From Apache Spark to Delta Lake: enabling complex data applications

Reynold Xin @rxin

HPTS 2019

databricks

Dan Primack Oct 23, 2019

Analytics company Databricks raises $400 million

Databricks, a San Francisco-based data analytics SaaS company, raised $400 million in Series F funding led by Andreessen Horowitz at a $6.2 billion valuation.

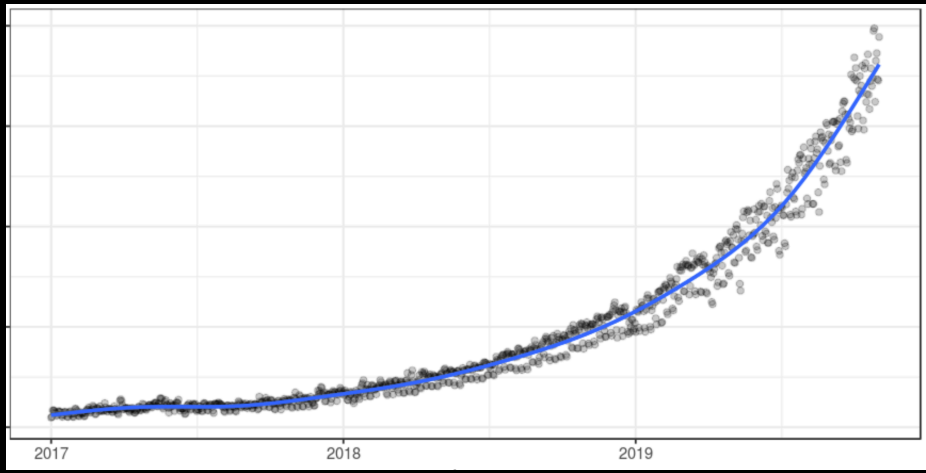
Why it matters: Because this one can legitimately lay claim to all the hottest enterprise software buzzwords, from open-source to machine learning to cloud.

- In short, its big promise is to help customers add AI to existing business software — kind of SaaS-upon-SaaS. Plus, it just added the CFO who helped take Splunk public.
- **Other investors** include Alkeon Capital Management, BlackRock, Coatue, Dragoneer, Geodesic, Green Bay Ventures, NEA, T. Rowe Price, and Tiger Global.

The bottom line: "Databricks' platform is used by enterprises to analyze data, build data pipelines across siloed storage systems and prepare labeled datasets for model building. The idea is that organizations can then use the

Databricks Platform

> 1 million VMs per day launched.



> 1 exabytes data processed per week (soon to be per day).

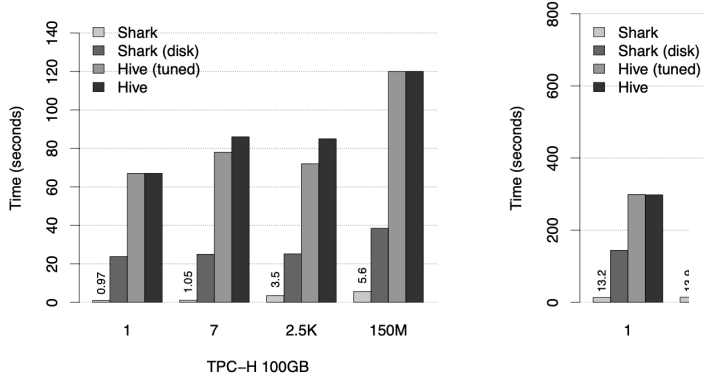
whoami

Databricks co-founder & Chief Architect

- Designed (and implemented) most of “modern day” Spark
- #1 code contributor to Spark by commits and **net lines deleted**

PhD in databases from Berkeley

I used to be obsessed with perf eng



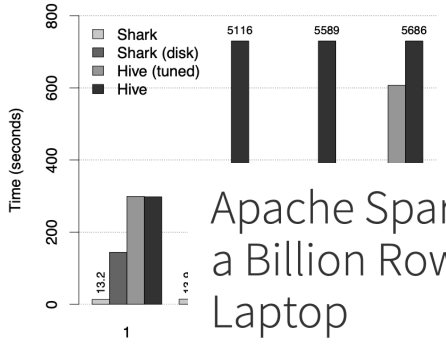


Figure 6: Aggregation queries on *lineitem* table. X-axis indicates the number of gr

$1.44 per terabyte: setting a new world record with Apache Spark

Architecting the most efficient cloud data processing platform

by Reynold Xin

2016

ffort by Nanjing University, Alibaba Group, and
rt, a well-established third-party benchmark.
y to sort 100 TB of data, using only $144.22
fficient than the previous world record.
ith our partners, demonstrated that the most
Spark in the cloud.

Apache Spark as a Compiler: Joining a Billion Rows per Second on a Laptop

Deep dive into the new Tungsten execution engine

by Sameer Agarwal, Davies Liu and Reynold Xin
Posted in **ENGINEERING BLOG** | May 23, 2016

Try this notebook in Databricks

When our team at Databricks planned our contributions to the upcoming Apache Spark 2.0 release, we set out with an ambitious goal by asking ourselves: **Apache Spark is already pretty fast, but can we make it 10x faster?**

ort Benchmark

100TB of data

$4.51
per terabyte

UC San Diego

Alibaba Group

databricks

In Collaboration

databricks

“The greatest performance improvement of all is when a system goes from not-working to working.”

John Ousterhout

This Talk

- What is Apache Spark?
- Delta Lake: scalable & reliable data lakes

This Talk

/ What is Apache Spark?

/ Delta Lake: scalable & reliable data lakes

MapReduce: A major step backwards

By David DeWitt on January 17, 2008 4:20 PM | Permalink | Comments (44) | TrackBacks (1)
[Note: Although the system attributes this post to a single author, it was written by David J. DeWitt and Michael Stonebraker]

On January 8, a Database Column reader asked for our views on new distributed database research efforts, and we'll begin here with our discuss it, since the recent trade press has been filled with news of the revolution of so-called "cloud computing." This paradigm entails processors working in parallel to solve a computing problem. In effect, this suggests constructing a data center by lining up a large numb smaller number of high-end servers.

For example, IBM and Google have announced plans to make a 1,000 processor cluster available to a few select universities to teach stu software tool called MapReduce [1]. Berkeley has gone so far as to plan on teaching their freshman how to program using the MapRedu

As both educators and researchers, we are amazed at the hype that the MapReduce proponents have spread about how it represents a par intensive applications. MapReduce may be a good idea for writing certain types of general-purpose computations, but to the database co

1. A giant step backward in the programming paradigm for large-scale data intensive applications
2. A sub-optimal implementation, in that it uses brute force instead of indexing
3. Not novel at all -- it represents a specific implementation of well known techniques developed nearly 25 years ago
4. Missing most of the features that are routinely included in current DBMS
5. Incompatible with all of the tools DBMS users have come to depend on

2013 pitch: a better MapReduce

What is Spark?

Fast and expressive cluster computing system interoperable with Apache Hadoop

Improves efficiency through:
- » In-memory computing primitives
- » General computation graphs

→ Up to 100 × faster (2-10 × on disk)

Improves usability through:
- » Rich APIs in Scala, Java, Python
- » Interactive shell

→ Often 5 × less code

Spark's evolution since 2013

SQL (can run all TPC-DS/H queries)

Column store (Parquet, ORC, etc)

Query optimizer (heuristics & cost-based)

External operations (handle data > memory)

Volcano execution engine -> Hyper-style code gen engine

Streaming (incremental view maintenance)

…

What's the point in reinventing a SQL query engine?

