Big Data Software: What’s Next?

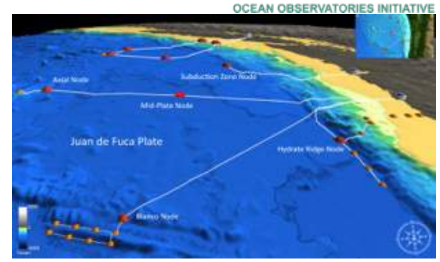
Michael Franklin
BDTC Beijing
December 2017

THE UNIVERSITY OF CHICAGO

Big Data = Nearly every field of endeavor is transitioning from “data poor” to “data rich”

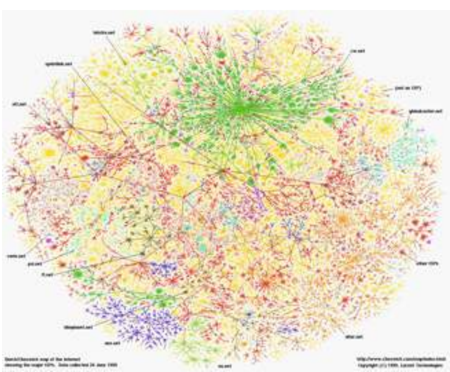
Astronomy: LSST

Physics: LHC



Oceanography

Neuroscience: EEG, fMRI



Sociology: The Web

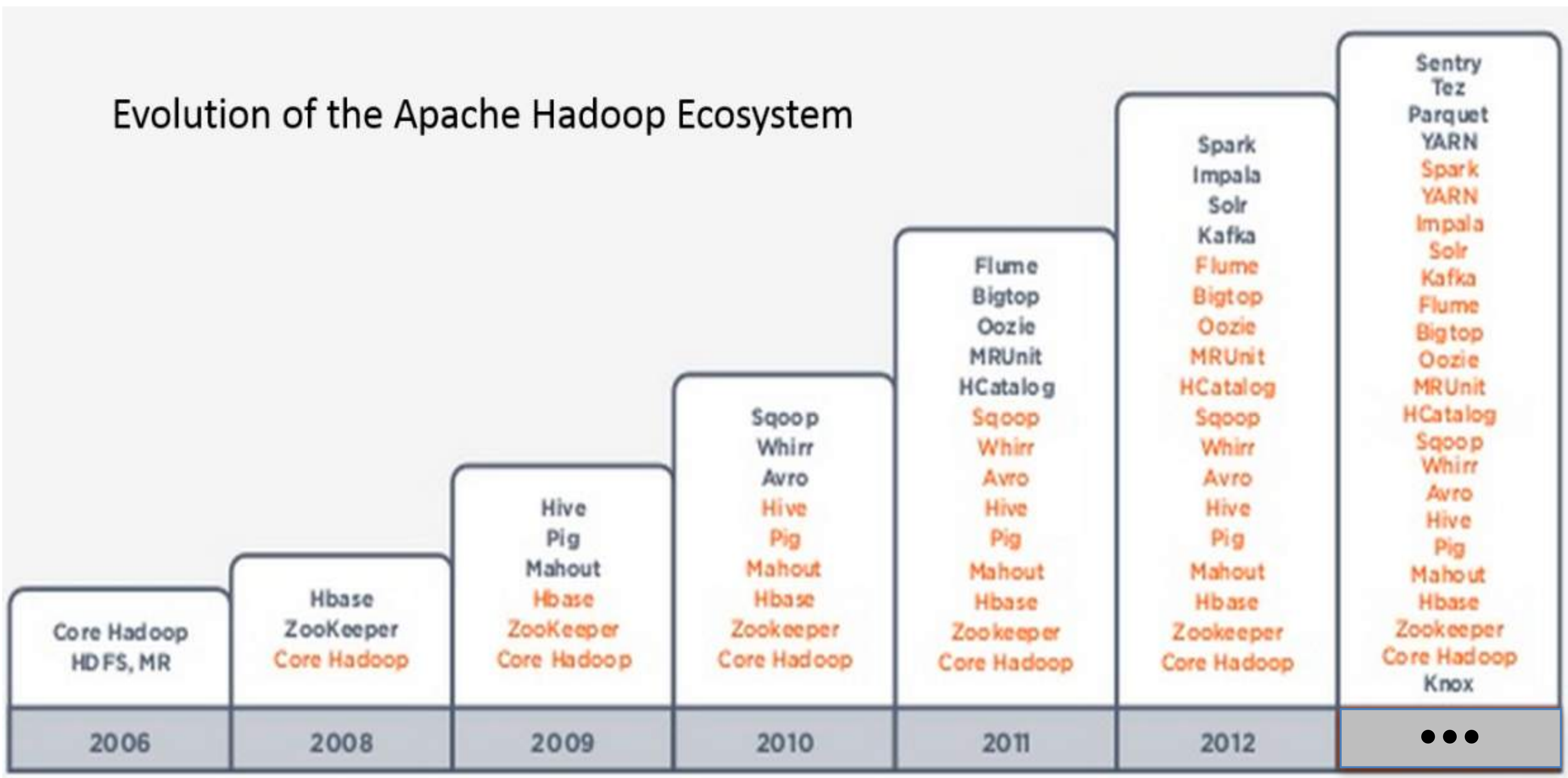
Biology: Sequencing

Economics: mobile, POS terminals

Data-Driven Medicine

Sports

Open Source Ecosystem & Context



Open Source Ecosystem & Context

2006-2010
Autonomic Computing & Cloud

Usenix HotCloud Workshop 2010

Spark: Cluster Computing with Working Sets

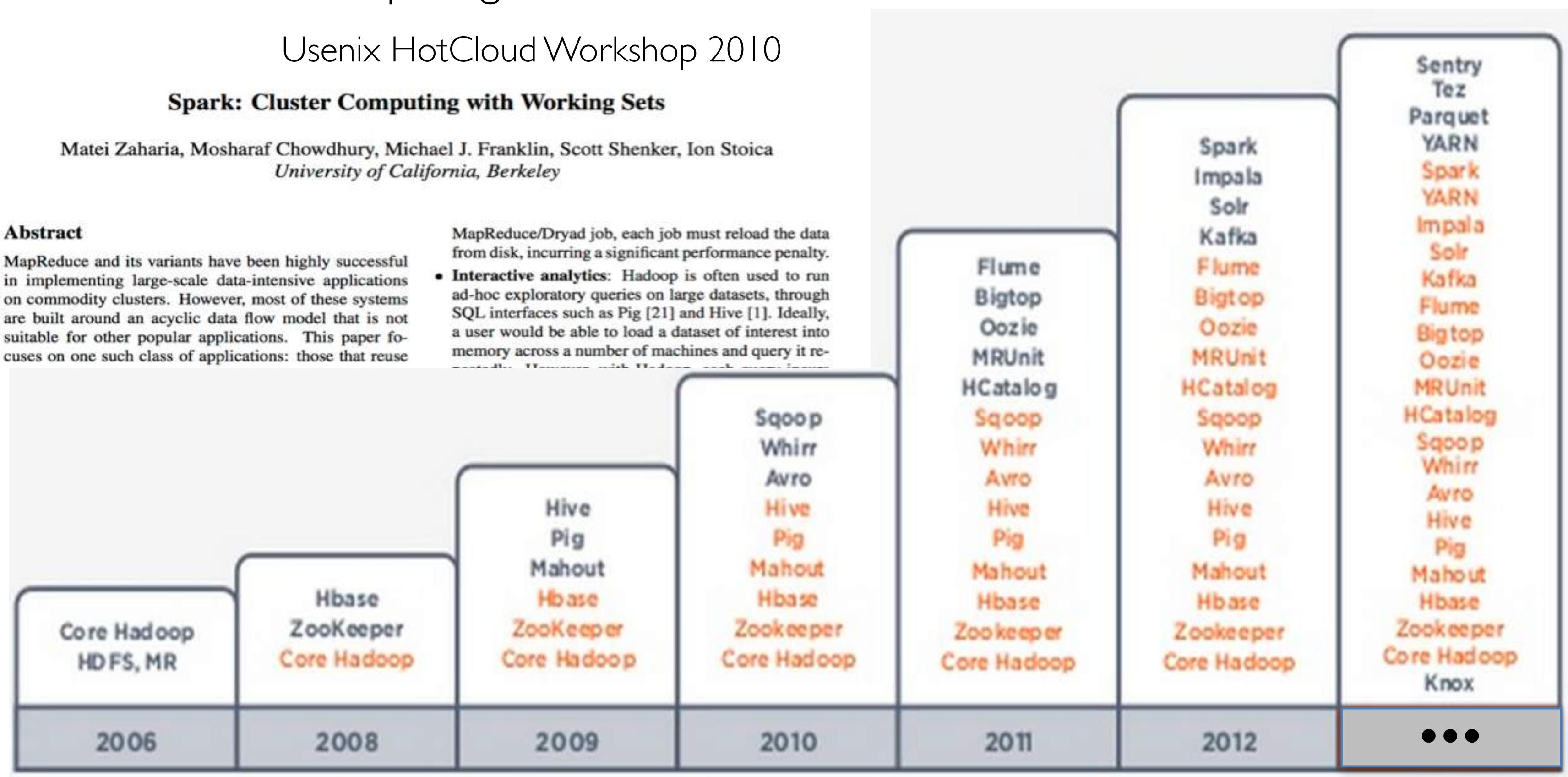
Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, Ion Stoica
University of California, Berkeley

Abstract

MapReduce and its variants have been highly successful in implementing large-scale data-intensive applications on commodity clusters. However, most of these systems are built around an acyclic data flow model that is not suitable for other popular applications. This paper focuses on one such class of applications: those that reuse

MapReduce/Dryad job, each job must reload the data from disk, incurring a significant performance penalty.

