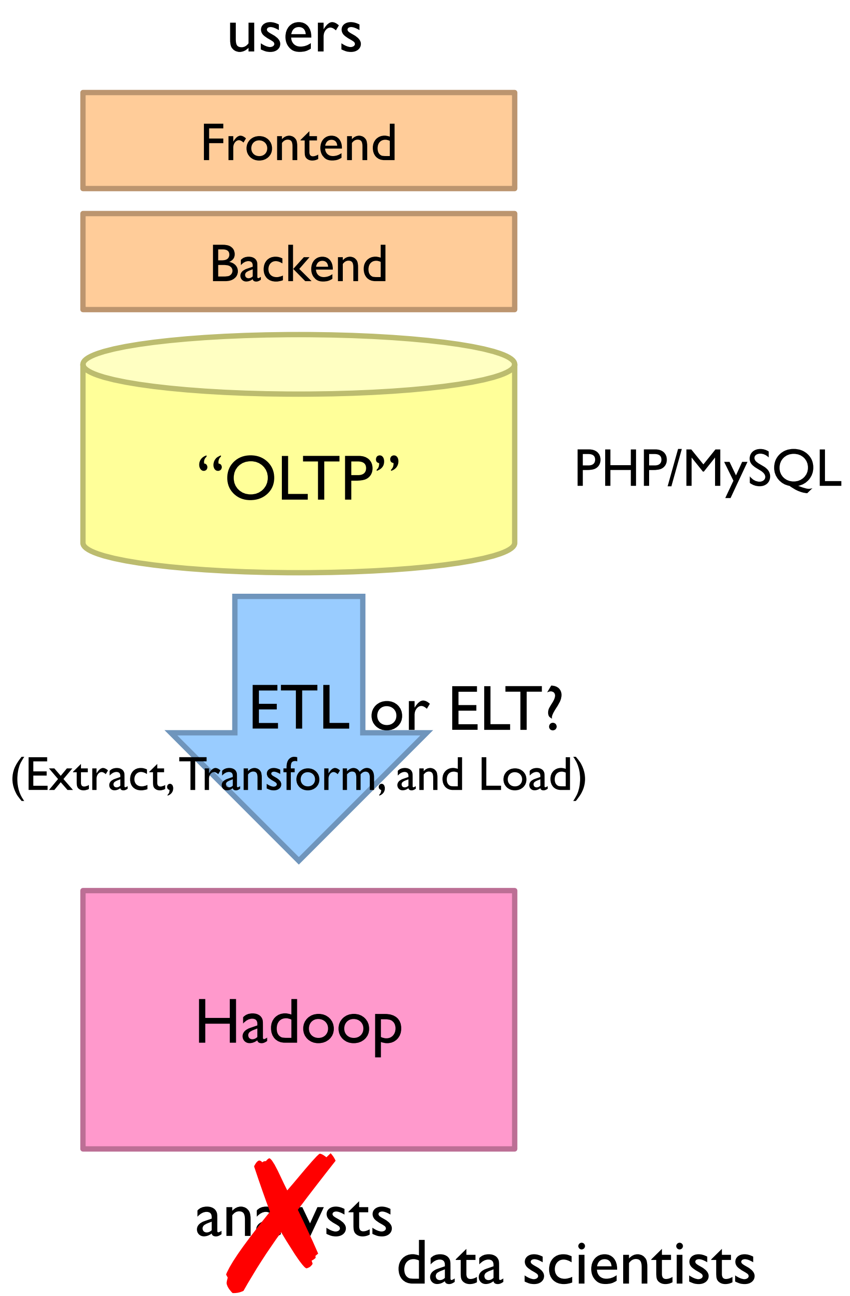
Jeff Hammerbacher, Information Platforms and the Rise of the Data Scientist.
In, *Beautiful Data*, O'Reilly, 2009.

"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."

Facebook context?




users
Frontend
Backend
"OLTP"
Adding friends
Updating profiles
Likes, comments
…
ETL
(Extract, Transform, and Load)
Data Warehouse
Feed ranking
Friend recommendation
Demographic analysis
…
BI tools
analysts


users
Frontend
Backend
"OLTP"
PHP/MySQL
ETL or ELT?
(Extract, Transform, and Load)
Hadoop
~~analysts~~
data scientists

What'

Droppi

Cheaper to store everything

5 MB hard drive in 1956

What's changed?

Dropping cost of disks

Cheaper to store everything than to figure out what to throw away

Types of data collected

From data that's *obviously* valuable to data whose value is less apparent

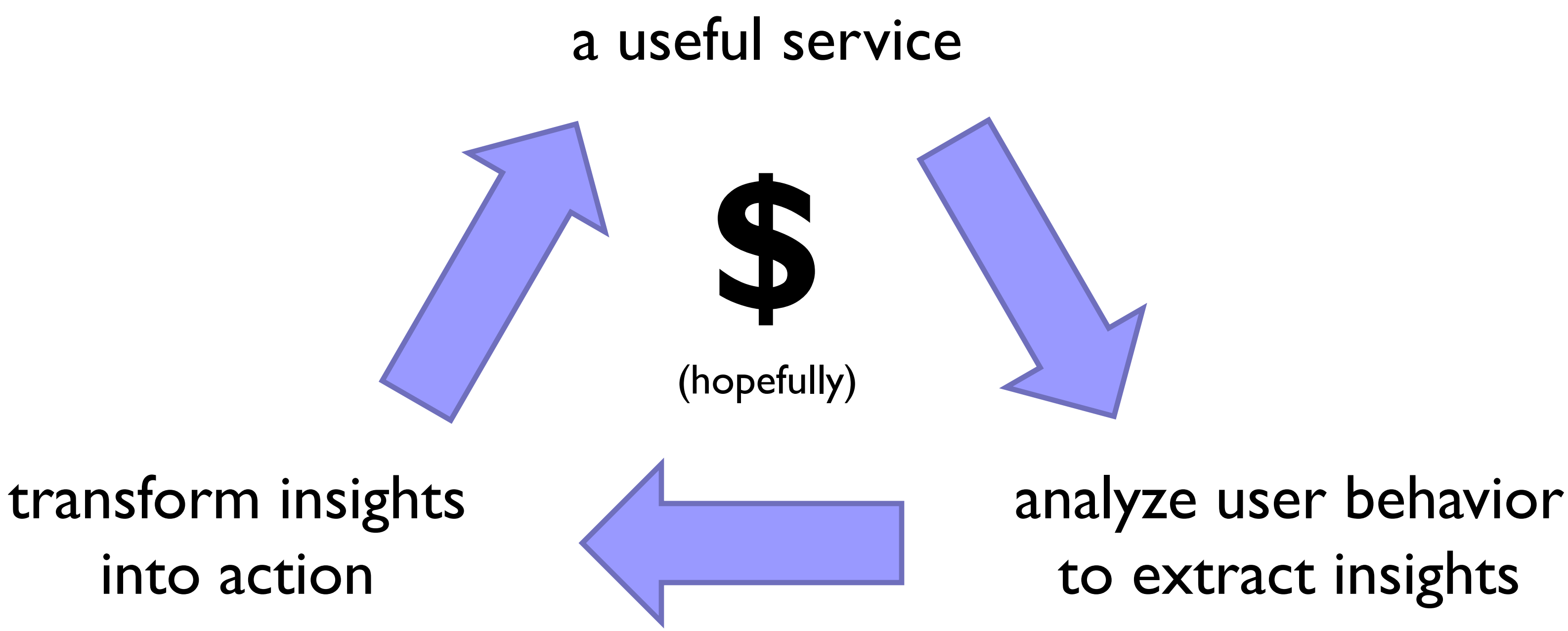
Rise of social media and user-generated content

Large increase in data volume

Growing maturity of data mining techniques

Demonstrates value of data analytics

Virtuous Product Cycle



Google. Facebook. Twitter. Amazon. Uber.

What do you actually do?

Report generation

Dashboards

Ad hoc analyses

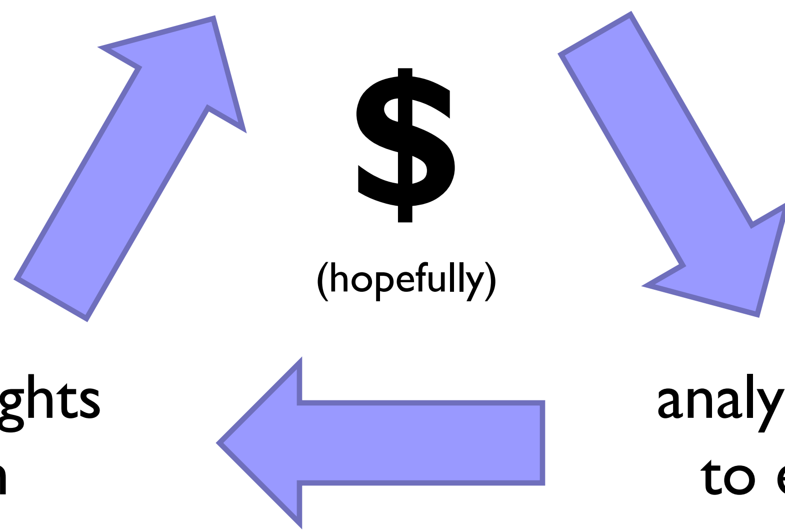
“Descriptive”

“Predictive”

Data products

Virtuous Product Cycle

a useful service



transform insights into action

analyze user behavior to extract insights

Google. Facebook. Twitter. Amazon. Uber.