Because databases suck at …

Fast iteration: installation, data modeling, etc, take LONG time.

Scalability.

Building complex data applications (focus of this talk).

Modern eng principles & practices

Modularity / separation of concerns
Incremental development
External dependencies, libraries, frameworks
Debugger
Change management
Testing (unit tests, integration tests)
Continuous integration / continuous delivery
…

SQL makes it virtually impossible to apply these.

So what is “modern day” Spark?

An open source relational query engine
with language-integrated APIs in Python, Scala, Java, C# …
and seamless UDFs

to enable
building complex apps
with
modern eng principles.

Subtract Mean

This example shows a simple use of grouped map Pandas UDFs: subtracting mean from each value in the group.

```
@pandas_udf(df.schema, PandasUDFType.GROUPED_MAP)
# Input/output are both a pandas.DataFrame
def subtract_mean(pdf):
    return pdf.assign(v=pdf.v - pdf.v.mean())

df.groupby('id').apply(subtract_mean)
```

This Talk

/ What is Apache Spark?

/ Delta Lake: scalable & reliable data lakes

Delta Lake

6 mo

since open source

4000+

use in companies

1 EB+

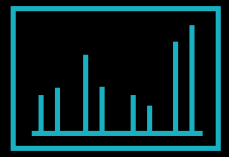
data processed / wk

What does a typical
data lake project look like?

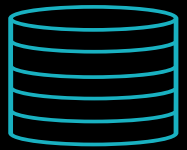
Evolution of a Cutting-Edge Data Lake

Events

?