- **Interactive analytics**: Hadoop is often used to run ad-hoc exploratory queries on large datasets, through SQL interfaces such as Pig [21] and Hive [1]. Ideally, a user would be able to load a dataset of interest into memory across a number of machines and query it re-



Open Source Ecosystem & Context

2006-2010
Autonomic Computing & Cloud

2011-2016
Big Data Analytics

Usenix HotCloud Workshop 2010

Spark: Cluster Computing with Working Sets

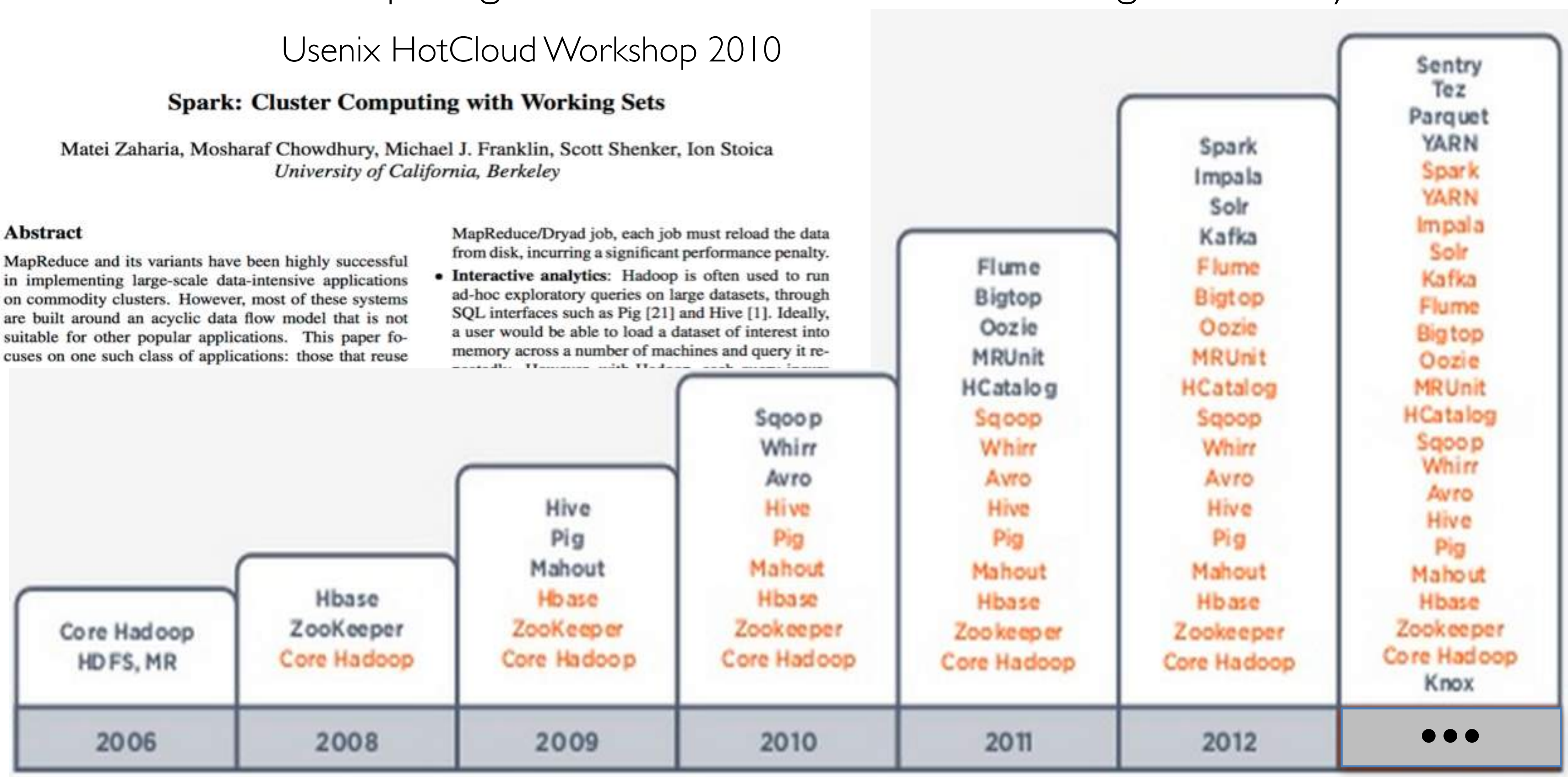
Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, Ion Stoica
University of California, Berkeley

Abstract

MapReduce and its variants have been highly successful in implementing large-scale data-intensive applications on commodity clusters. However, most of these systems are built around an acyclic data flow model that is not suitable for other popular applications. This paper focuses on one such class of applications: those that reuse

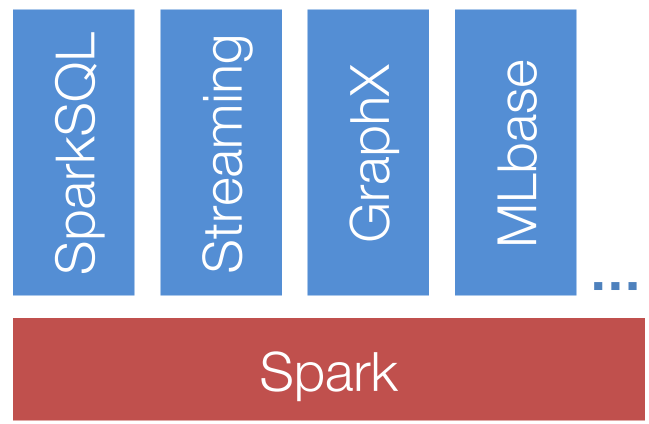
MapReduce/Dryad job, each job must reload the data from disk, incurring a significant performance penalty.

- **Interactive analytics:** Hadoop is often used to run ad-hoc exploratory queries on large datasets, through SQL interfaces such as Pig [21] and Hive [1]. Ideally, a user would be able to load a dataset of interest into memory across a number of machines and query it re-



Spark's Philosophy

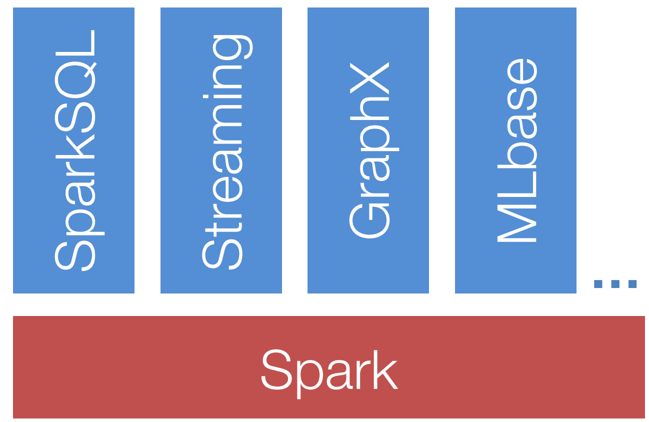
- Specializing MapReduce leads to incompatible, stovepiped systems
- Instead, **generalize** MapReduce:



Spark's Philosophy

- Specializing MapReduce leads to incompatible, stovepiped systems
- Instead, **generalize** MapReduce:

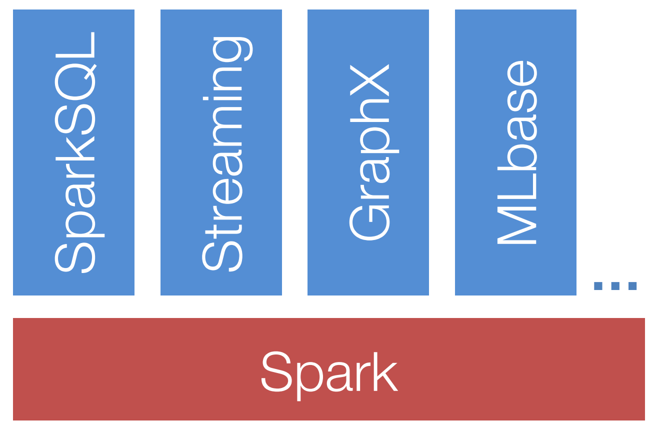
  1. Richer Programming Model
     ➔More operators than map and reduce