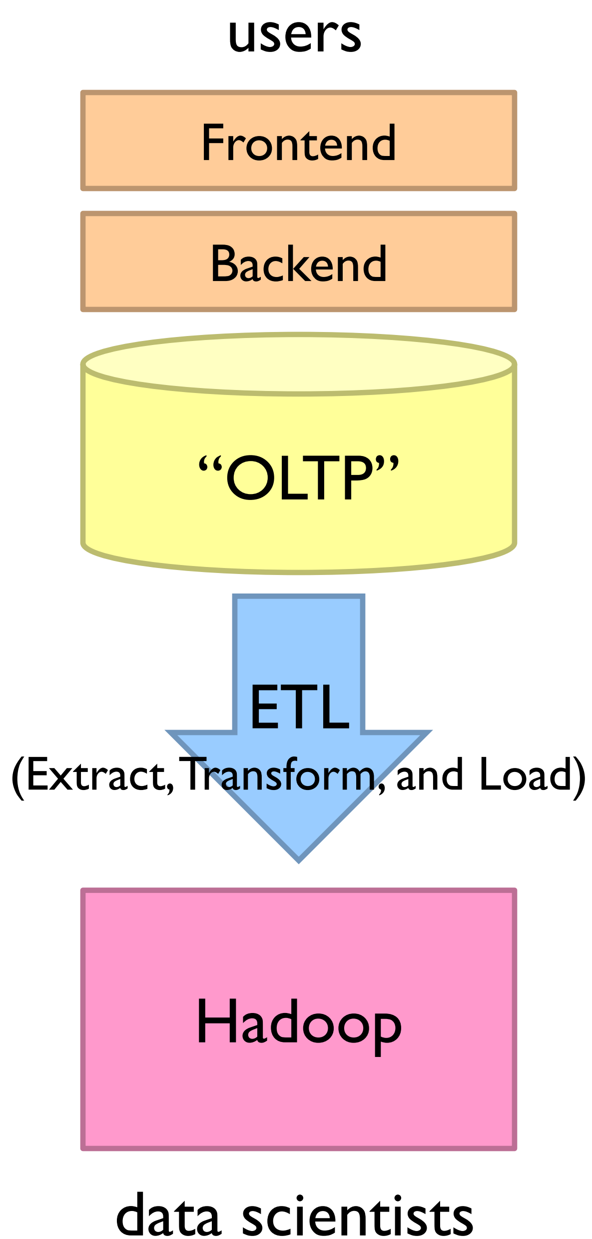
data products

data science

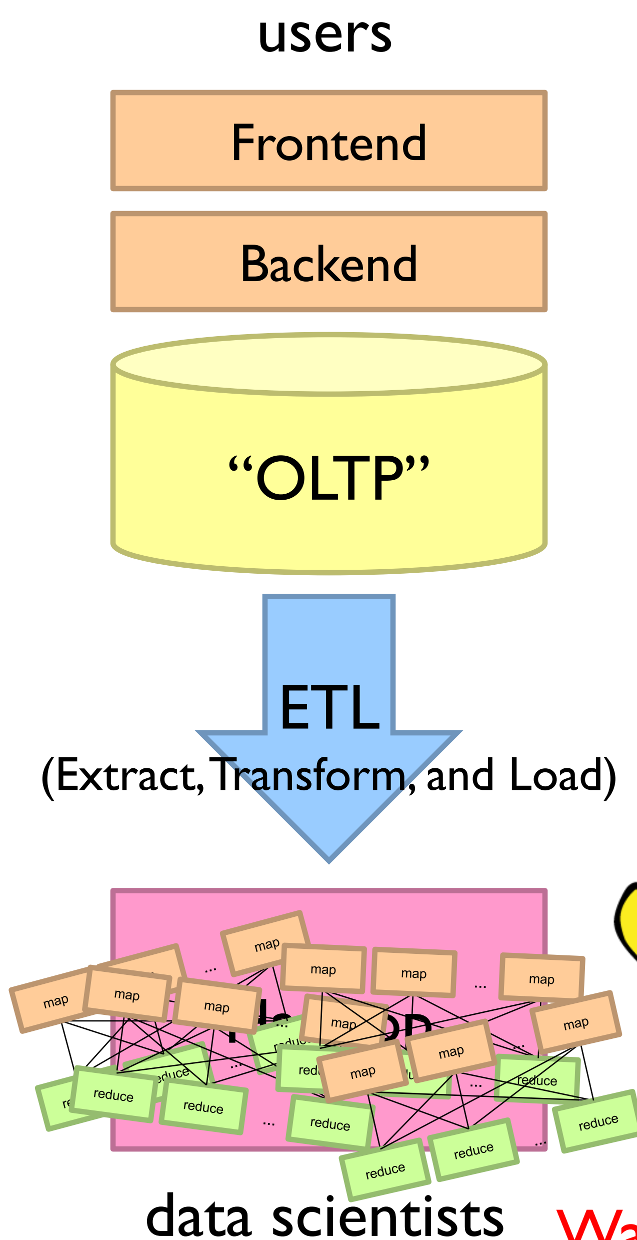
facebook®

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."


users
Frontend
Backend
“OLTP”
ETL
(Extract, Transform, and Load)
Hadoop
data scientists

The Irony…



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

Why not just use a database?

SQL is awesome

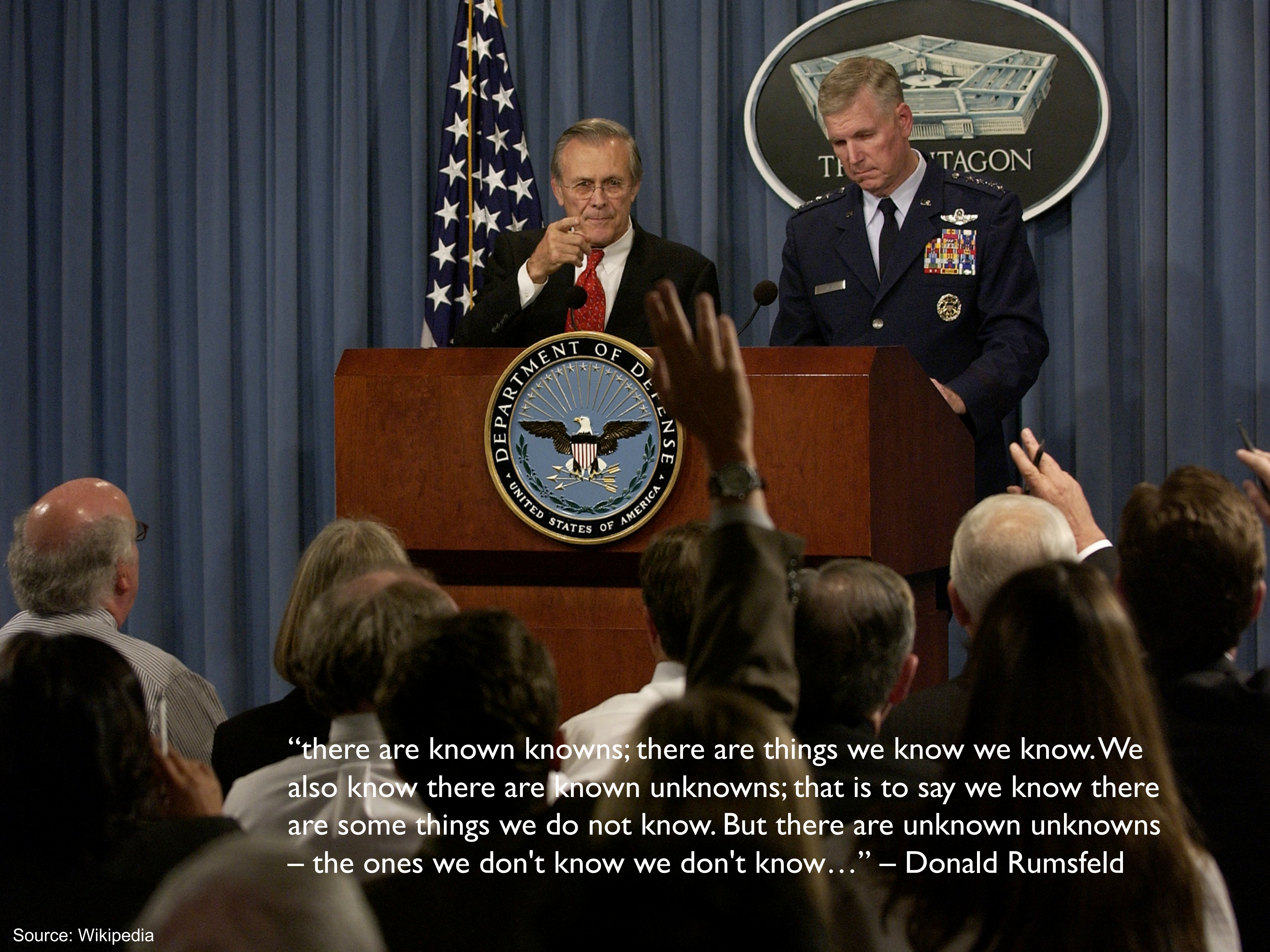
Scalability. Cost.

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!

Advantages of Hadoop dataflow languages

Don't need to know the schema ahead of time

Raw scans are the most common operations

Many analyses are better formulated imperatively

Much faster data ingest rate

What do you actually do?