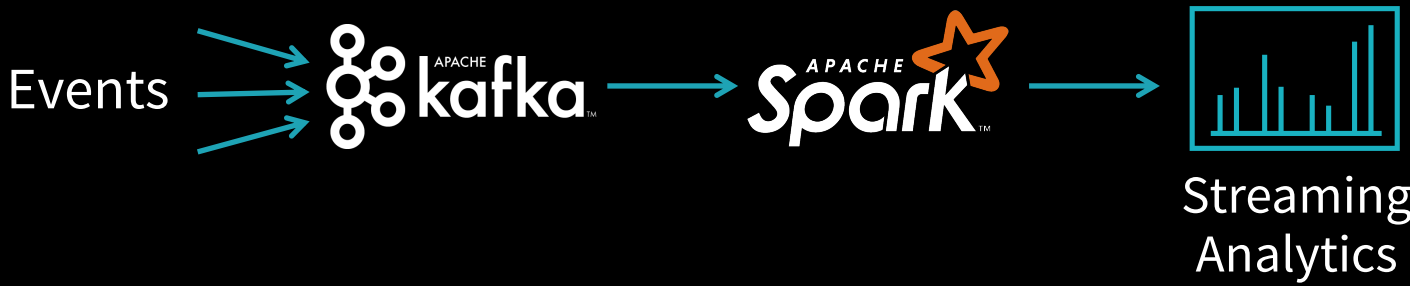
Streaming Analytics



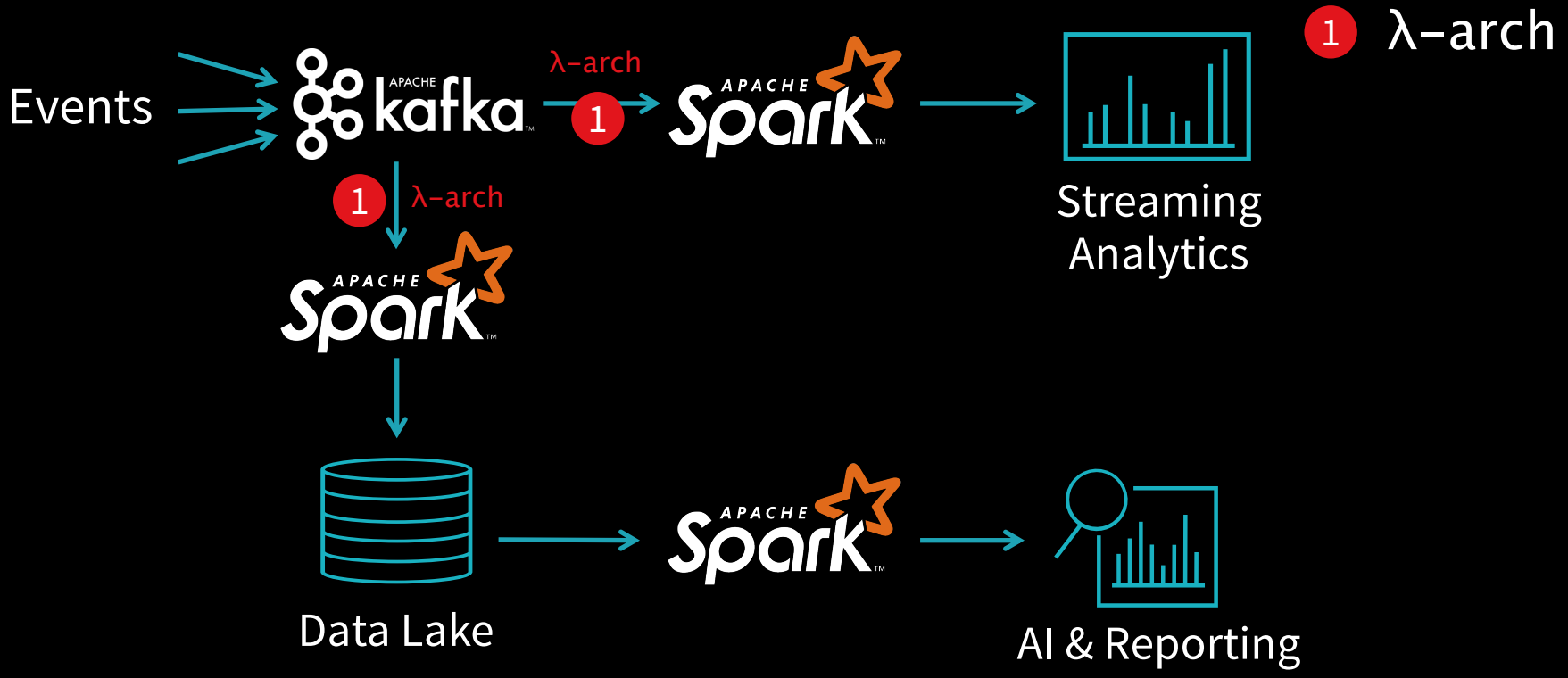
Data Lake

AI & Reporting

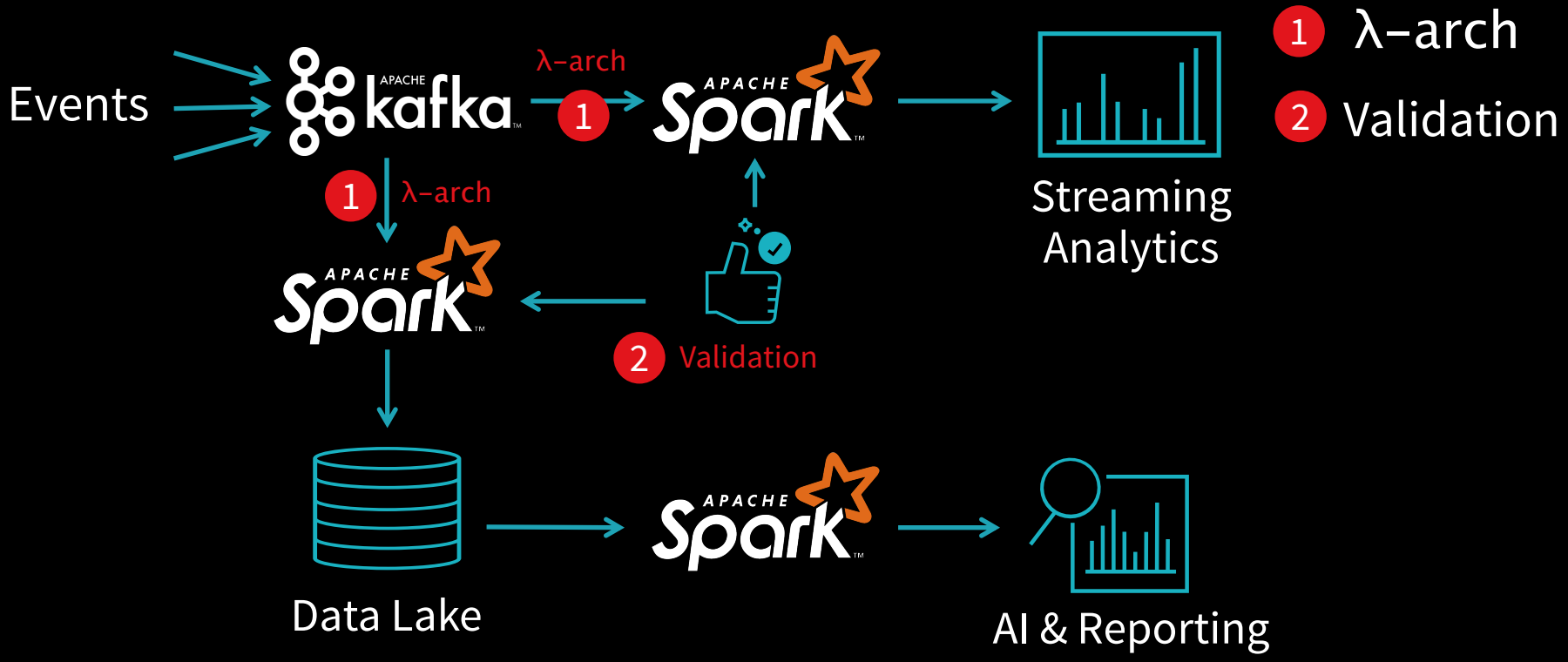
Evolution of a Cutting-Edge Data Lake



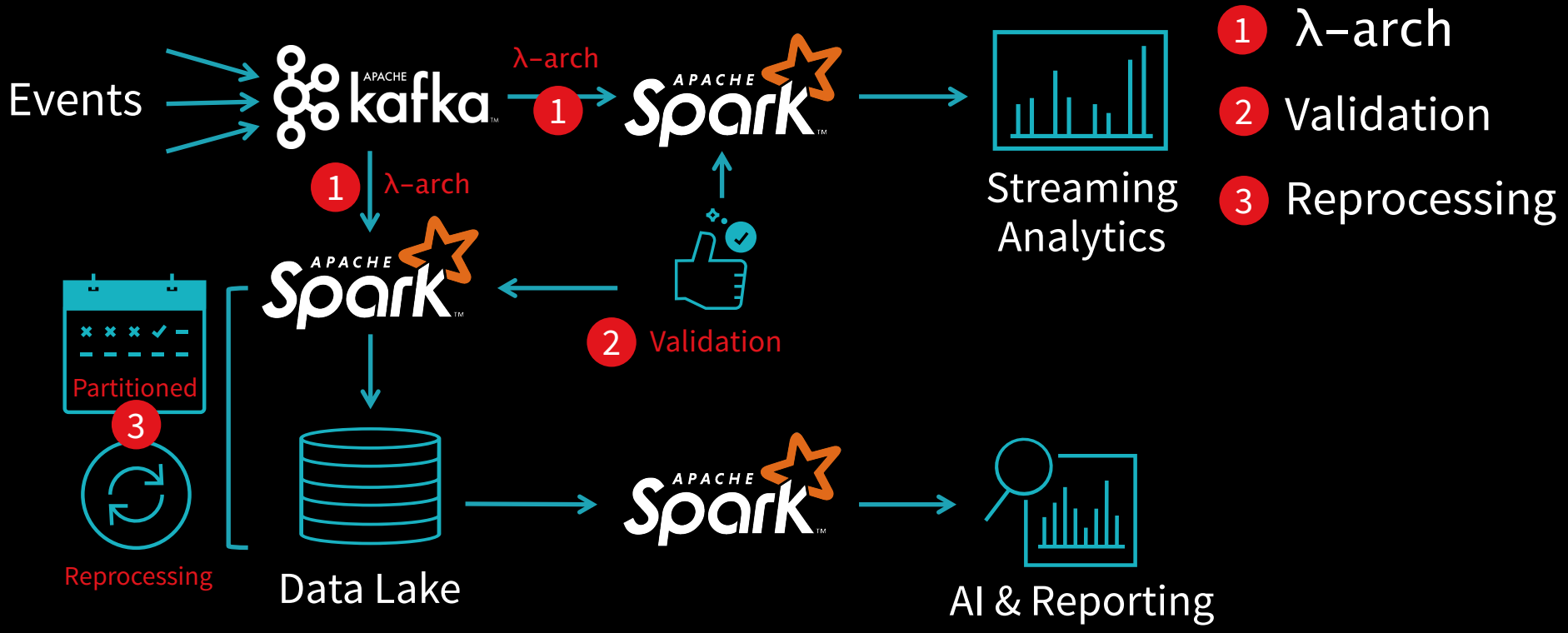
Challenge #1: Historical Queries?