Spark's Philosophy

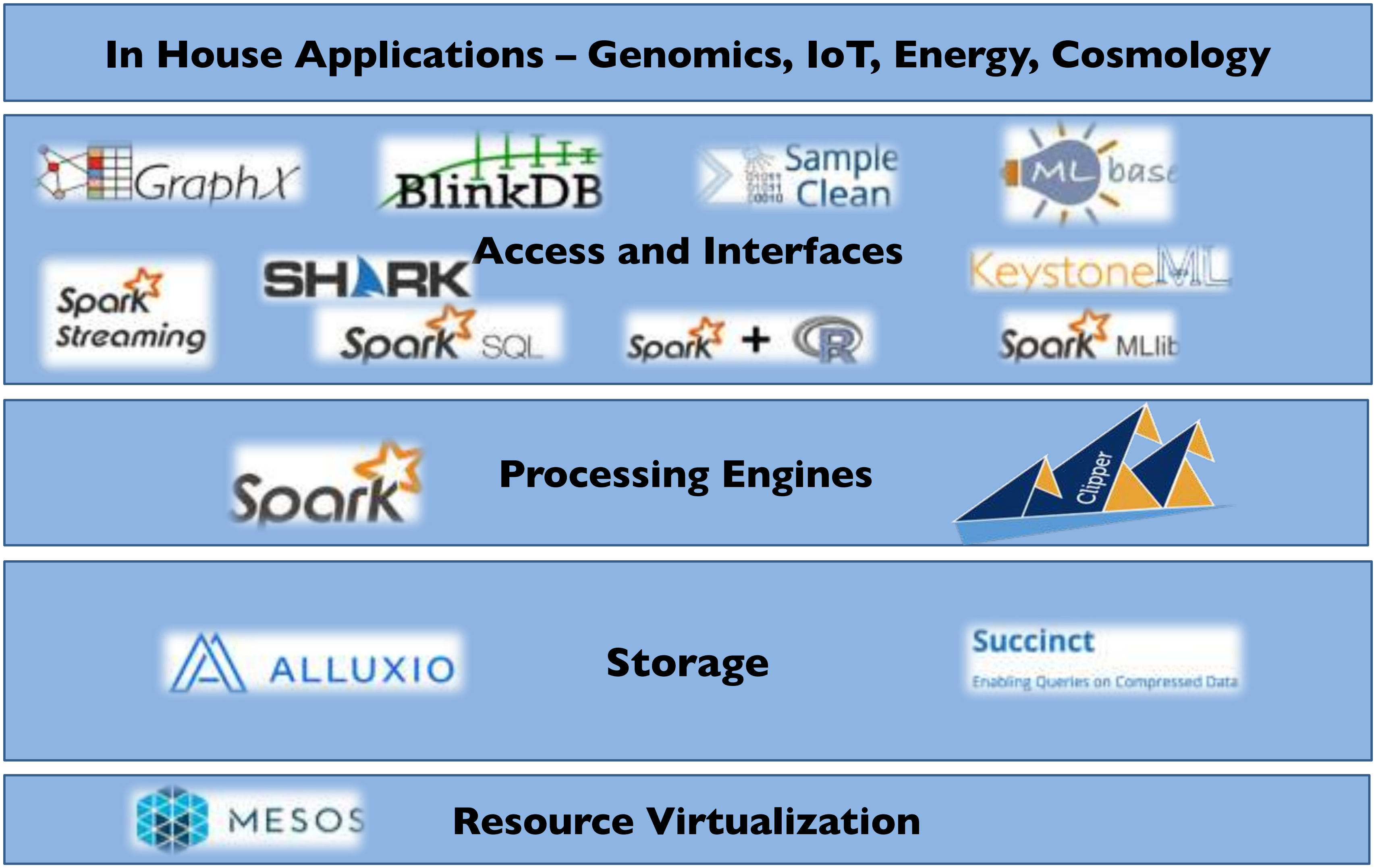
- Specializing MapReduce leads to incompatible, stovepiped systems
- Instead, **generalize** MapReduce:

1. Richer Programming Model
   ➔ More operators than map and reduce



2. Memory Management
   ➔ Less data movement leads to better performance for complex analytics

Berkeley Data Analytics Stack

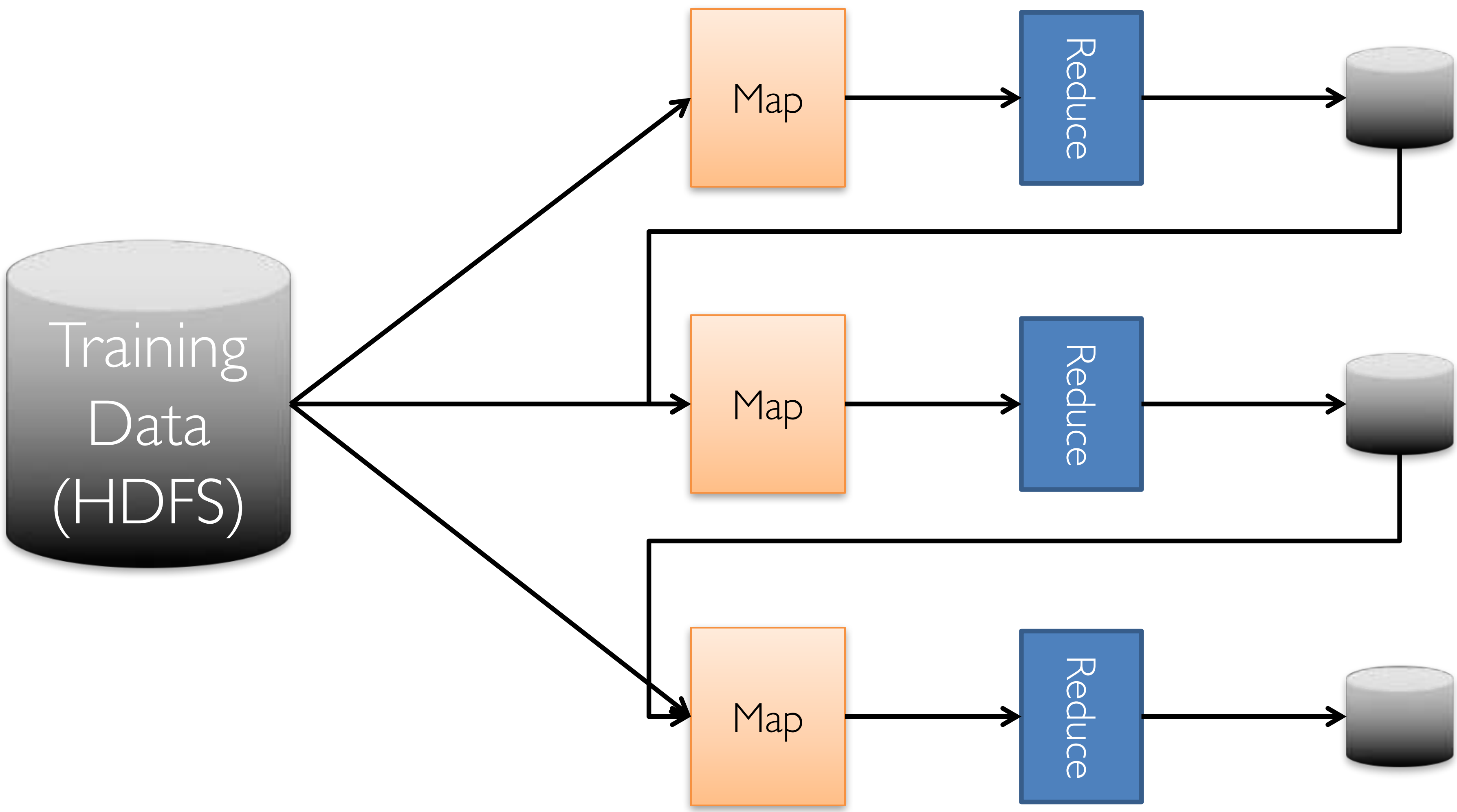


Apache Spark Meetups (Dec 2017)