Report generation

Dashboards

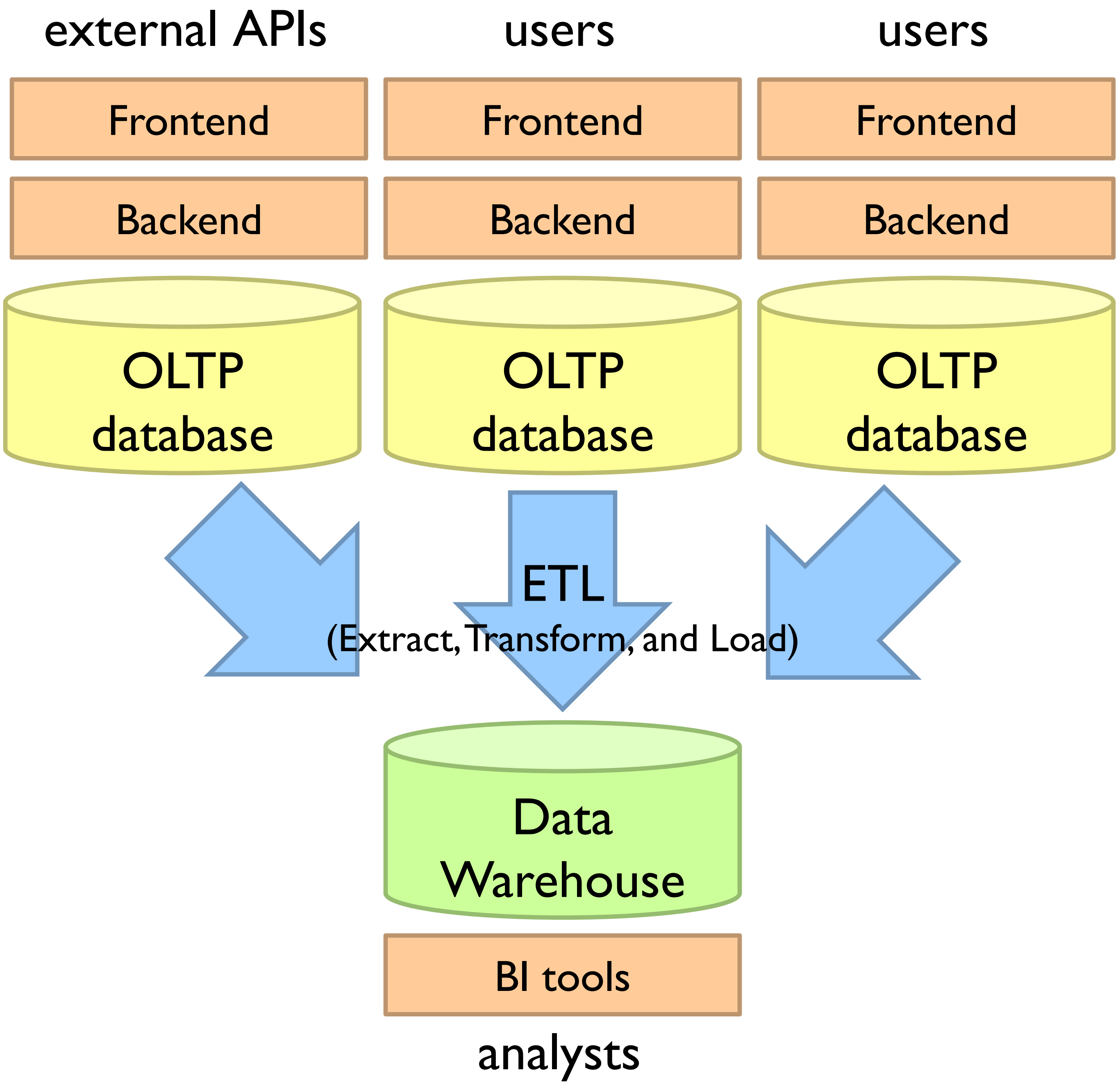
Ad hoc analyses

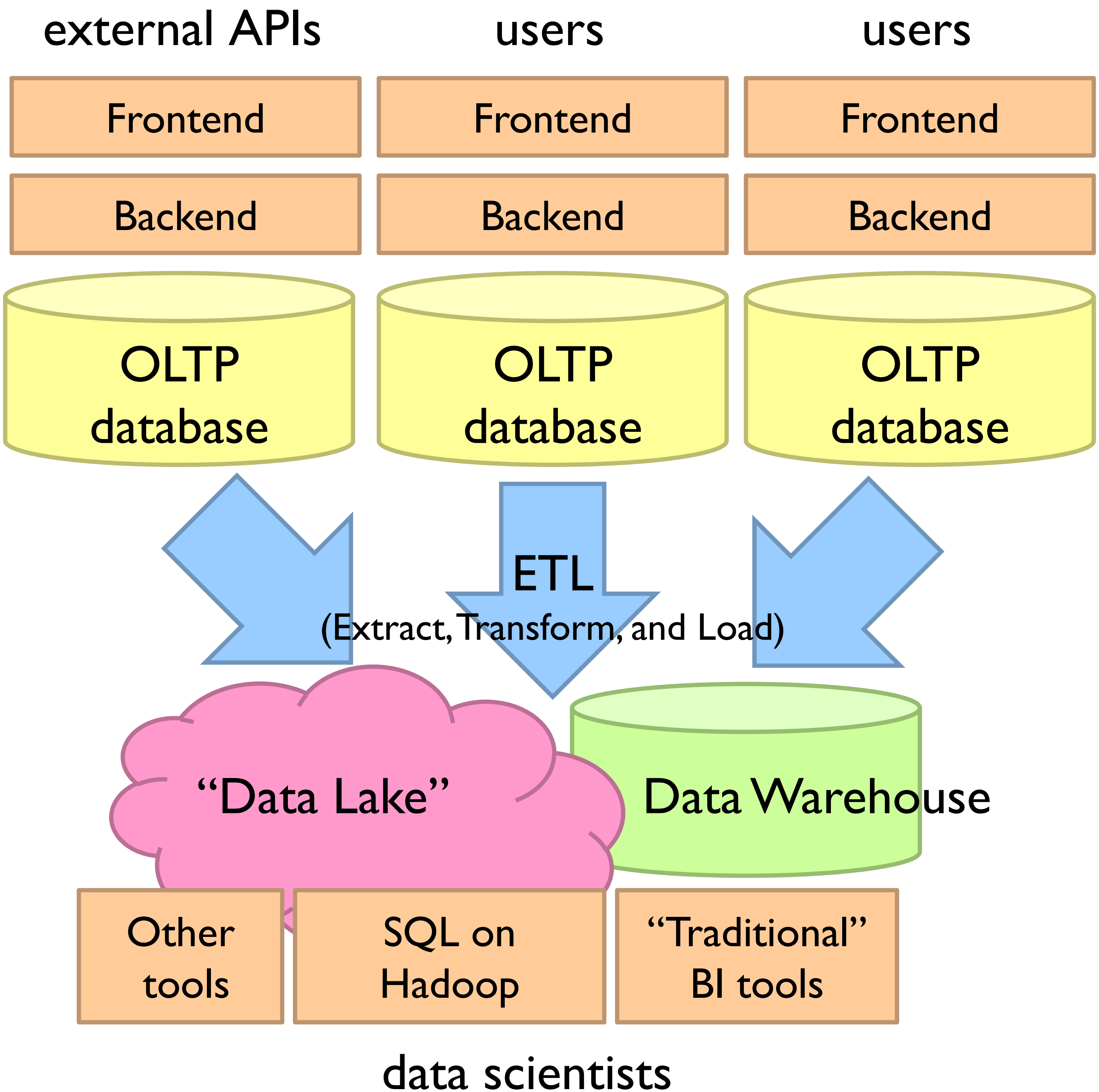
“Descriptive”

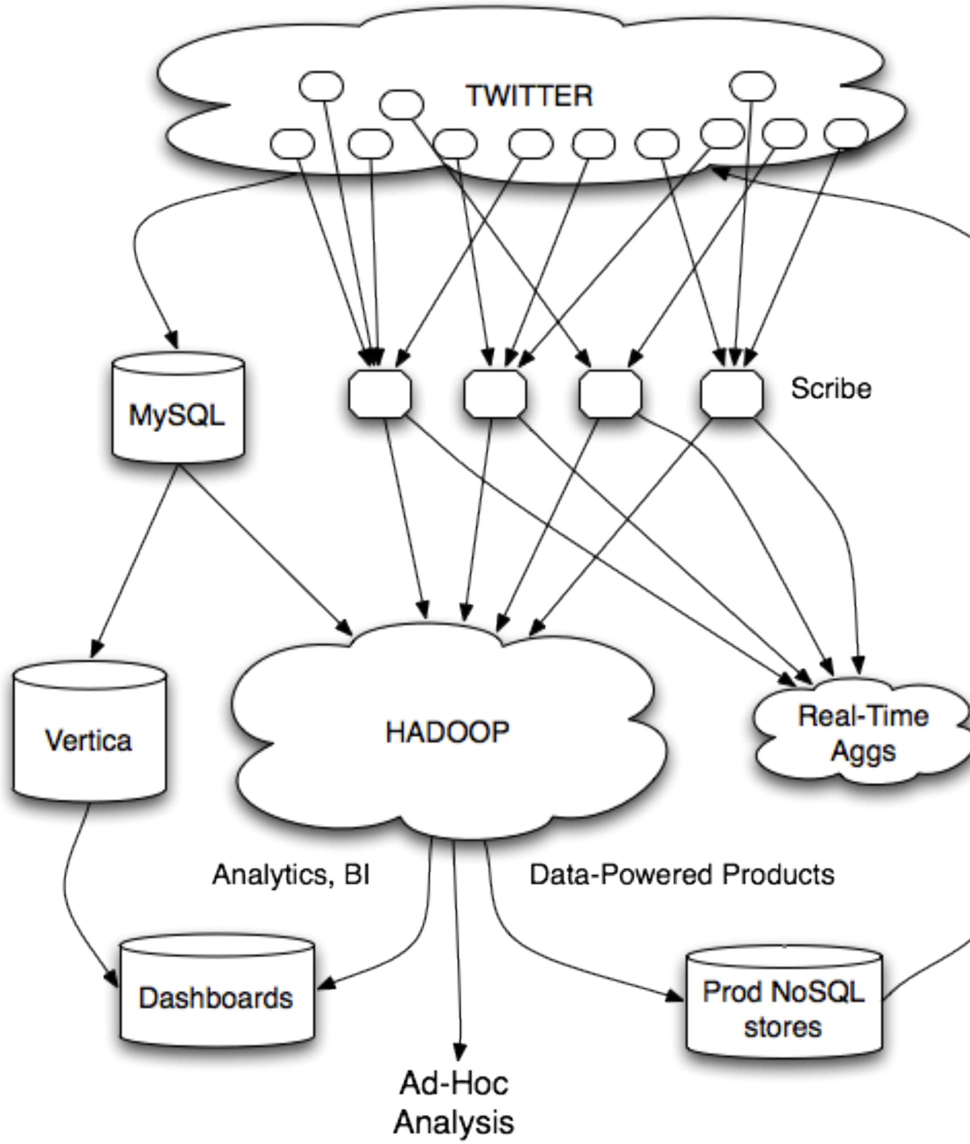
“Predictive”

Data products

Which are known unknowns and unknown unknowns?


external APIs
users
users
Frontend
Frontend
Frontend
Backend
Backend
Backend
OLTP database
OLTP database
OLTP database
ETL
(Extract, Transform, and Load)
Data Warehouse
BI tools
analysts


external APIs
users
users
Frontend
Frontend
Frontend
Backend
Backend
Backend
OLTP database
OLTP database
OLTP database
ETL
(Extract, Transform, and Load)
“Data Lake”
Data Warehouse
Other tools
SQL on Hadoop
“Traditional” BI tools
data scientists