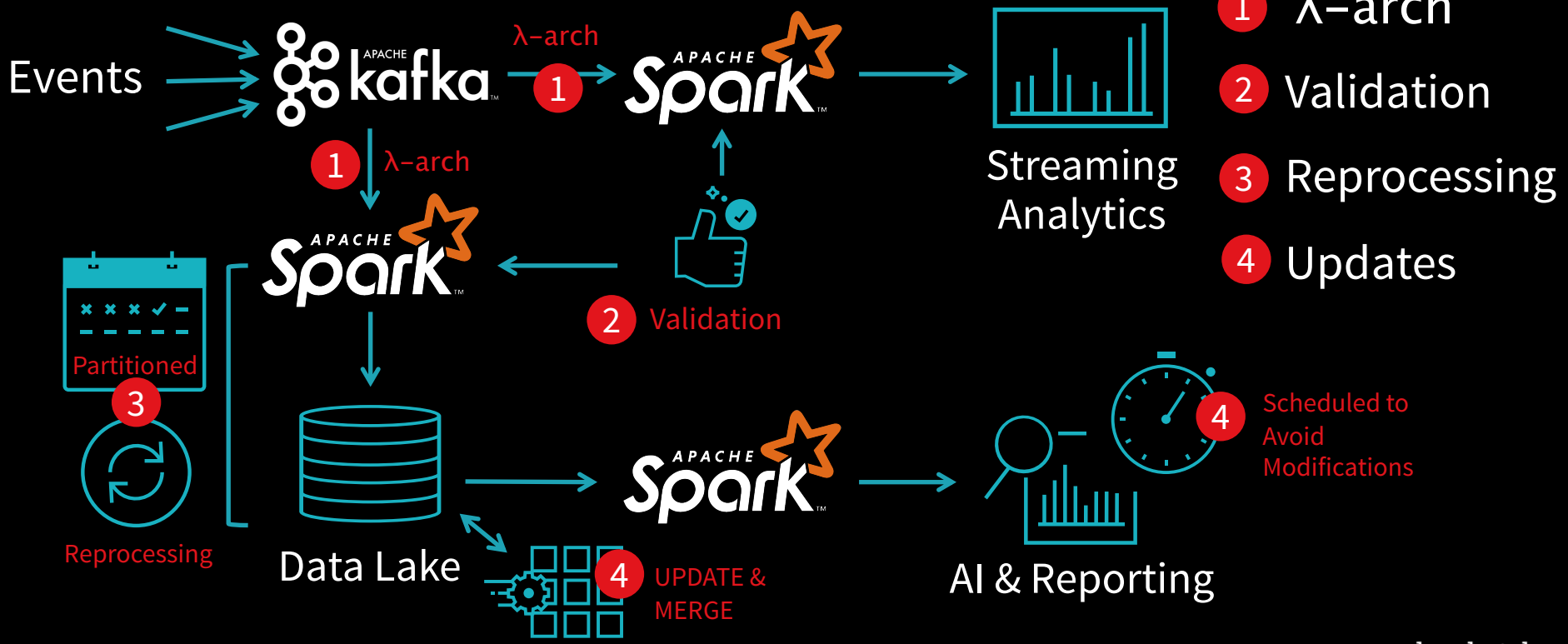
Challenge #2: Messy Data?



Challenge #3: Mistakes and Failures?



Challenge #4: Updates?



Wasting Time & Money

Solving Systems Problems

Instead of Extracting Value From Data

Data Lake Distractions

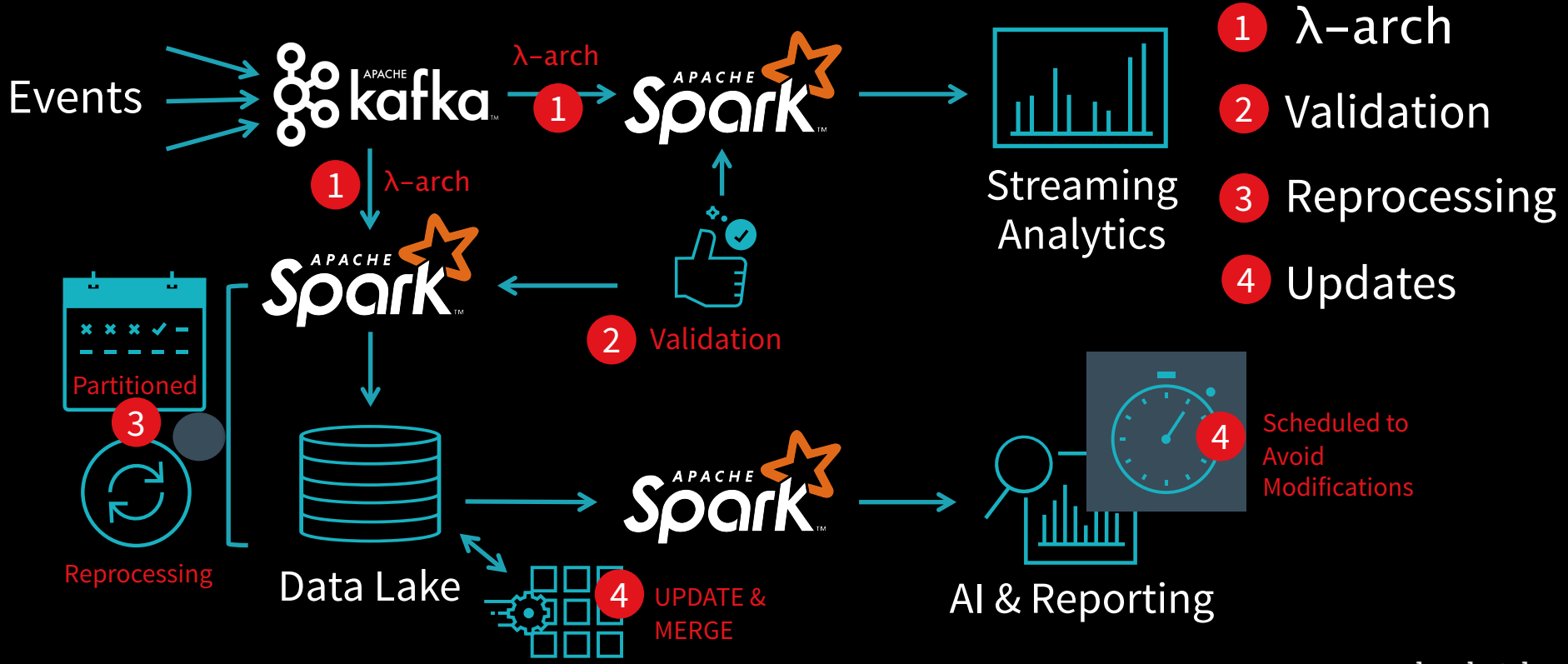
No atomicity means failed production jobs leave data in corrupt state requiring tedious recovery

No quality enforcement creates inconsistent and unusable data

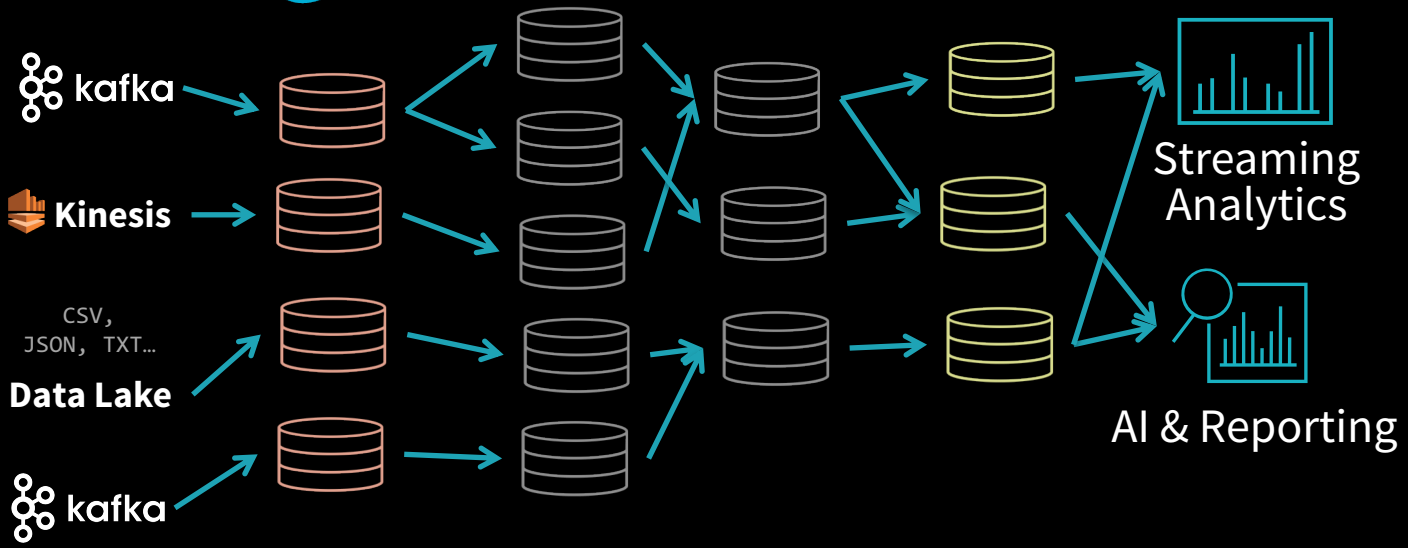
No consistency / isolation makes it almost impossible to mix appends and reads, batch and streaming