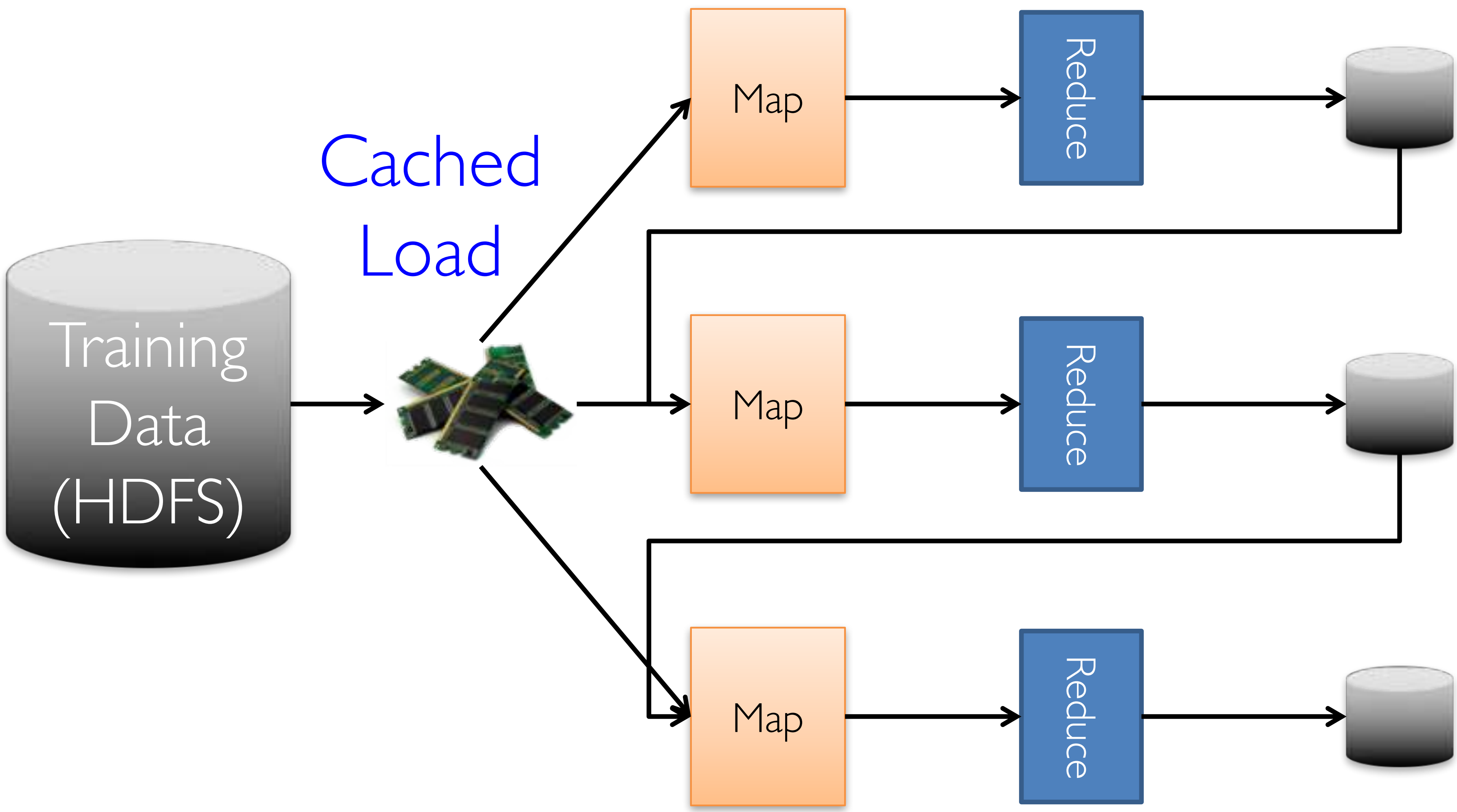
635 groups with 452,749 **members**

spark.meetup.com

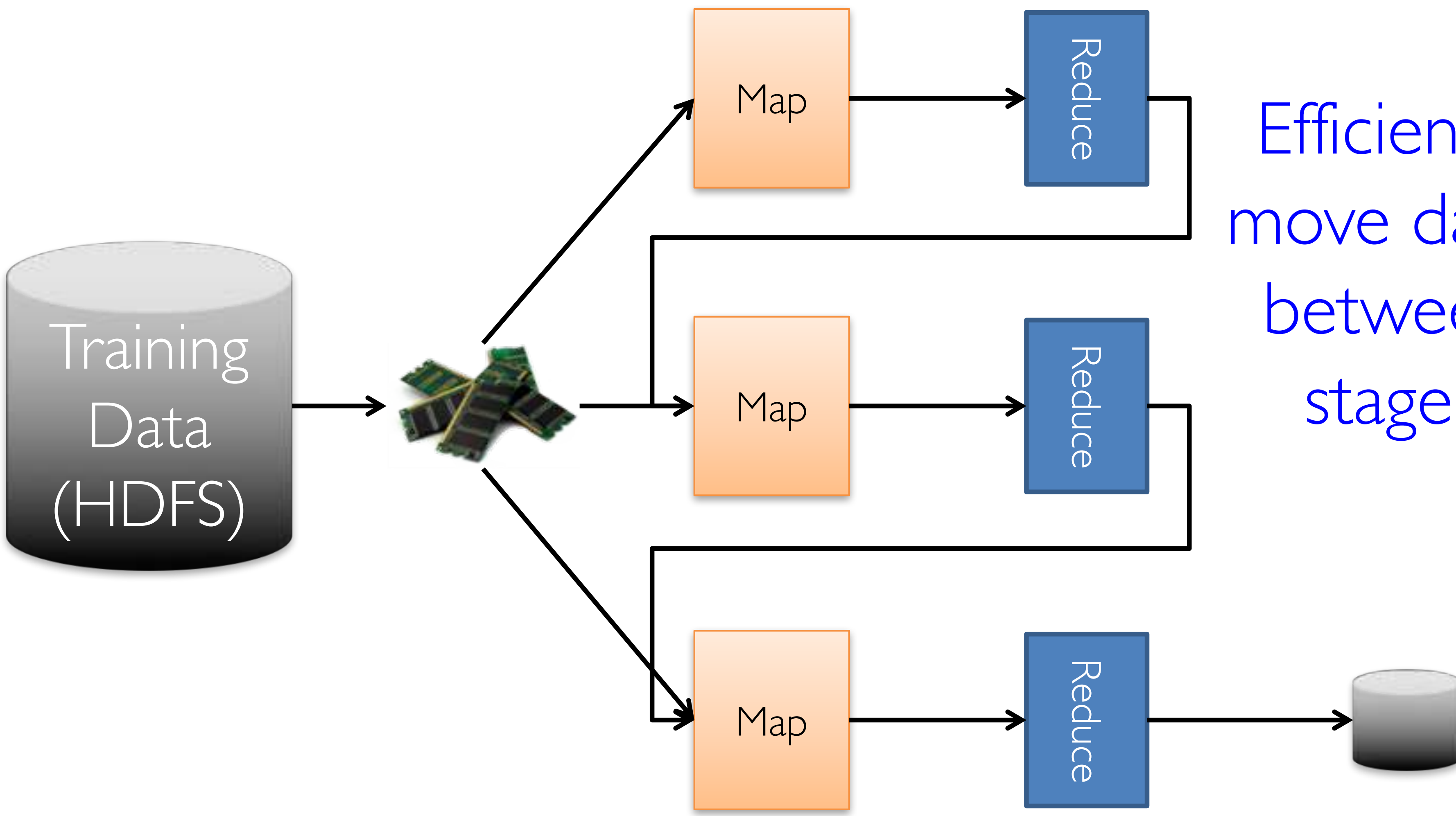
Memory Mgmt in Hadoop MR



Memory Management in Spark

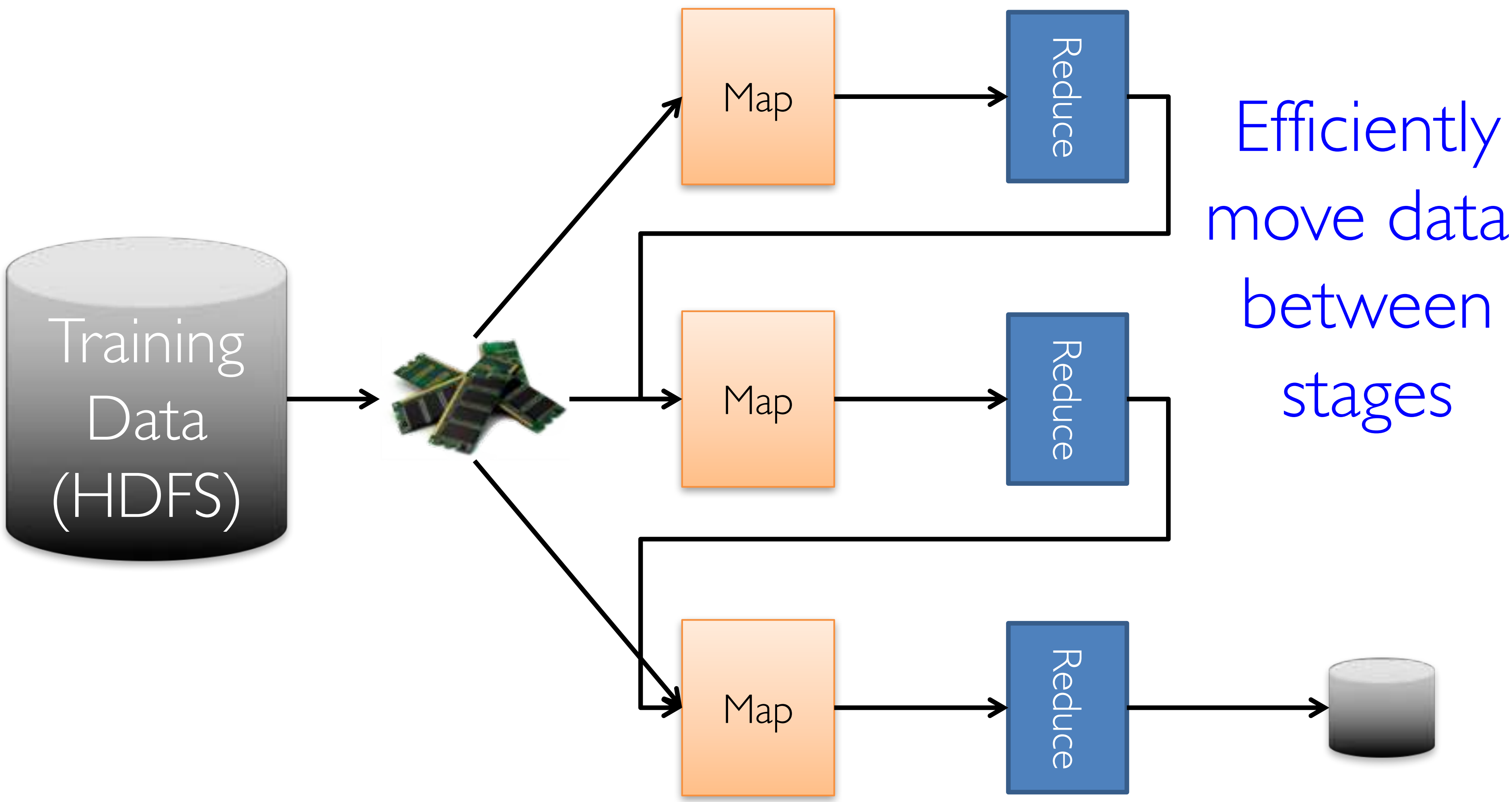


Memory Management in Spark



Efficiently move data between stages

Memory Management in Spark



10-100× speed up vs.Hadoop MapReduce
with no HDFS data migration needed

Lineage for Fault Tolerance

RDDs: **Immutable** collections of objects that can be stored in memory or disk across a cluster