Twitter’s data warehousing architecture (circa 2012)

circa ~2010

~150 people total

~60 Hadoop nodes

~6 people use analytics stack daily

circa ~2012

~1400 people total

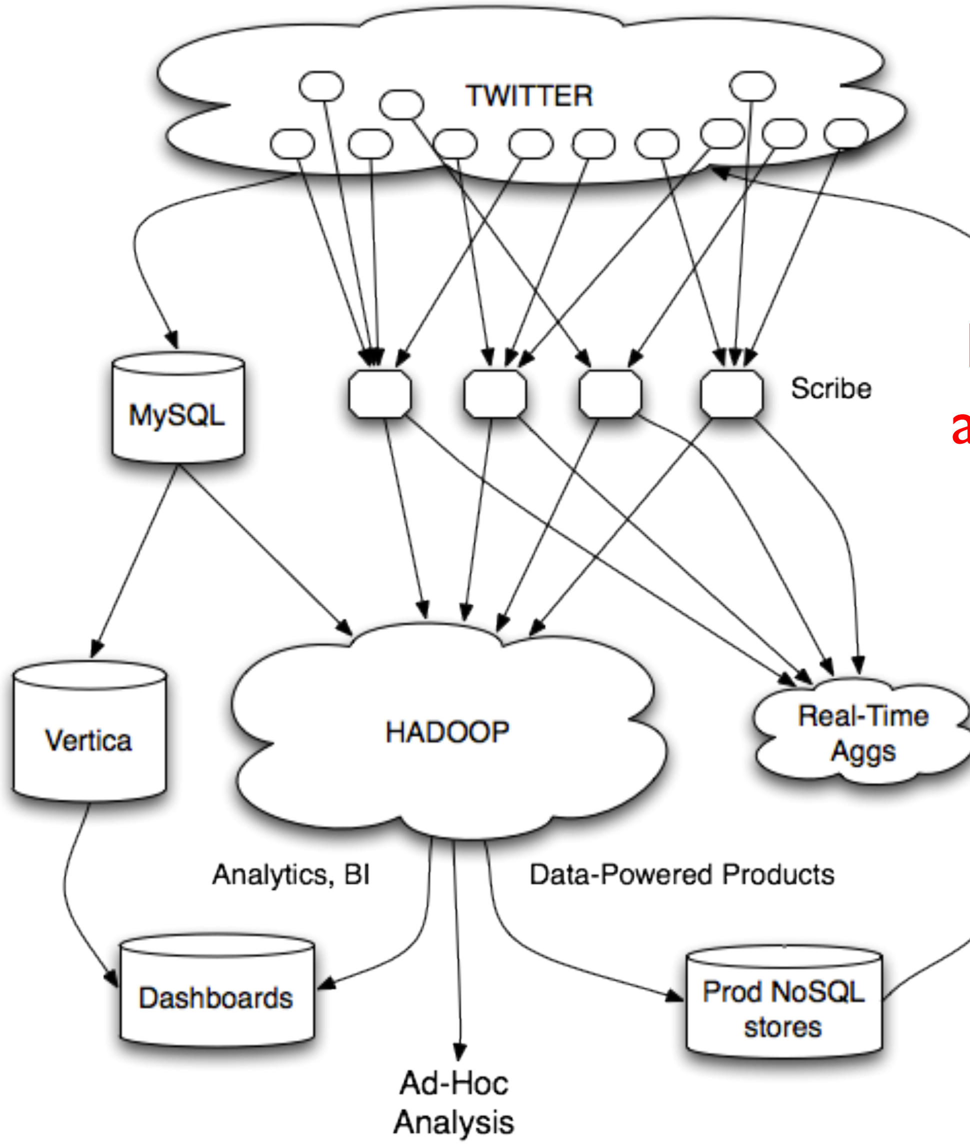
10s of Ks of Hadoop nodes, multiple DCs

10s of PBs total Hadoop DW capacity

~100 TB ingest daily

dozens of teams use Hadoop daily

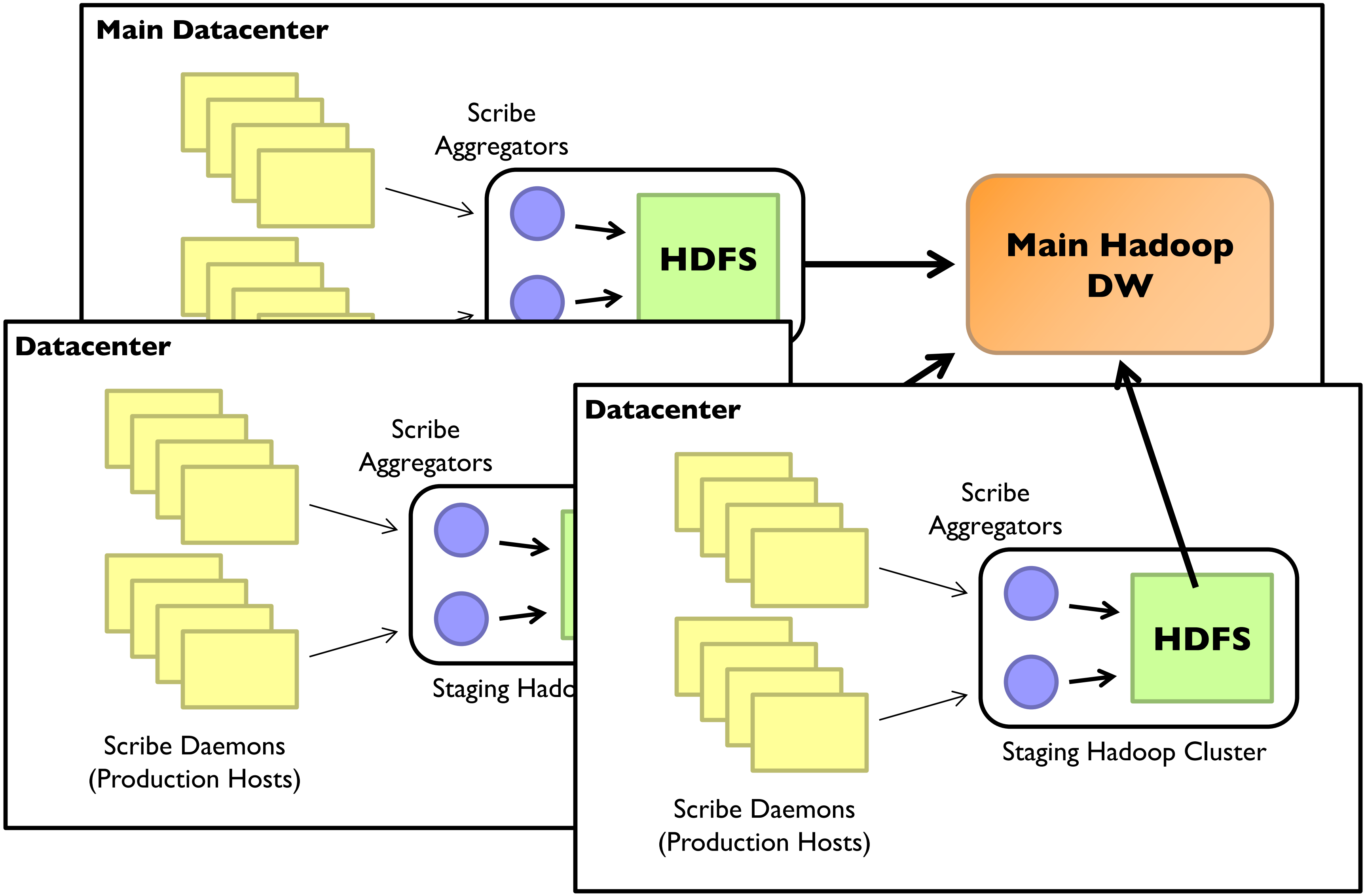
10s of Ks of Hadoop jobs daily



How does ETL actually happen?

Twitter's data warehousing architecture (circa 2012)