Let’s try it instead with

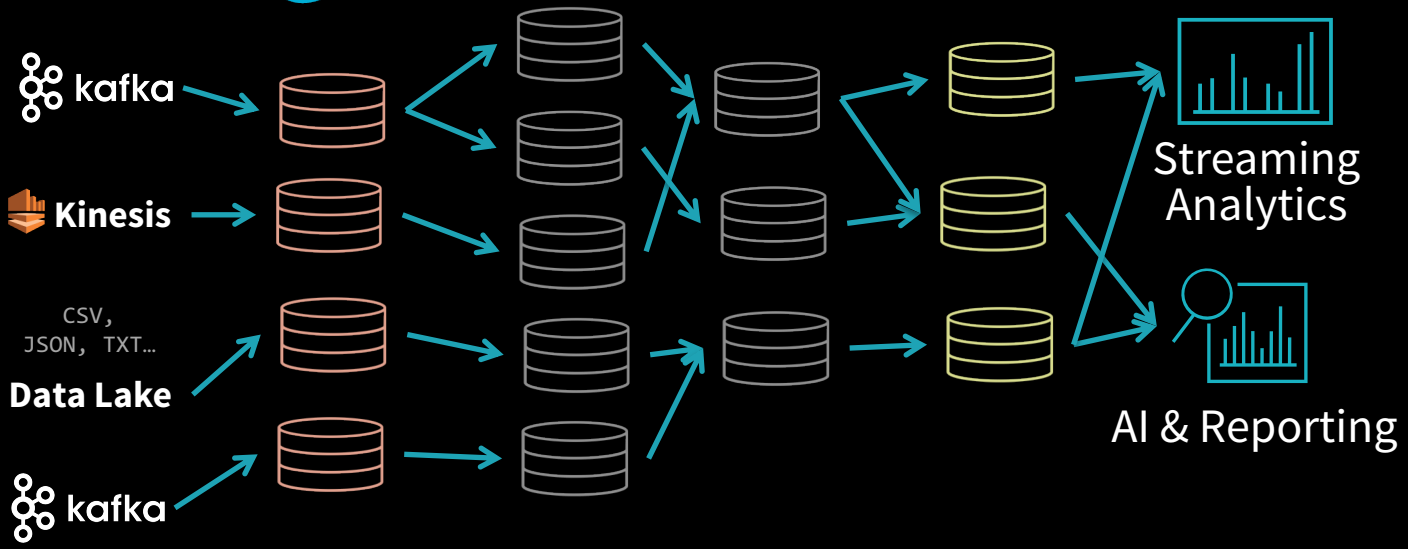
Challenges of the Data Lake



The DELTA LAKE Architecture



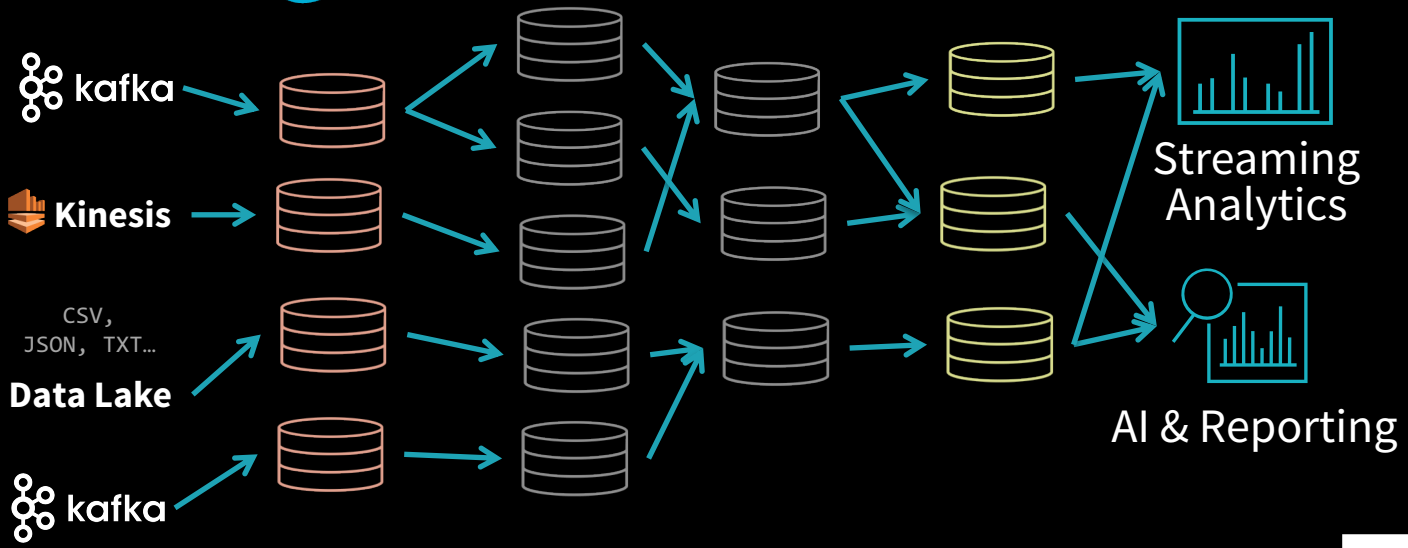
The DELTA LAKE Architecture



Full ACID Transaction

Focus on your data flow, instead of worrying about failures.

The DELTA LAKE Architecture

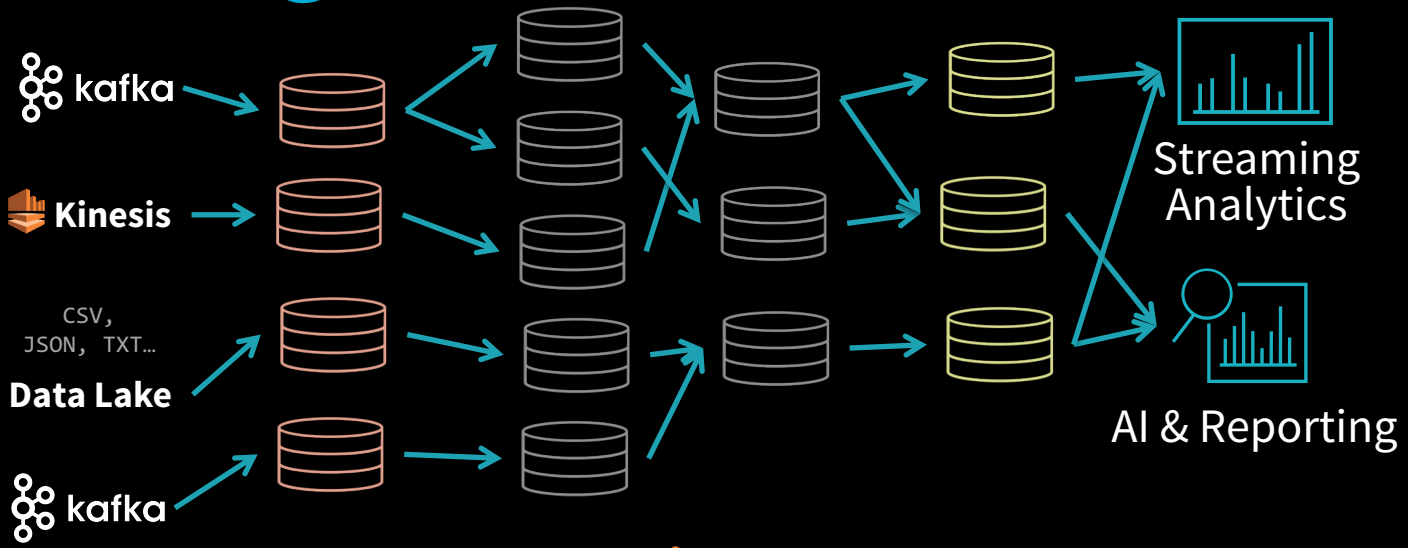


Open Standards, Open Source

THE LINUX FOUNDATION

Store petabytes of data without worries of lock-in. Growing community including Presto, Spark and more.