- Built via parallel transformations (map, filter, …)
- Automatically rebuilt on (partial) failure

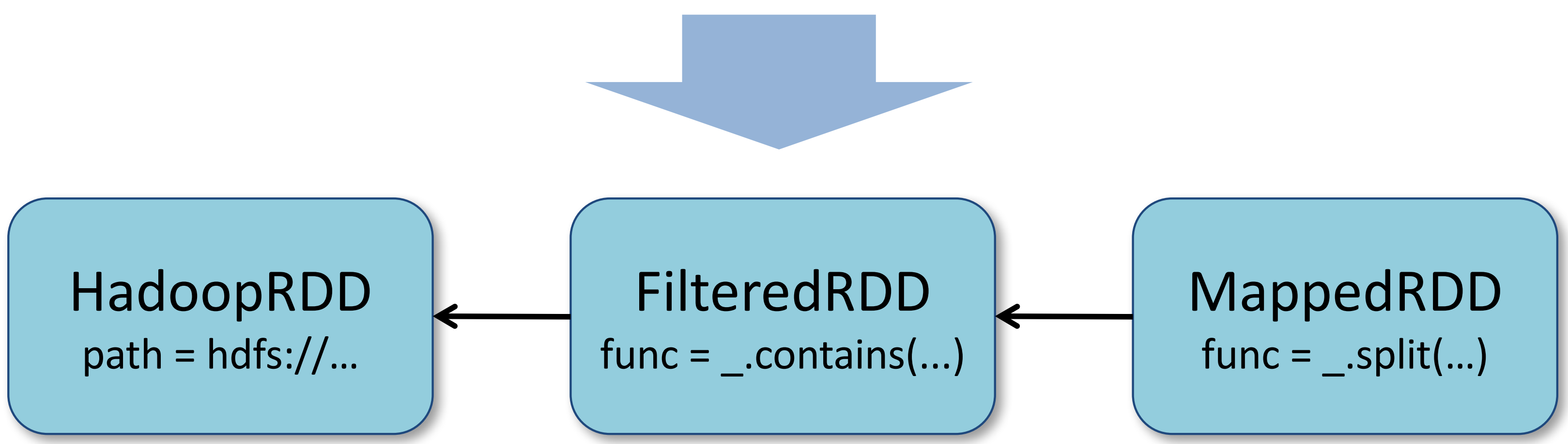
M. Zaharia, et al, Resilient Distributed Datasets: A fault-tolerant abstraction for in-memory cluster computing, NSDI 2012.

Lineage for Fault Tolerance

RDDs: **Immutable** collections of objects that can be stored in memory or disk across a cluster

- Built via parallel transformations (map, filter, …)
- Automatically rebuilt on (partial) failure

```
messages = textFile(...).filter(_.contains("error"))
                        .map(_.split('\t')(2))
```



M. Zaharia, et al, Resilient Distributed Datasets: A fault-tolerant abstraction for in-memory cluster computing, NSDI 2012.

SQL and DataFrame Support

SQL increasingly supported by Big Data platforms:

- Apache Drill, Flink, Hive, Kafka, Spark, Cloudera Impala, HAWQ, IBM Big SQL, Presto, …

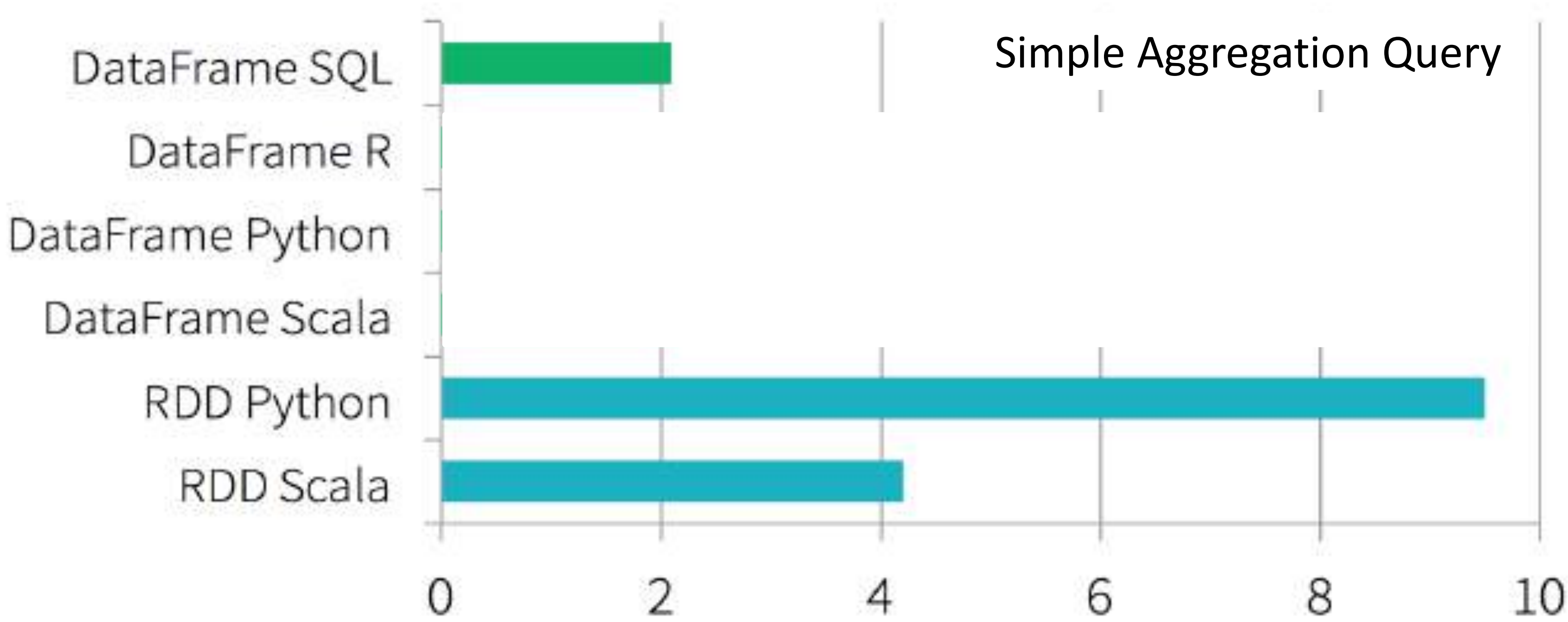
Spark supports SQL and also “Dataframes”:

```
people.filter("age > 30")
    .join(dept, people("deptId") === dept("id"))
    .groupBy(dept("name"), "gender")
    .agg(avg(people("salary")), max(people("age")))
```