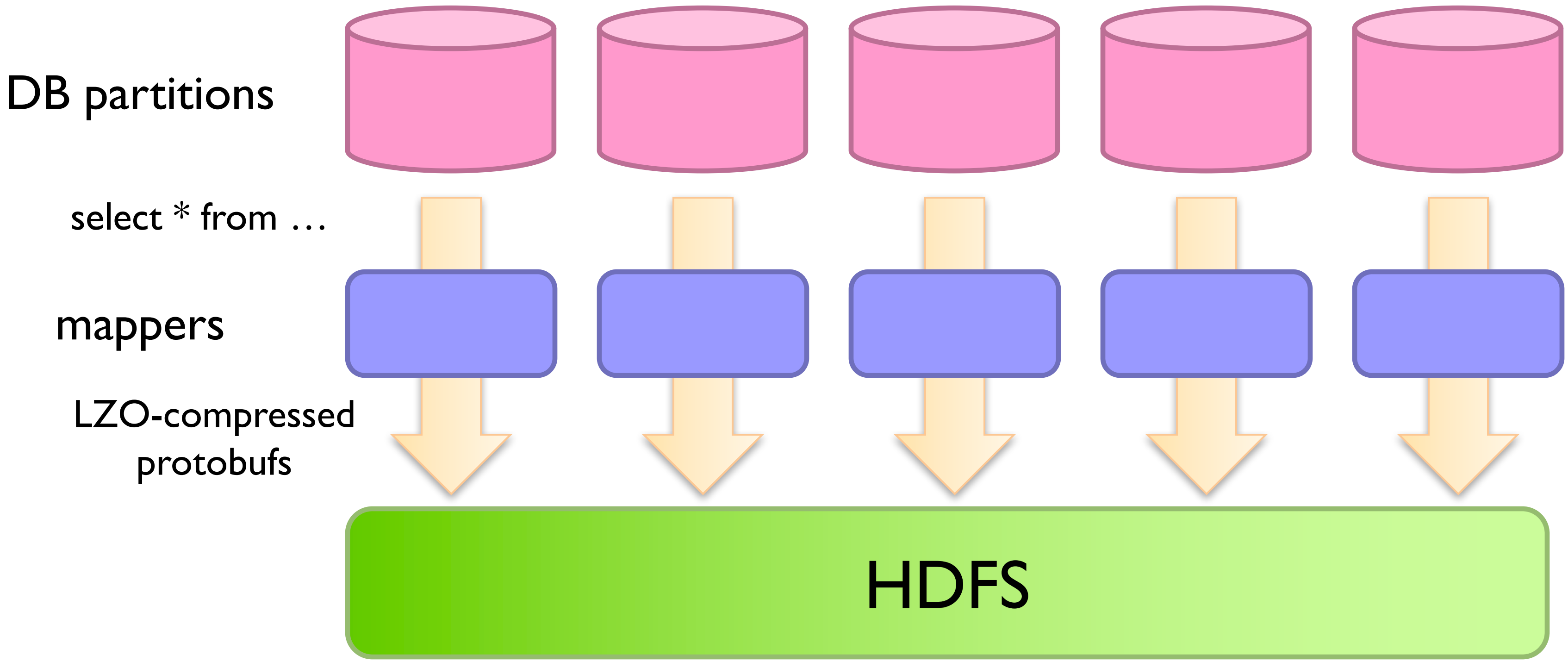
Importing Log Data



Importing Log Data*

Tweets, graph, users profiles

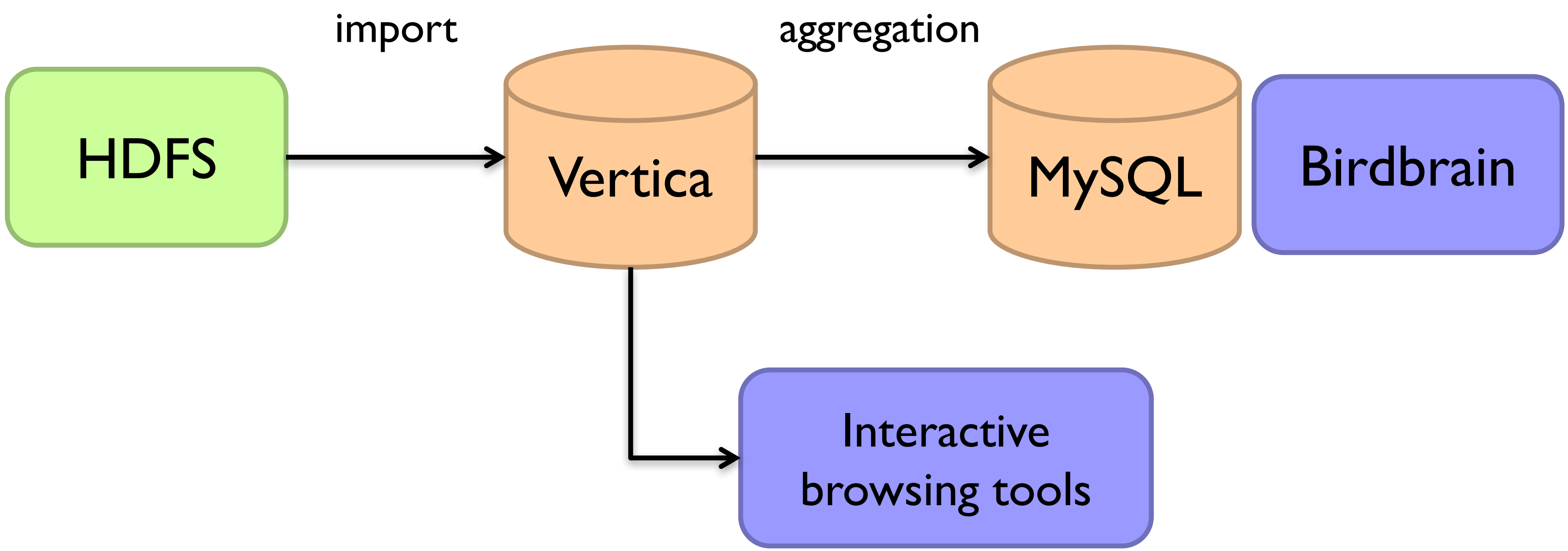
Different periodicity (e.g., hourly, daily snapshots, etc.)



Important: Must carefully throttle resource usage…

* Out of date – for illustration only

Vertica Pipeline*



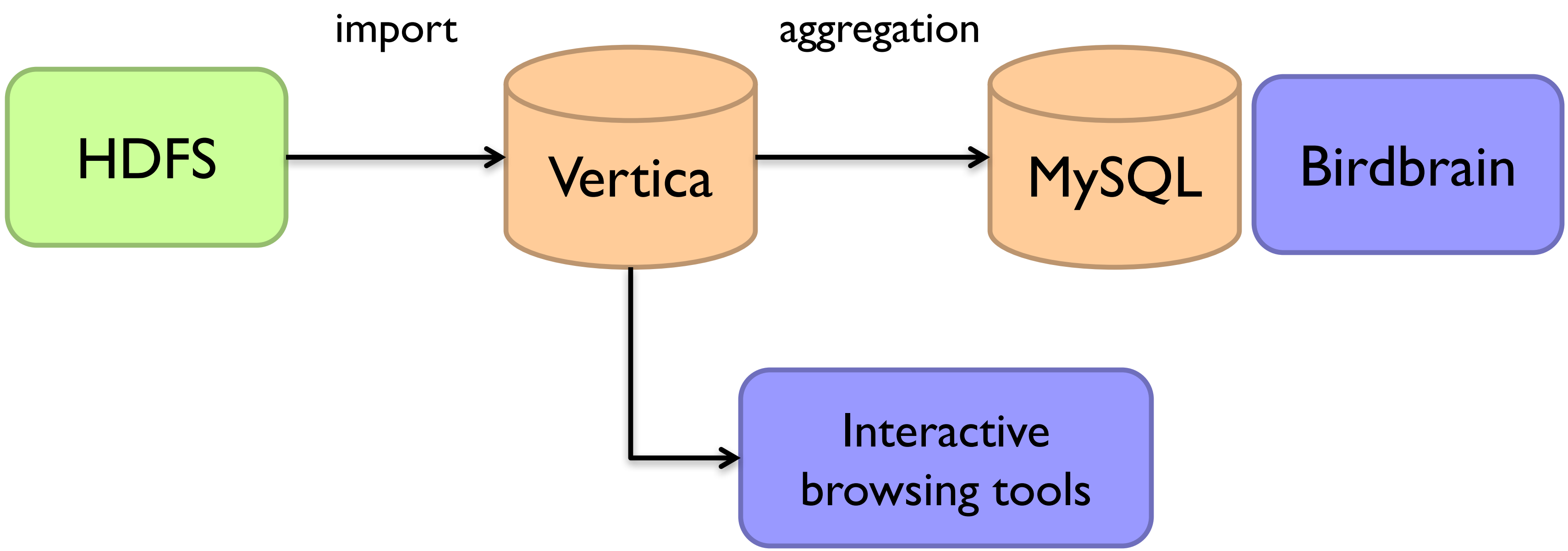
Why?

Vertica provides *orders of magnitude* faster aggregations!