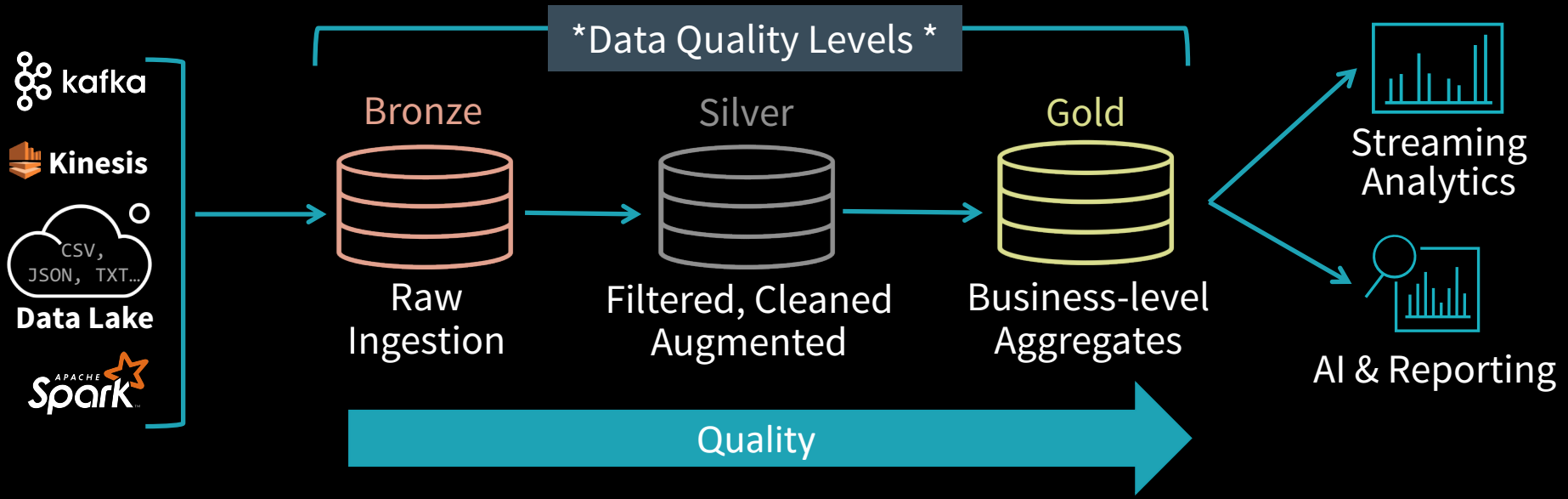
The DELTA LAKE Architecture



Powered by APACHE Spark

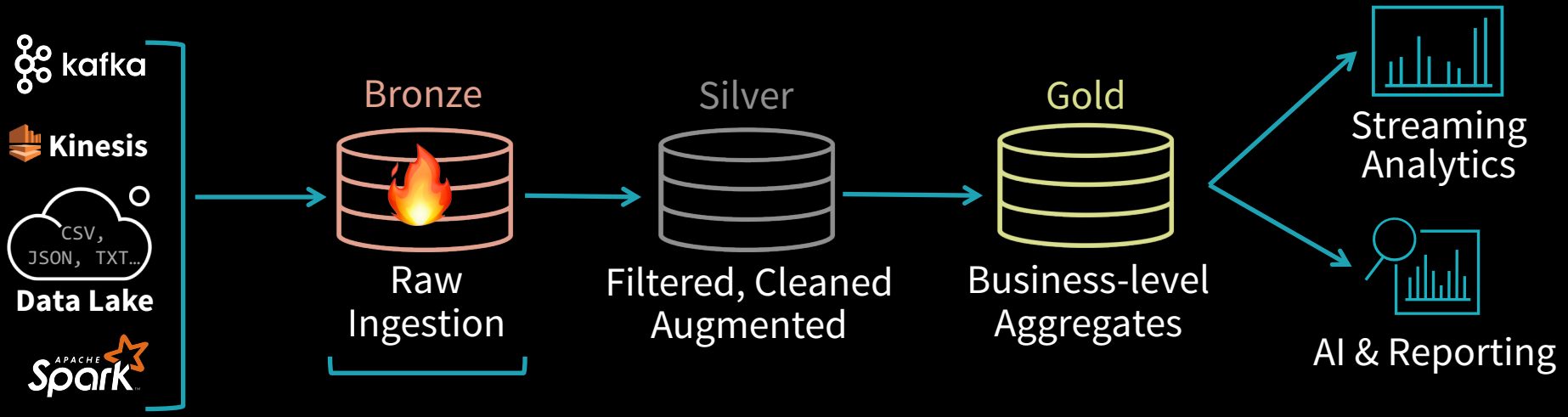
Unifies Streaming / Batch with unlimited scalability.

The DELTA LAKE



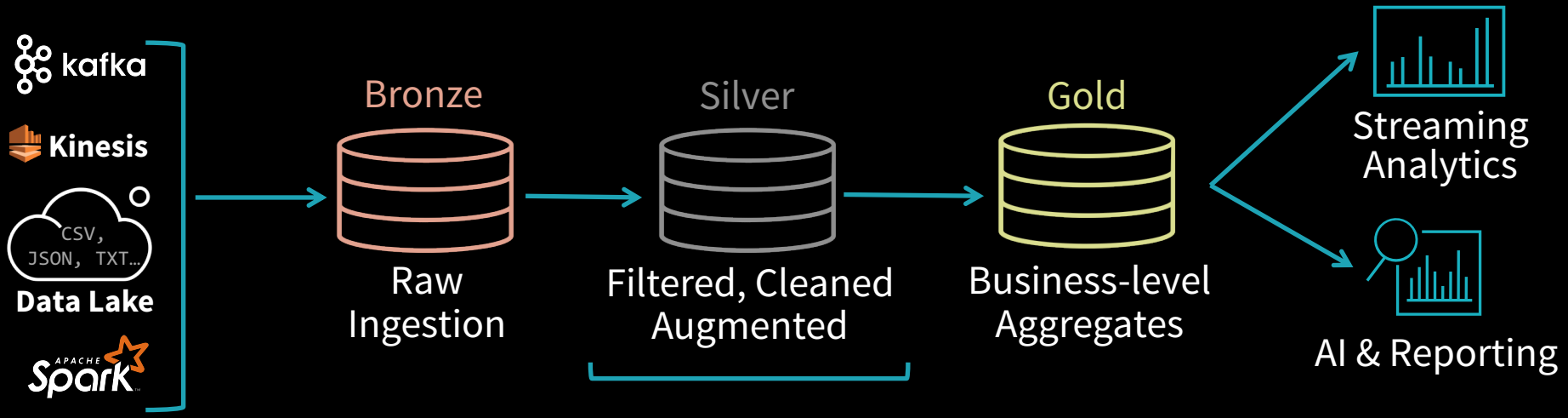
Delta Lake allows you to *incrementally* improve the quality of your data until it is ready for consumption.