M. Armbrust, et al, Spark SQL: Relational Data Processing in Spark, SIGMOD 2015.

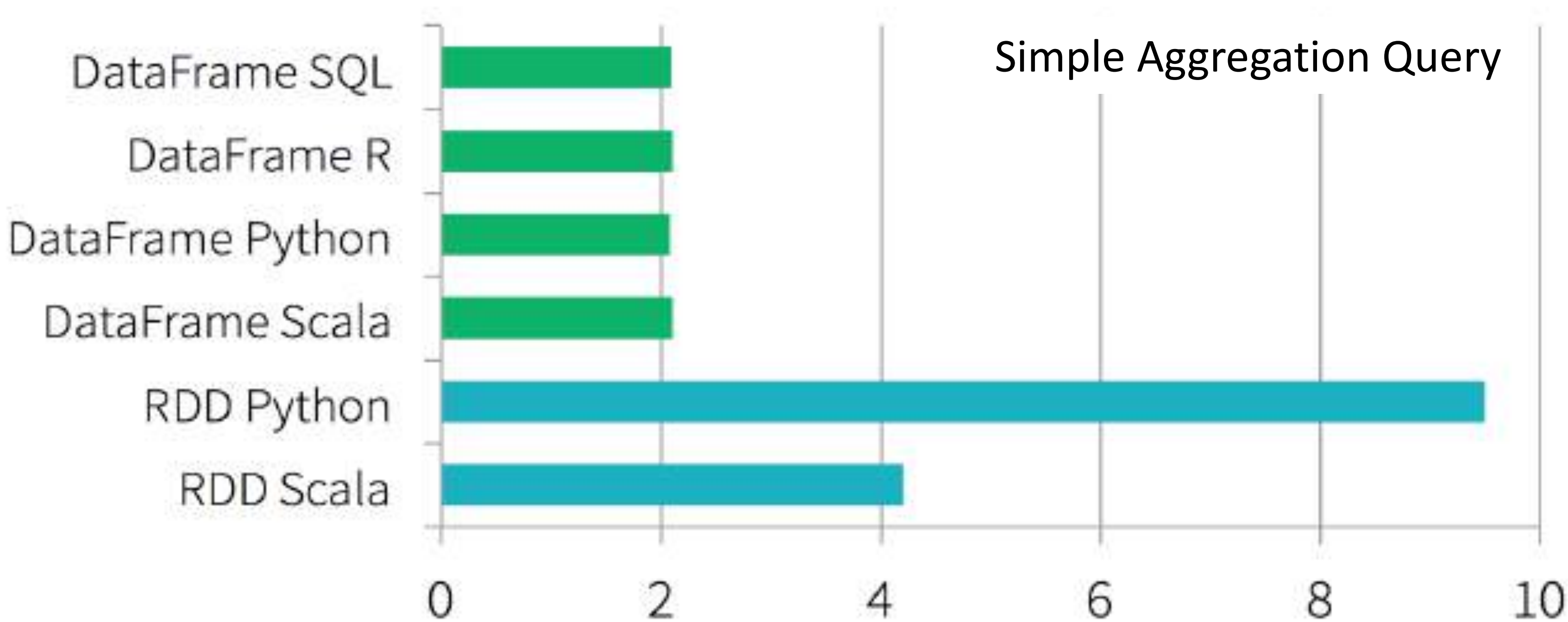
SparkSQL/Catalyst Optimizer

- Typical DB optimizations across SQL and Dataframes
  - Extensibility via Optimization Rules written in Scala
  - Open Source optimizer evolution!
- Code Generation (inner loops and iterator removal)
- Cost-based (as of V2.2)



SparkSQL/Catalyst Optimizer

- Typical DB optimizations across SQL and Dataframes
  - Extensibility via Optimization Rules written in Scala
  - Open Source optimizer evolution!
- Code Generation (inner loops and iterator removal)
- Cost-based (as of V2.2)



Putting it all Together: Multimodal Analytics

SQL

```
// Load historical data as an RDD using Spark SQL
val trainingData = sql(
  "SELECT location, language FROM old_tweets")
```

Machine Learning

```
// Train a K-means model using MLlib
val model = new KMeans()
  .setFeaturesCol("location")
  .setPredictionCol("language")
  .fit(trainingData)
```

Streaming

```
// Apply the model to new tweets in a stream
TwitterUtils.createStream(...)
  .map(tweet => model.predict(tweet.location))
```

Putting it all Together: Multimodal Analytics

SQL

```
// Load historical data as an RDD using Spark SQL
val trainingData = sql(
  "SELECT location, language FROM old_tweets")
```

Machine Learning

```
// Train a K-means model using MLlib
val model = new KMeans()
  .setFeaturesCol("location")
  .setPredictionCol("language")
  .fit(trainingData)
```

Streaming

```
// Apply the model to new tweets in a stream
TwitterUtils.createStream(...)
  .map(tweet => model.predict(tweet.location))
```

Putting it all Together: Multimodal Analytics

SQL

```
// Load historical data as an RDD using Spark SQL
val trainingData = sql(
  "SELECT location, language FROM old_tweets")
```

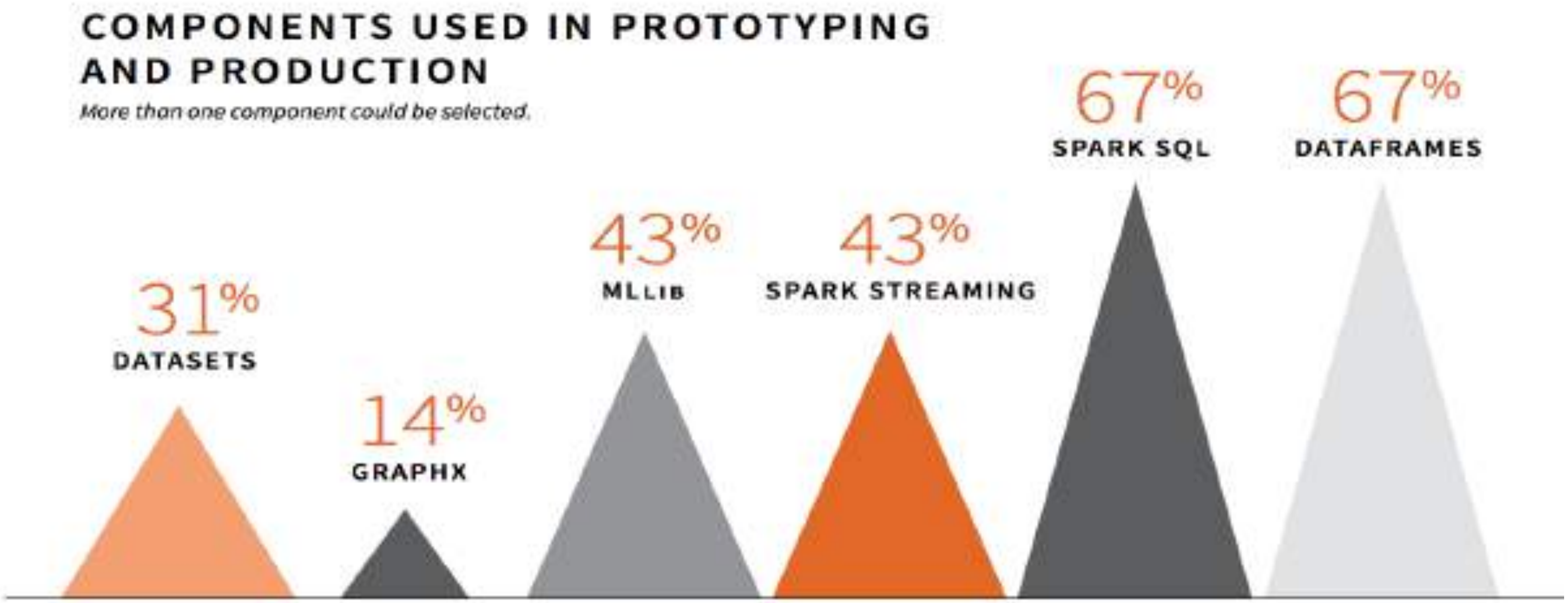
Machine Learning

```
// Train a K-means model using MLlib
val model = new KMeans()
  .setFeaturesCol("location")
  .setPredictionCol("language")
  .fit(trainingData)
```

Streaming

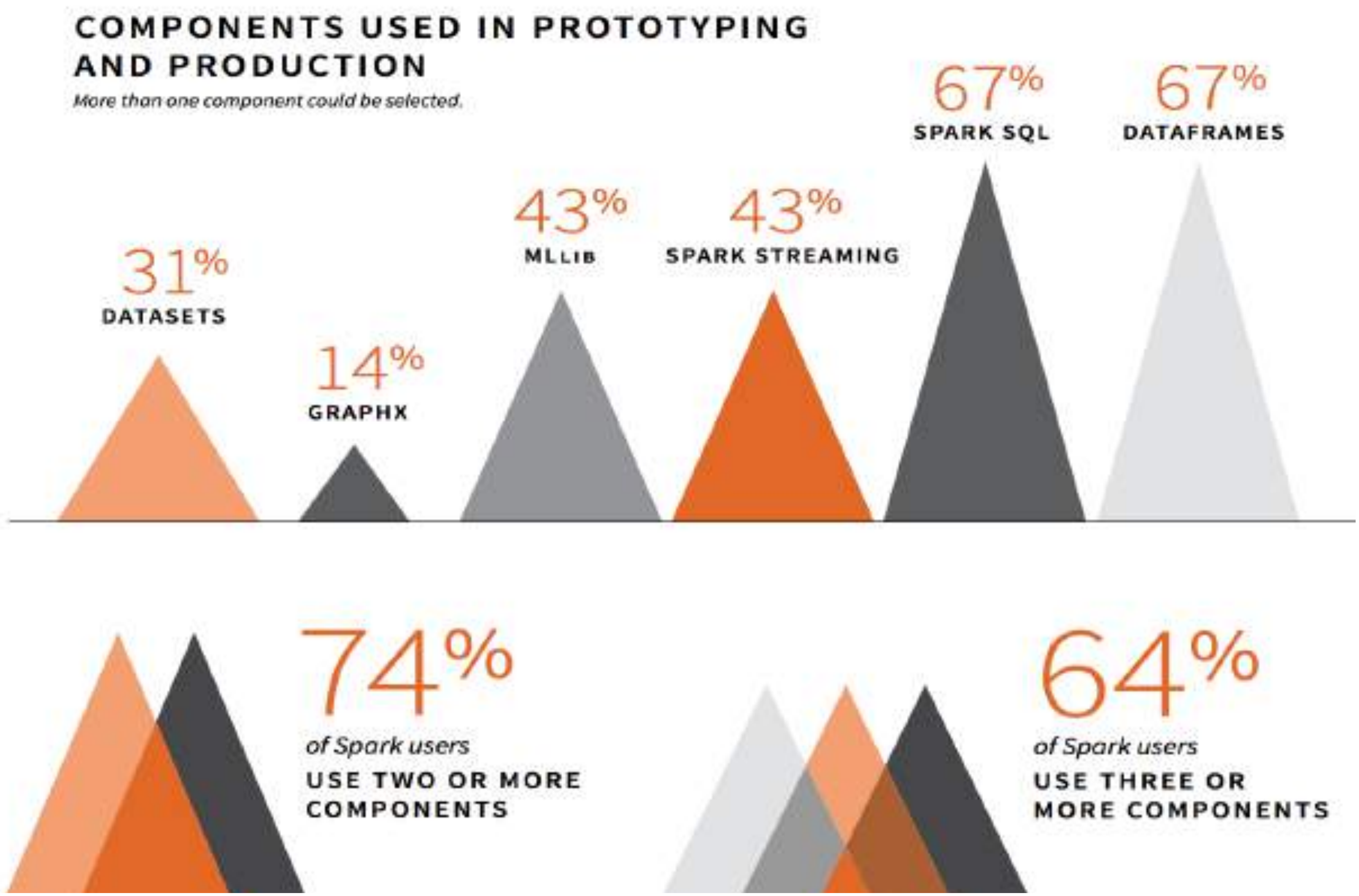
```
// Apply the model to new tweets in a stream
TwitterUtils.createStream(...)
  .map(tweet => model.predict(tweet.location))
```