"Basically, we use Vertica as a cache for HDFS data."
@squarecog

* Out of date – for illustration only

Vertica Pipeline*



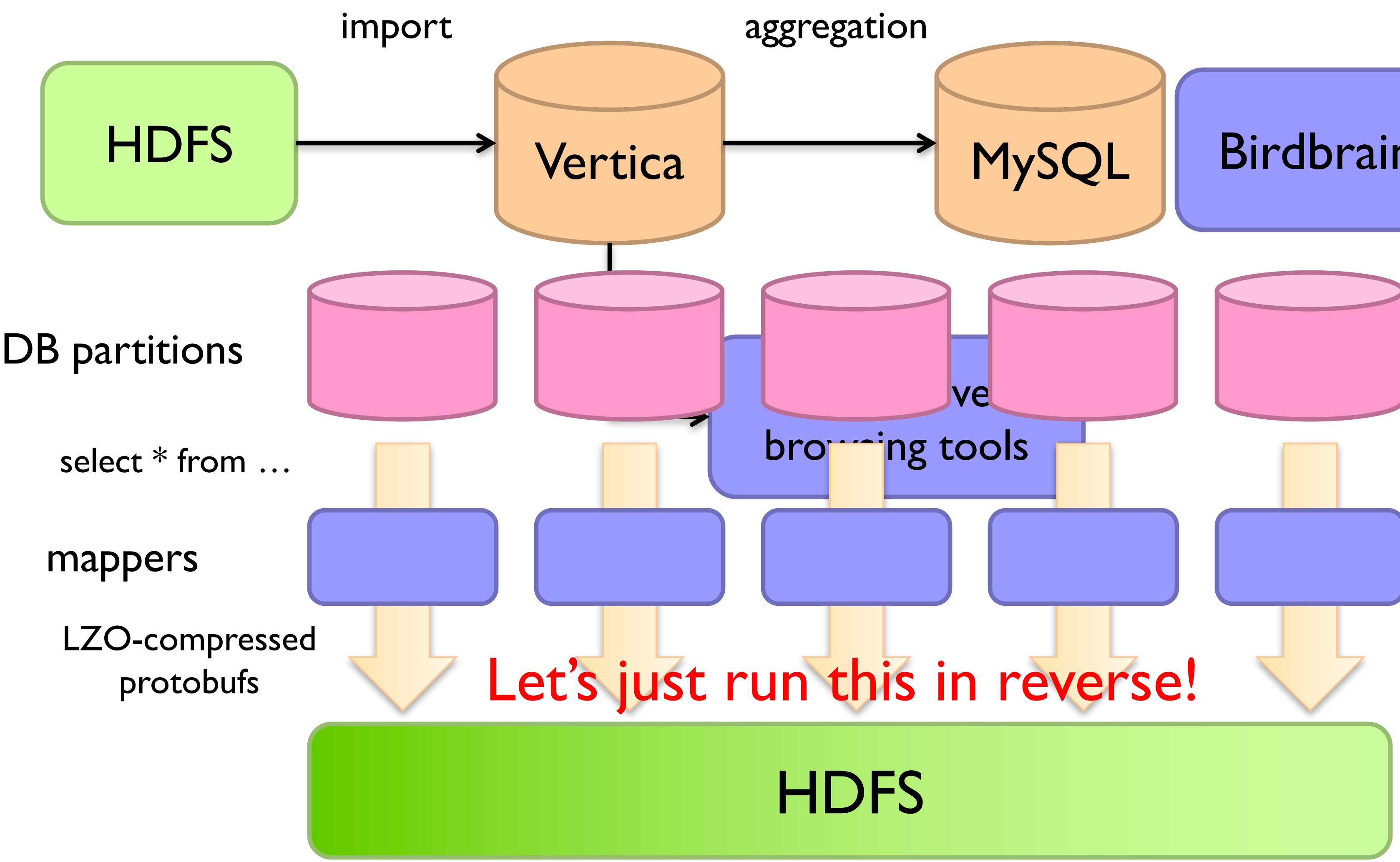
The catch…

Performance must be balanced against integration costs

Vertica integration is non-trivial

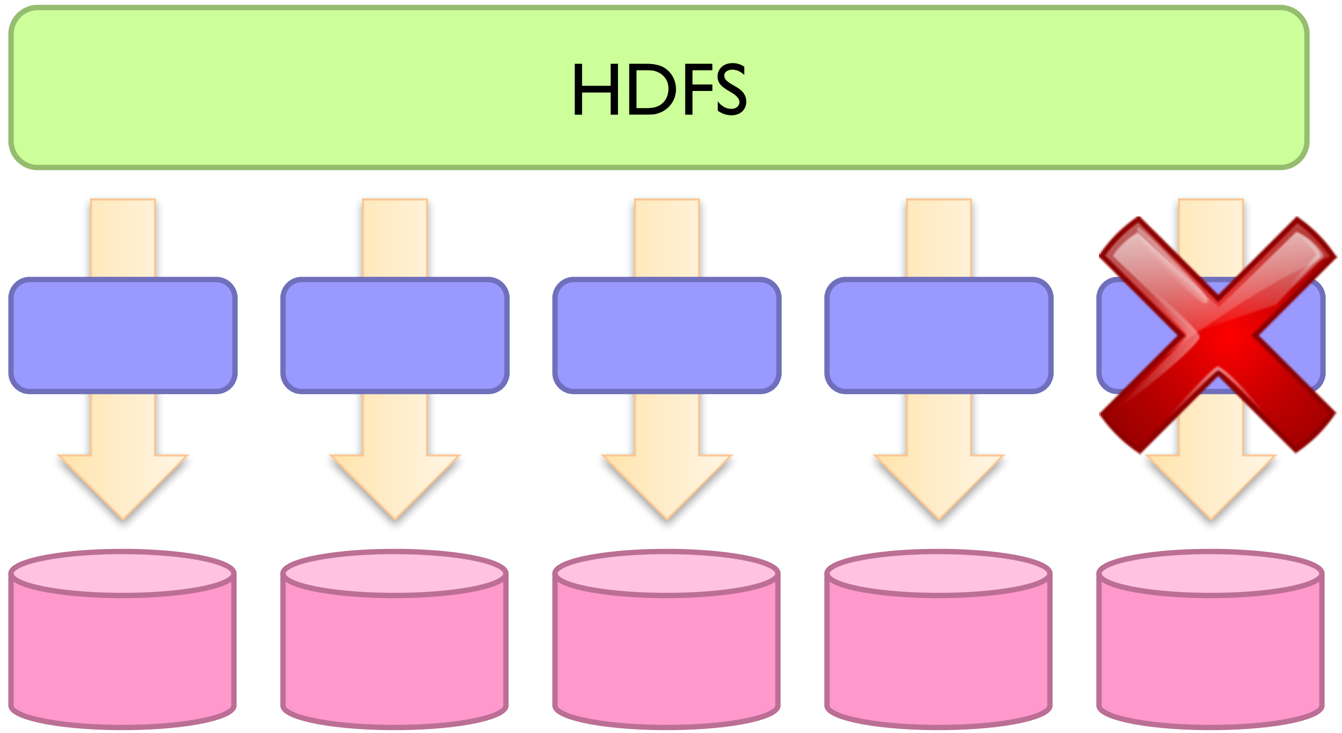
* Out of date – for illustration only

Vertica Data Ingestion



* Out of date – for illustration only

Vertica Pig Storage*



reducers

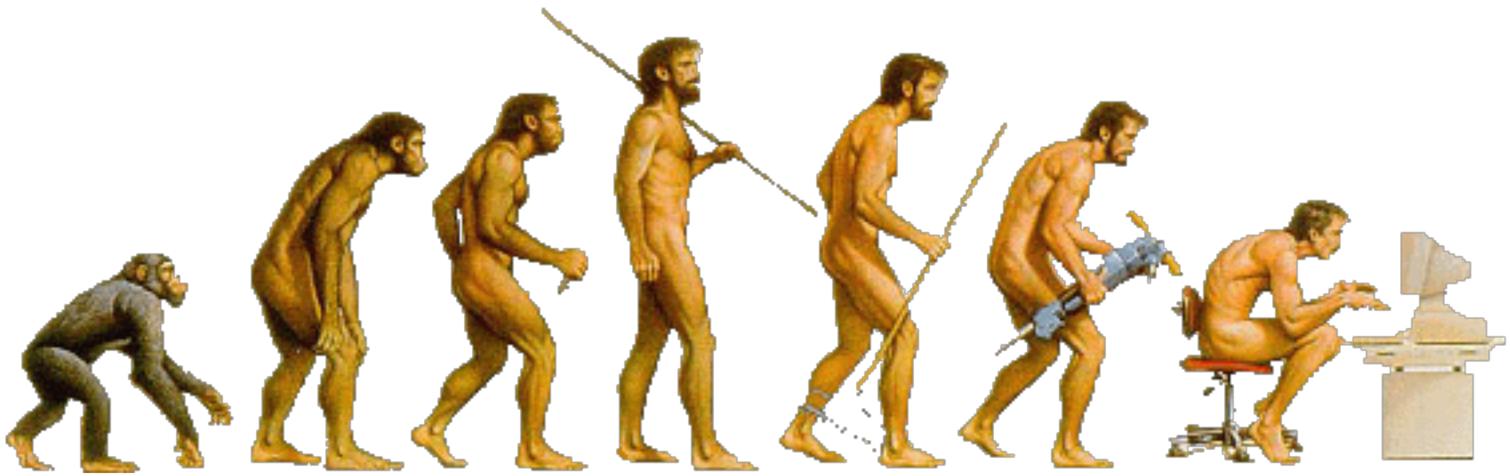
Vertica guarantees that each of these batch inserts are atomic

Vertica partitions

So what's the challenge?

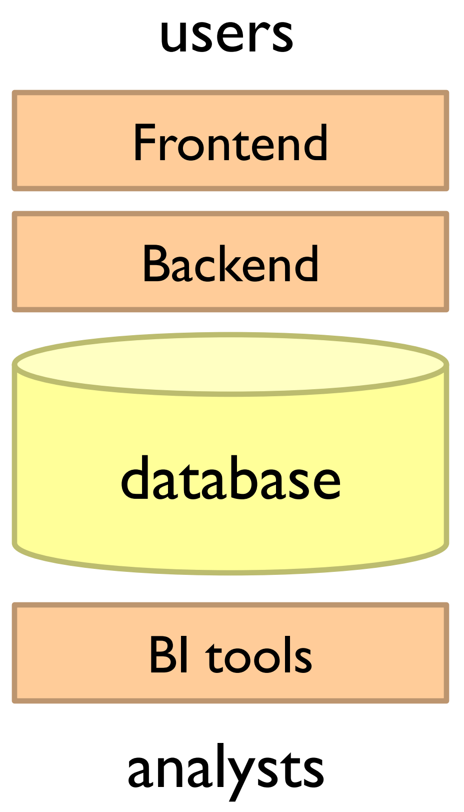
- Did you remember to turn off speculative execution?
- What happens when a task dies?