The DELTA LAKE



- Dumping ground for raw data
- Often with long retention (years)
- Avoid error-prone parsing

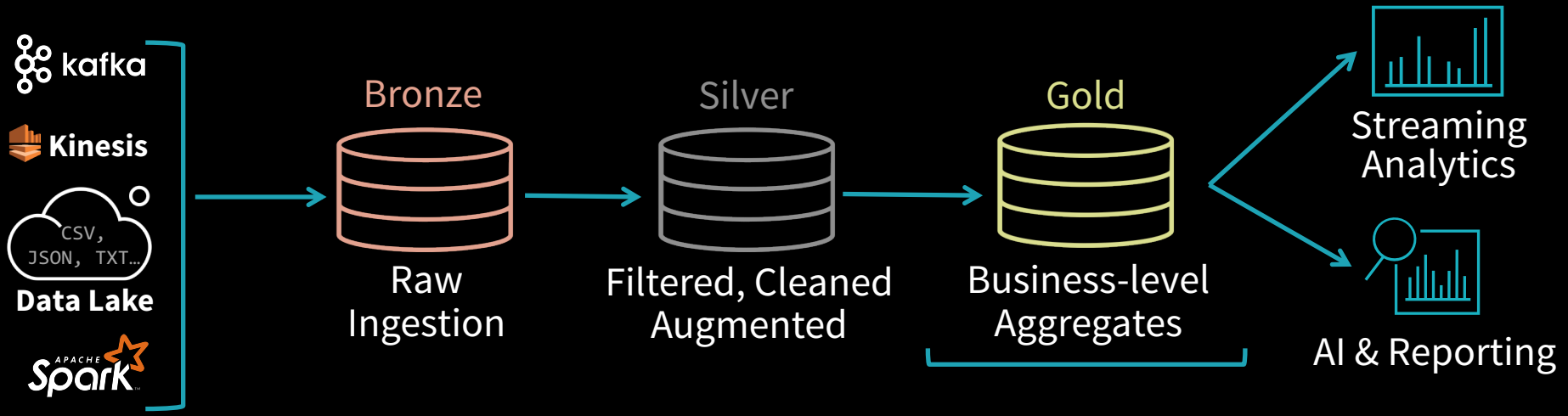
The DELTA LAKE



Intermediate data with some cleanup applied.

Queryable for easy debugging!

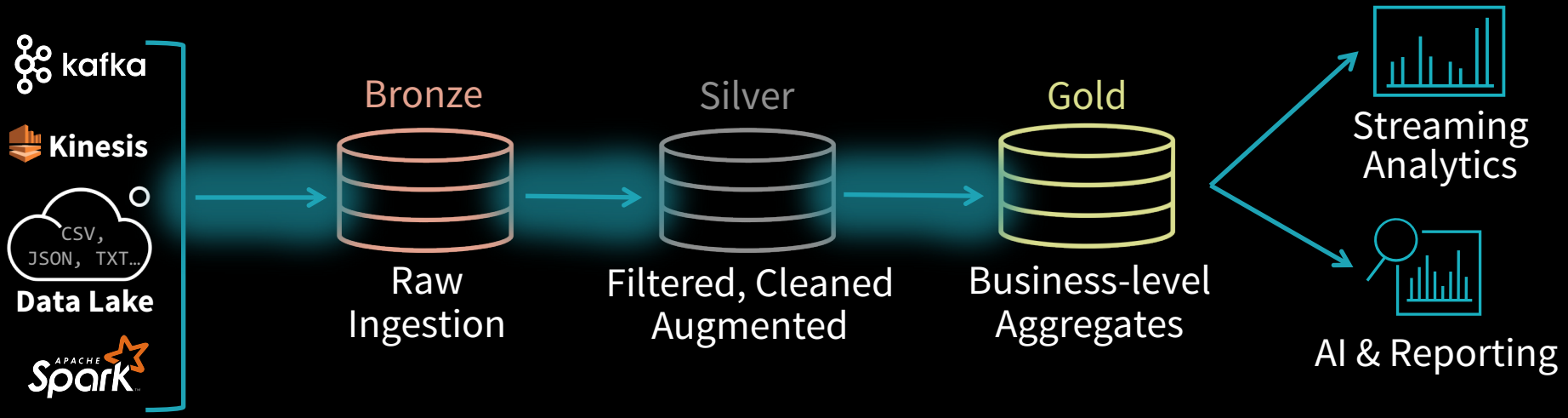
The DELTA LAKE



Clean data, ready for consumption.

Read with Spark or Presto

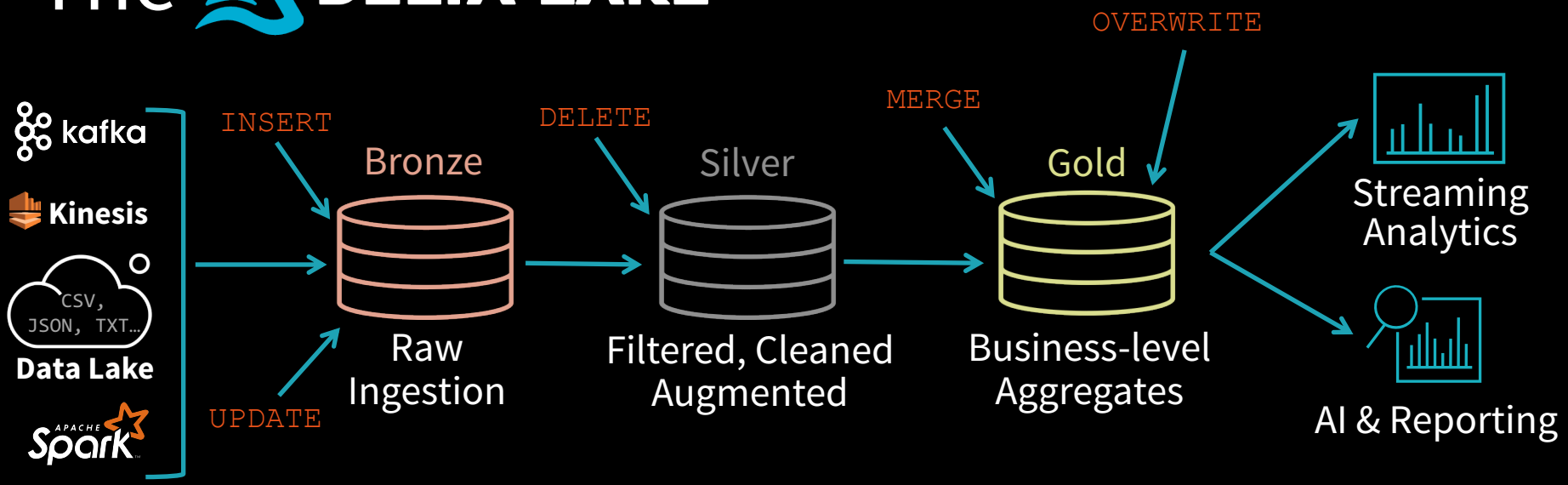
The DELTA LAKE



Streams move data through the Delta Lake

- Low-latency or manually triggered
- Eliminates management of schedules and jobs