Multimodal Advanced Analytics



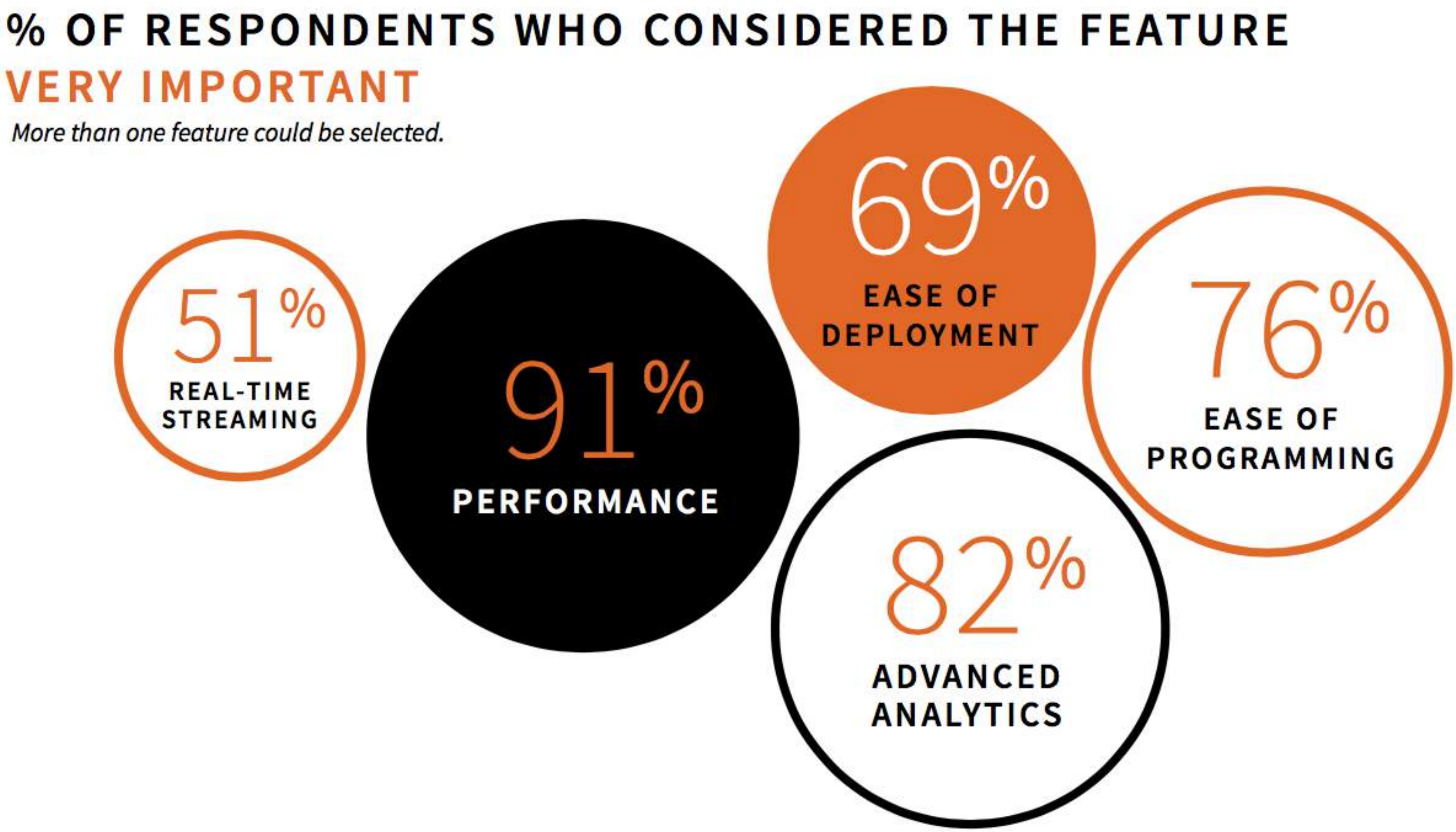
From: Spark User Survey 2016, 1615 respondents from 900 organizations
http://go.databricks.com/2016-spark-survey

Multimodal Advanced Analytics



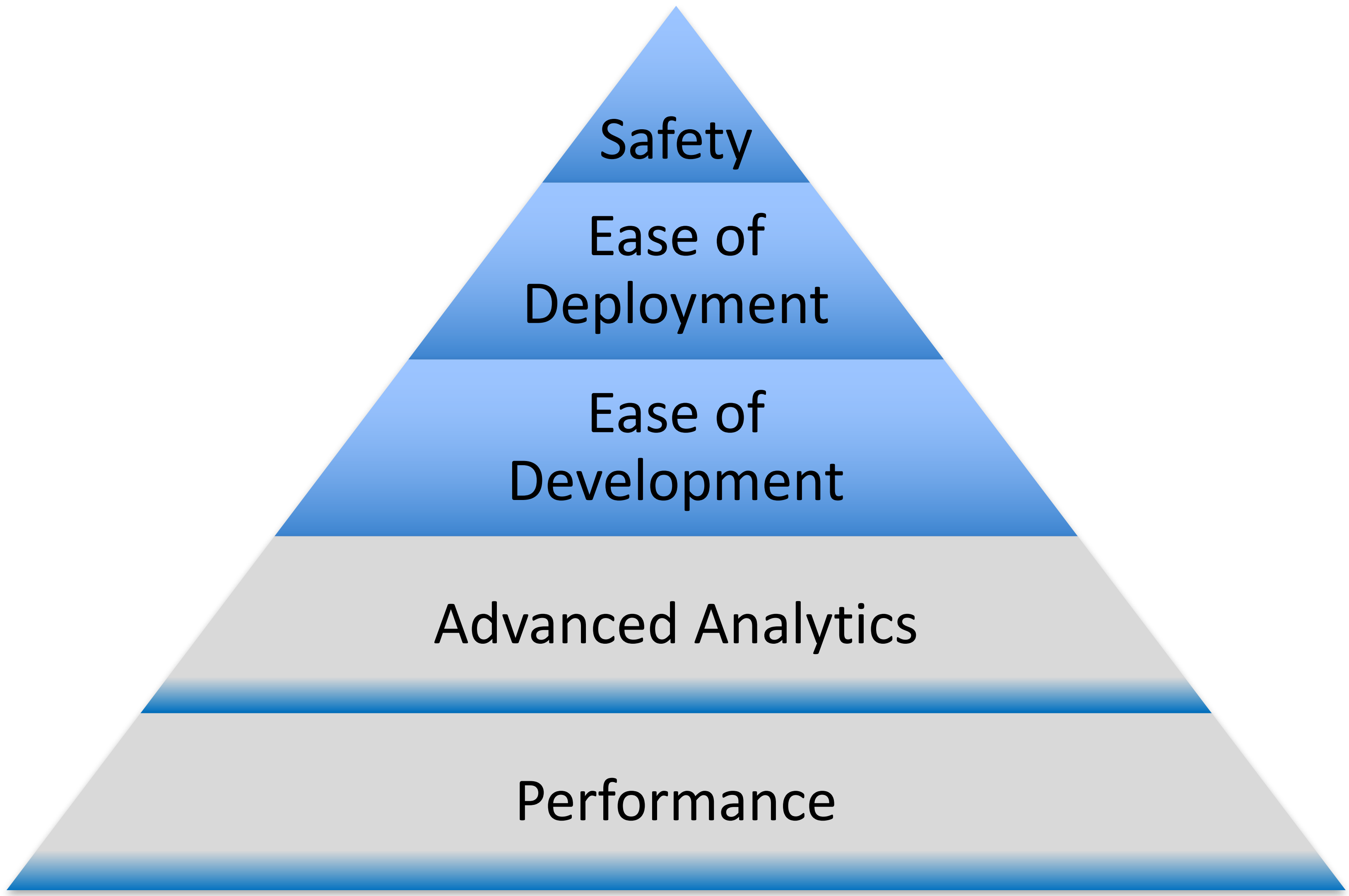
From: Spark User Survey 2016, 1615 respondents from 900 organizations
http://go.databricks.com/2016-spark-survey

What Do Users Want?