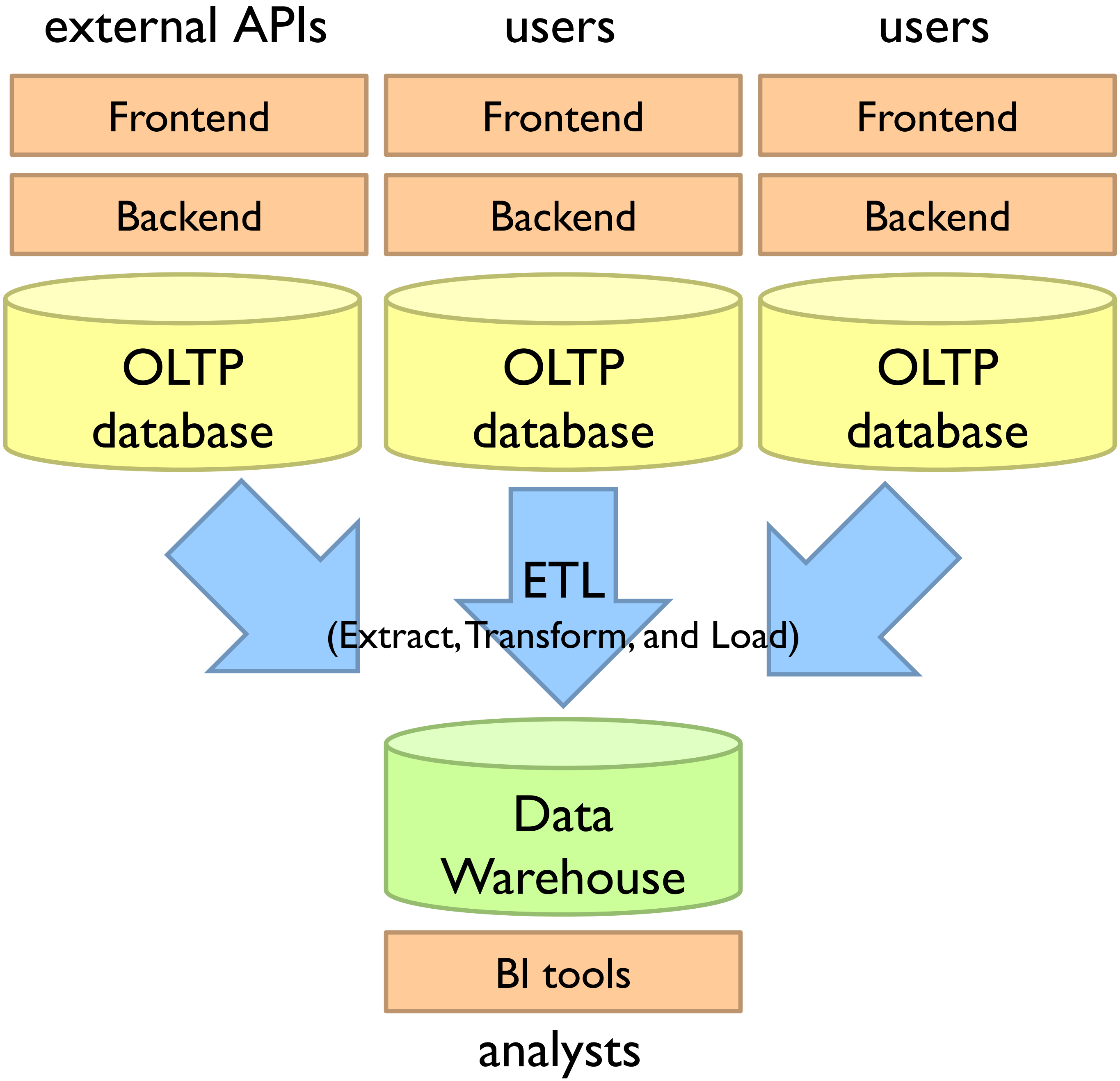
* Out of date – for illustration only

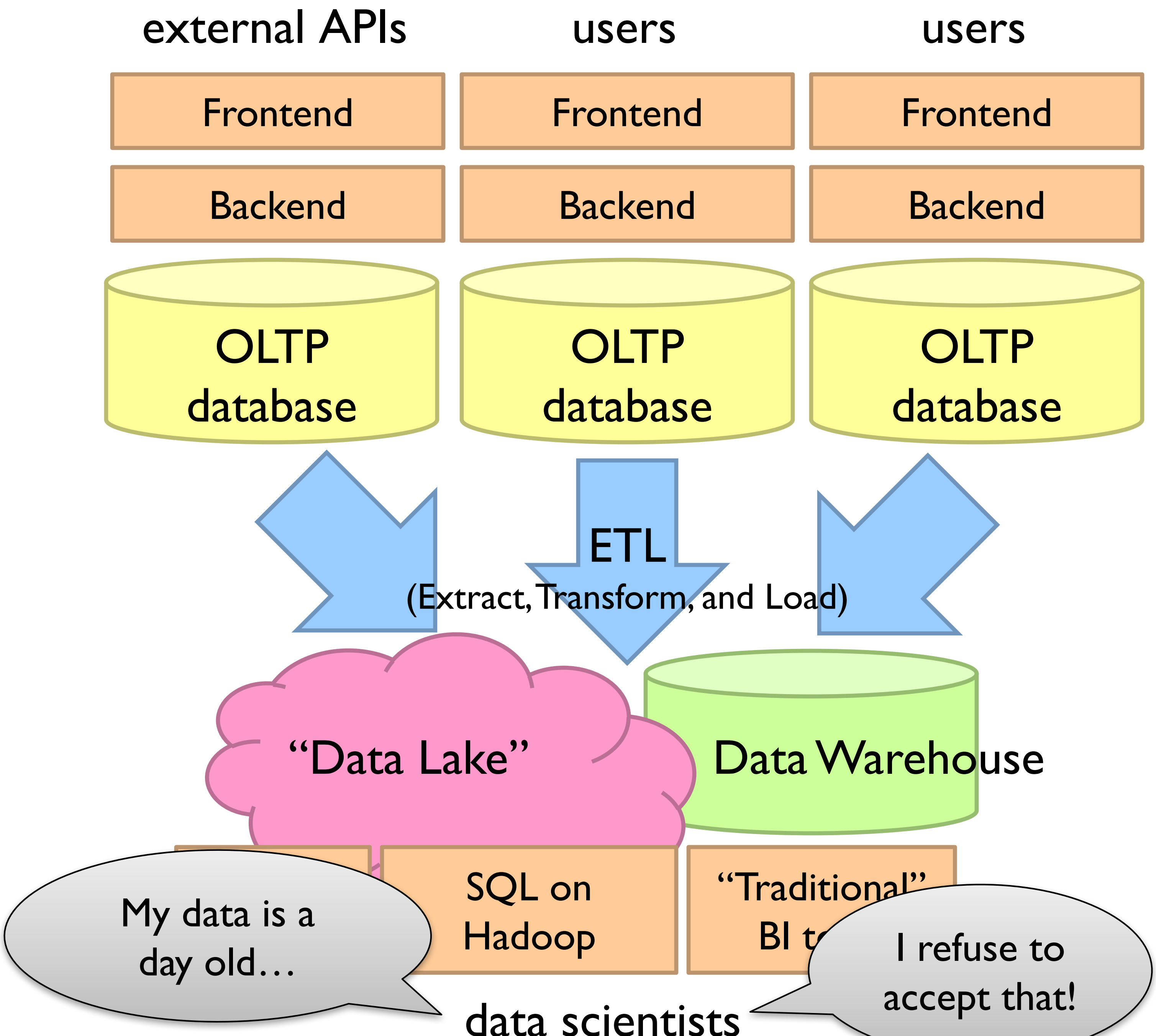


What's Next?

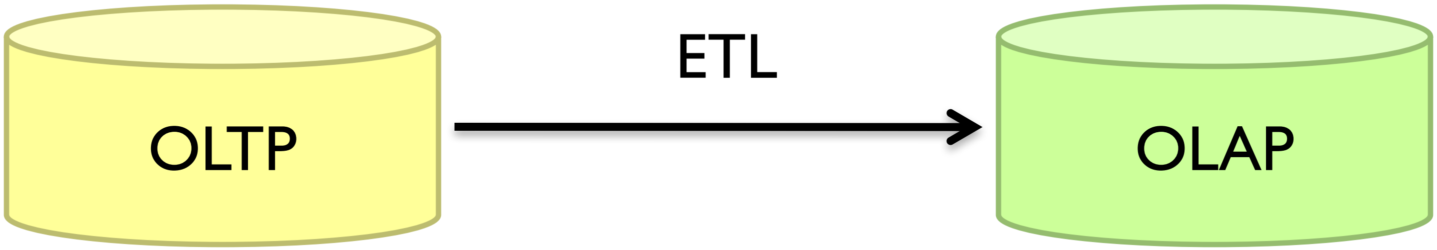
Two developing trends…


users
Frontend
Backend
database
BI tools
analysts

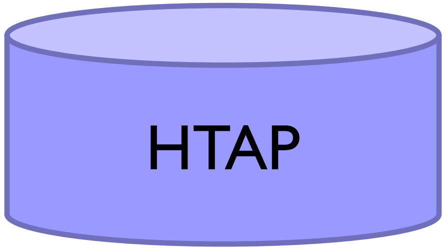

external APIs
users
users
Frontend
Frontend
Frontend
Backend
Backend
Backend
OLTP database
OLTP database
OLTP database
ETL
(Extract, Transform, and Load)
Data Warehouse
BI tools
analysts



external APIs
users
users
Frontend
Frontend
Frontend
Backend
Backend
Backend
OLTP database
OLTP database
OLTP database
ETL
(Extract, Transform, and Load)
"Data Lake"
Data Warehouse
SQL on Hadoop
"Traditional" BI t
My data is a day old…
I refuse to accept that!
data scientists