The DELTA LAKE

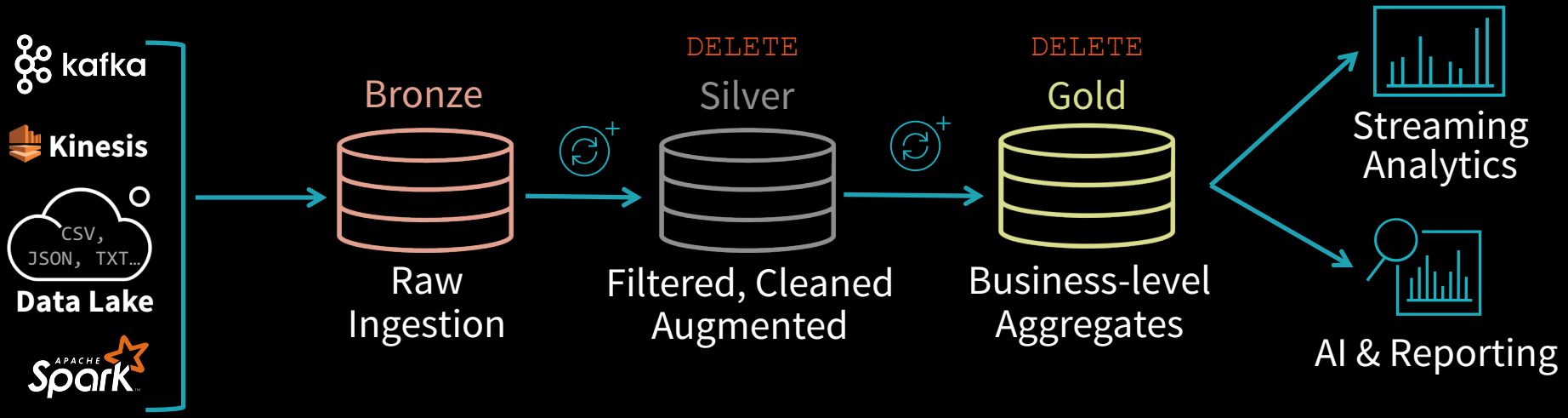


Delta Lake also supports batch jobs and standard DML

- Retention
- Corrections
- GDPR
- UPSERTS

databricks

The DELTA LAKE



Easy to recompute when business logic changes:

- Clear tables
- Restart streams

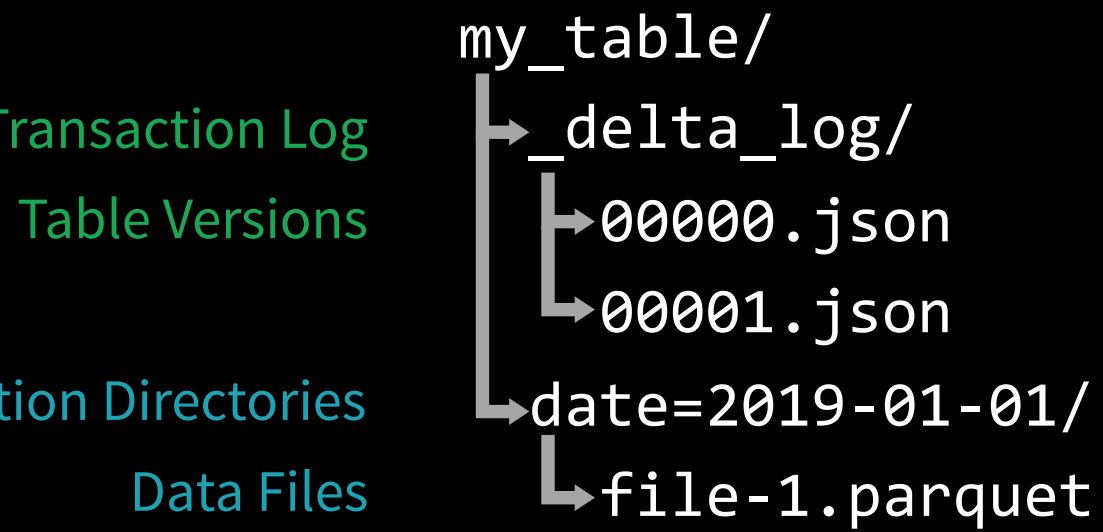
Delta under the hood

Transaction Log

Table Versions

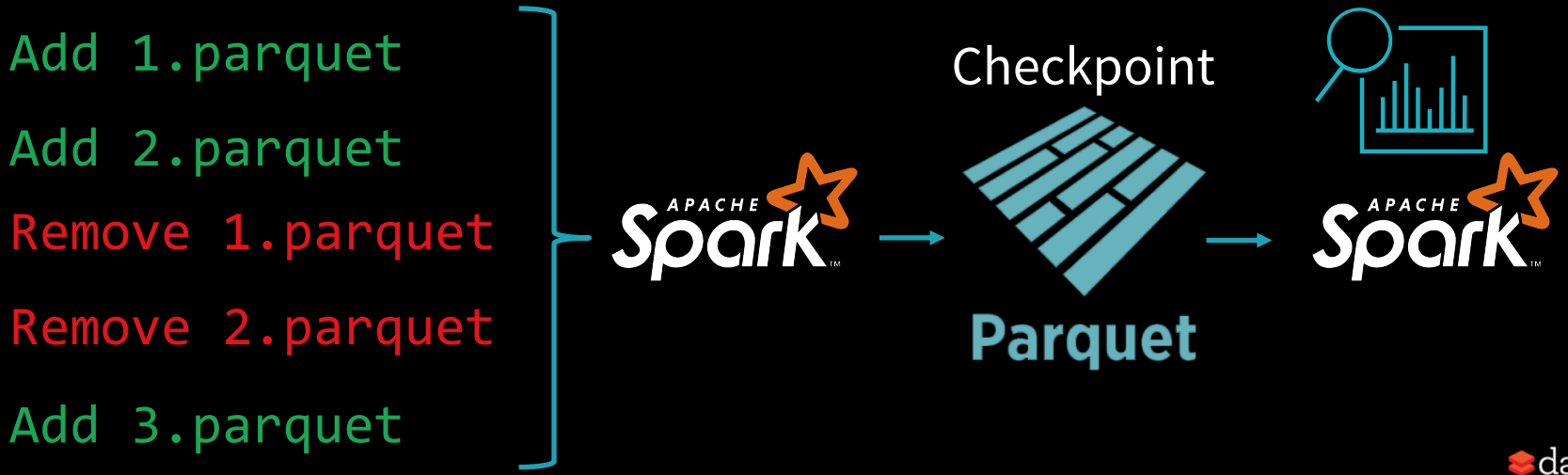
(Optional) Partition Directories

Data Files



Handling Massive Metadata

Large tables can have millions of files in them! How do we scale the metadata? Use Spark for scaling!



Conclusion

Complex data applications can only be built with modern eng principles & practices, and SQL is a terrible choice.

We are building a new generation of tools to incorporate the best of SQL & support modern eng practices.

Spark + Delta Lake are just the beginning. Hiring in SF, AMS, YYZ.