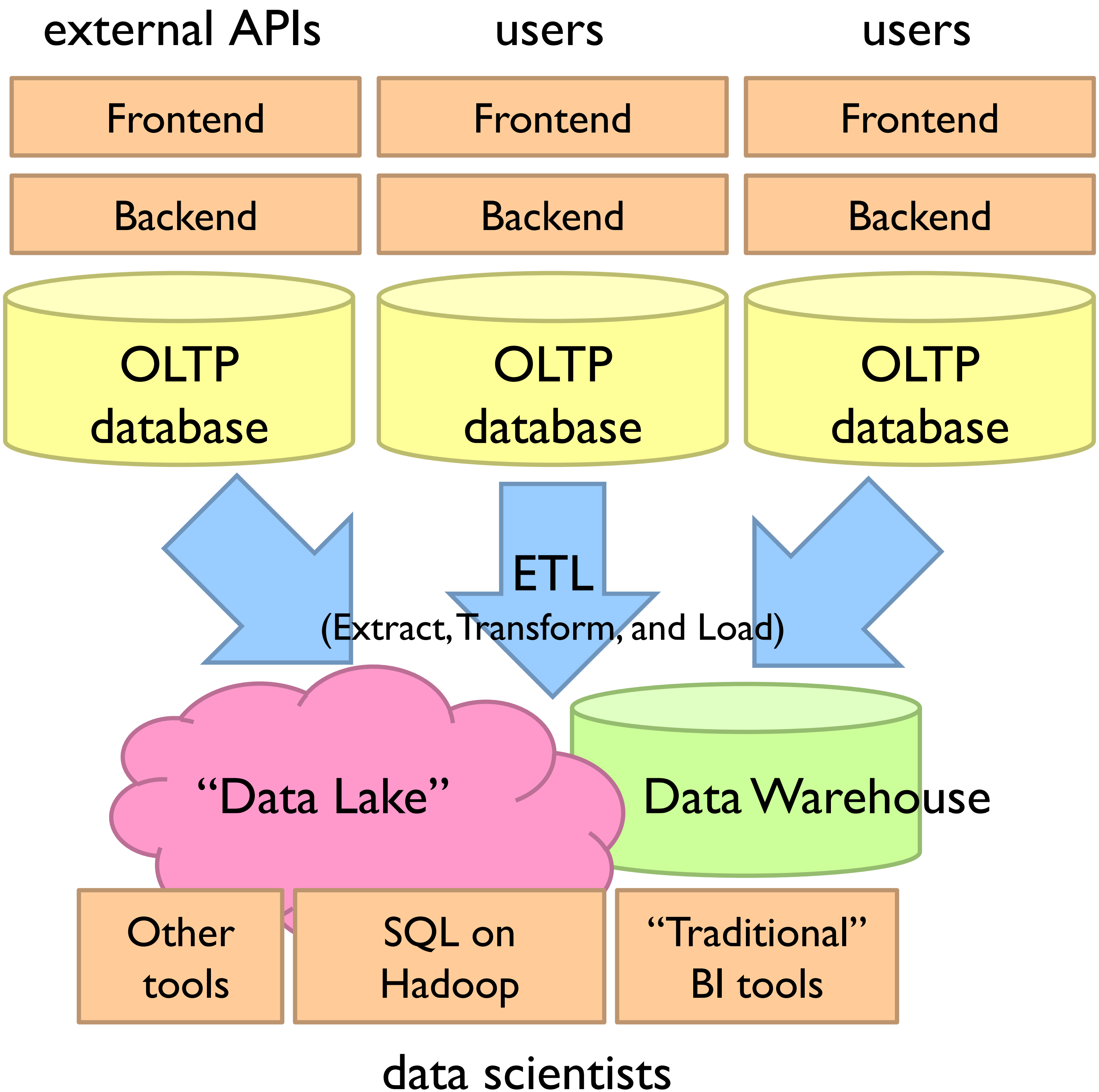


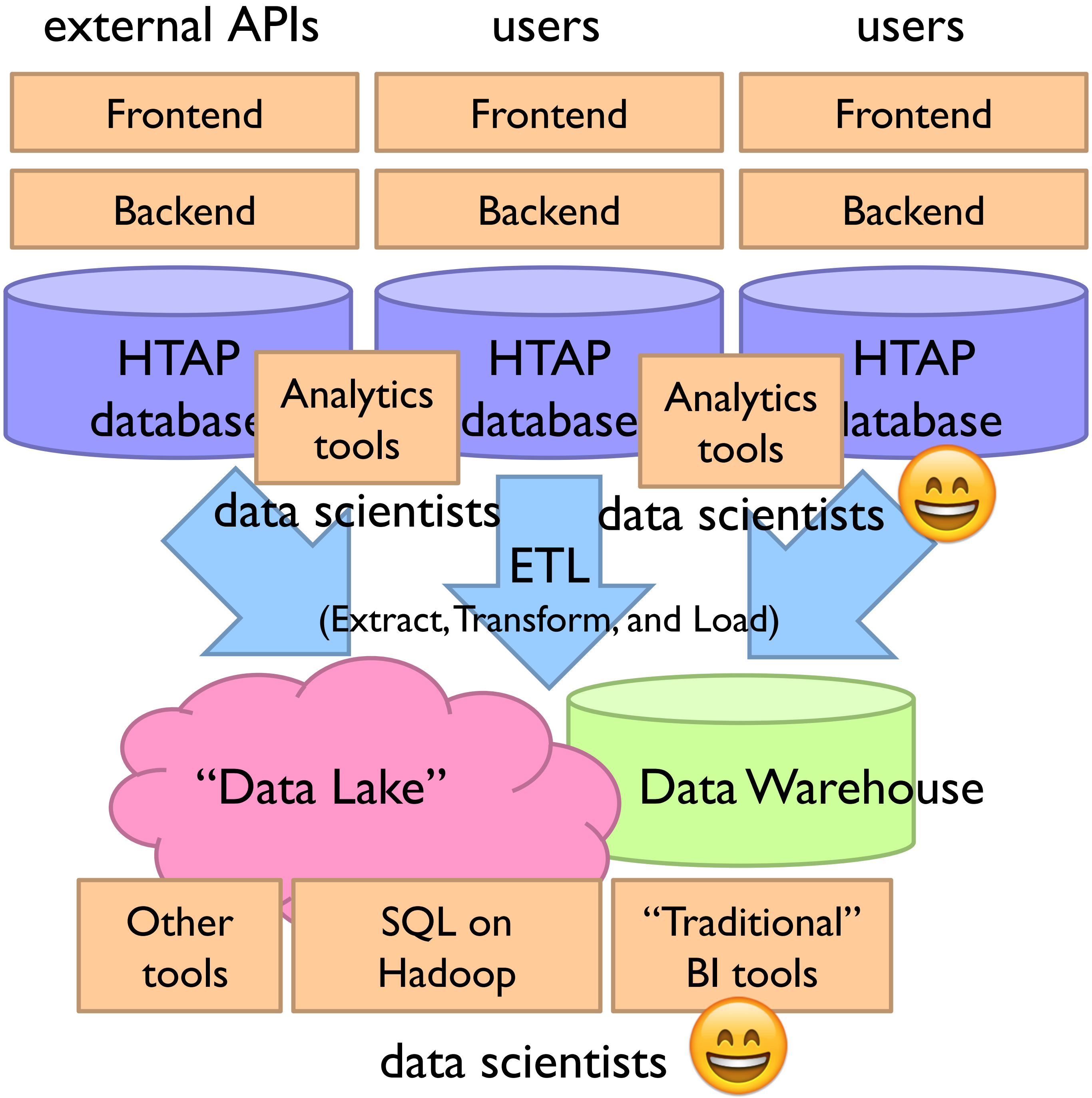
What if you didn't have to do this?

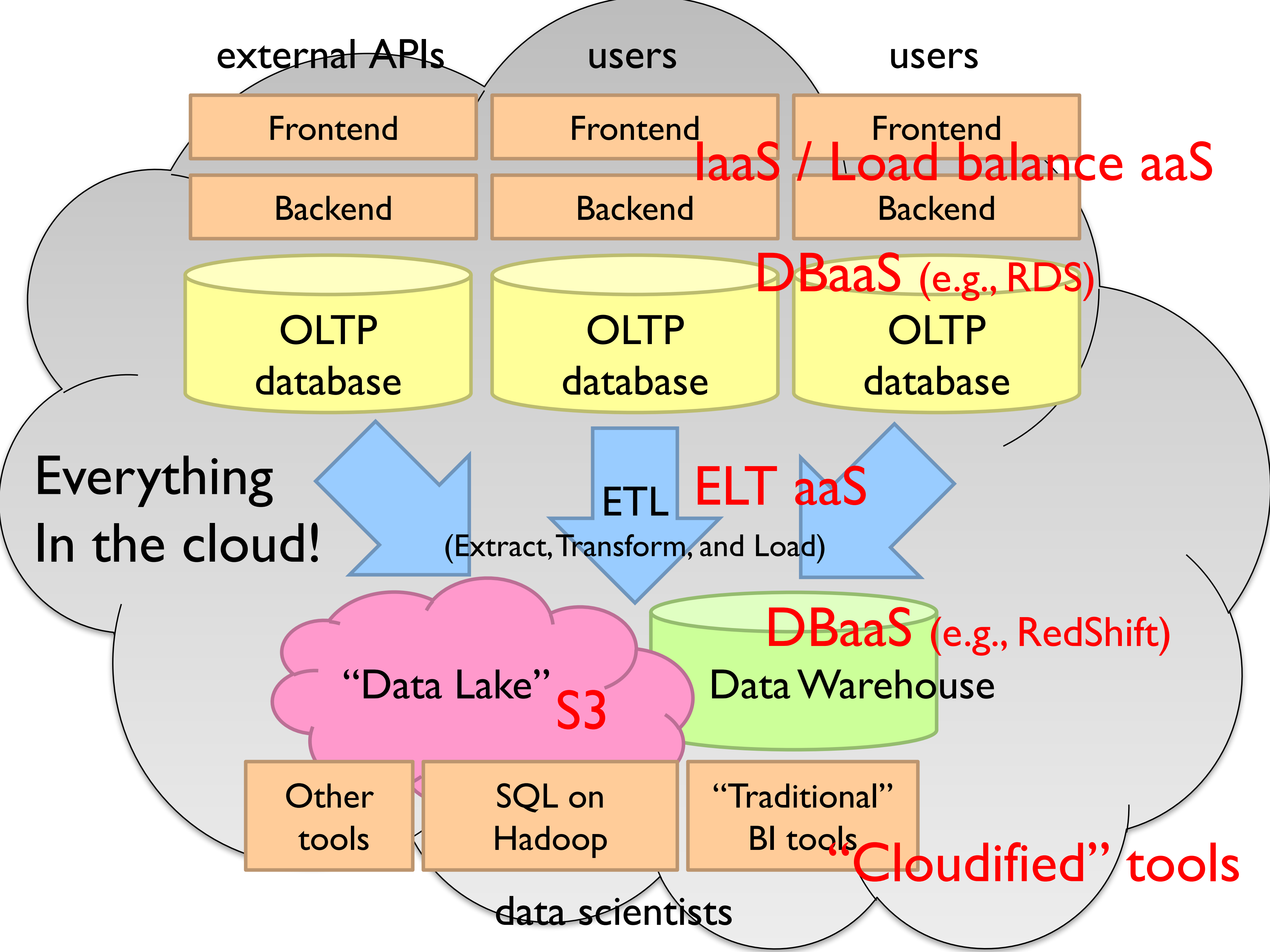


Hybrid Transactional/Analytical Processing (HTAP)

Coming back full circle?


external APIs
users
users
Frontend
Frontend
Frontend
Backend
Backend
Backend
OLTP database
OLTP database
OLTP database
ETL
(Extract, Transform, and Load)
"Data Lake"
Data Warehouse
Other tools
SQL on Hadoop
"Traditional" BI tools
data scientists


external APIs
users
users
Frontend
Frontend
Frontend
Backend
Backend
Backend
HTAP database
HTAP database
HTAP database
Analytics tools
Analytics tools
data scientists
data scientists
ETL
(Extract, Transform, and Load)
"Data Lake"
Data Warehouse
Other tools
SQL on Hadoop
"Traditional" BI tools
data scientists


external APIs
users
users
Frontend
Frontend
Frontend
IaaS / Load balance aaS
Backend
Backend
Backend
DBaaS (e.g., RDS)
OLTP database
OLTP database
OLTP database
Everything
In the cloud!
ETL
(Extract, Transform, and Load)
ELT aaS
DBaaS (e.g., RedShift)
"Data Lake"
S3
Data Warehouse
Other tools
SQL on Hadoop
"Traditional" BI tools
"Cloudified" tools
data scientists

Source: Wikipedia (Japanese rock garden)