From: Spark User Survey 2016, 1615 respondents from 900 organizations
http://go.databricks.com/2016-spark-survey

Maslow's Hierarchy of Analytics?



What's Next?

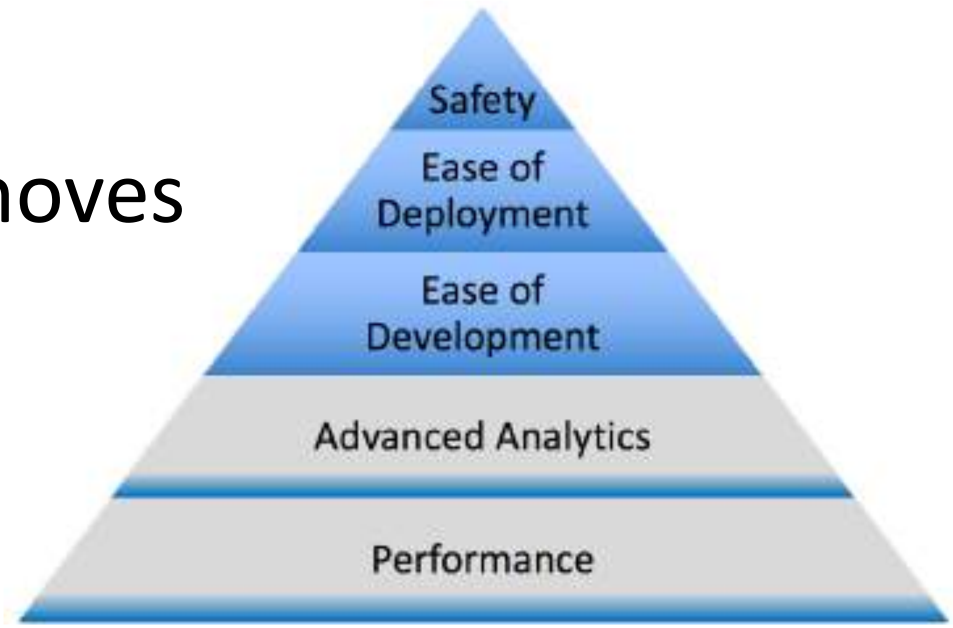
Rapidly changing hardware means that there is still a lot of research to be done in performance, scalability and fault tolerance

Likewise, new analytics approaches and AI techniques (e.g., Deep Learning) are becoming increasingly mainstream

Lots of work to be done in these areas, but…

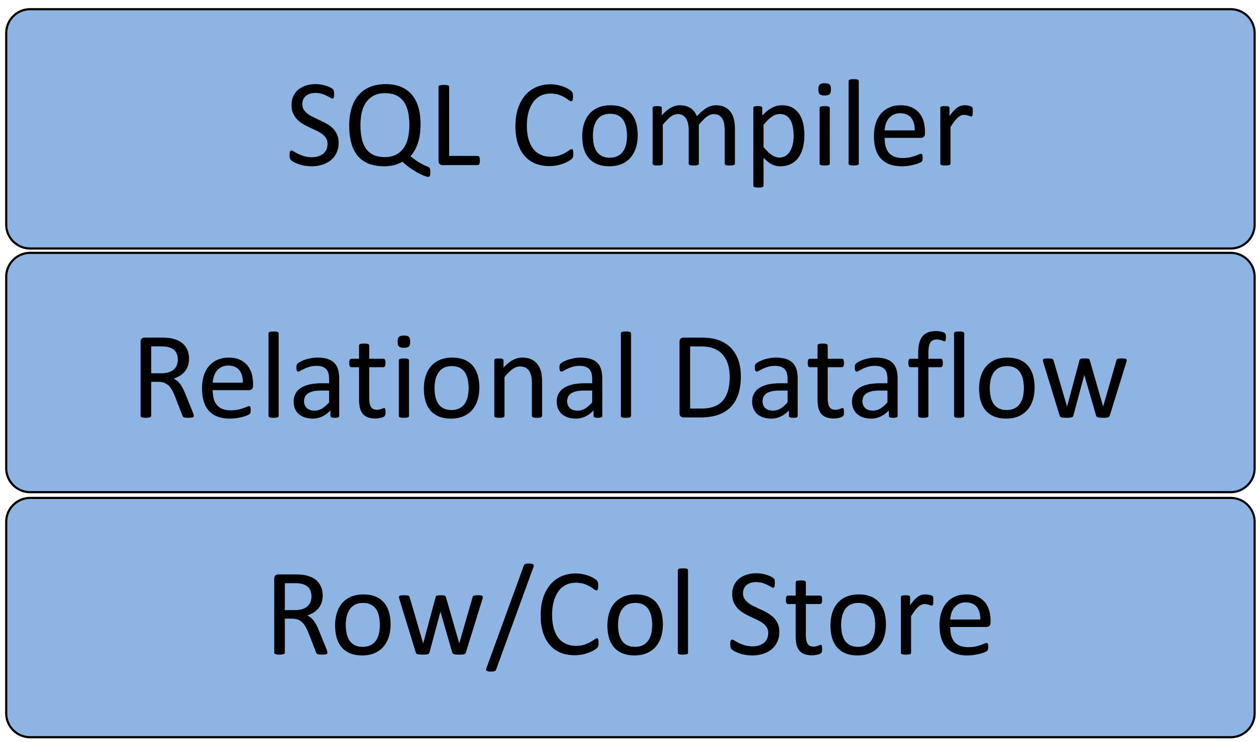
as we Move Up the Hierarchy…

a new set of concerns moves to the fore:



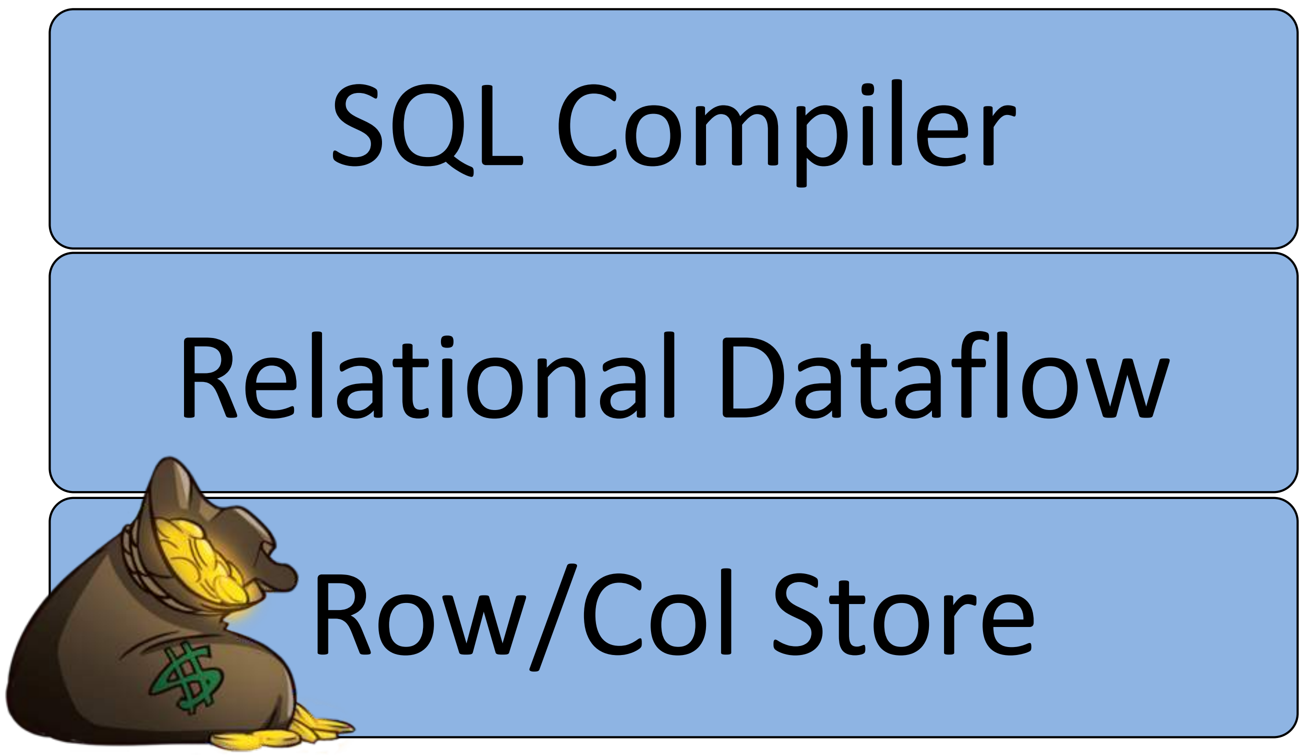
1) Reducing Friction:
   Ease of Development and Deployment
2) Data Science/Analytics Full Lifecycle Concerns
3) "Safe" Data Science and Human Factors

Database Systems: One way in/out



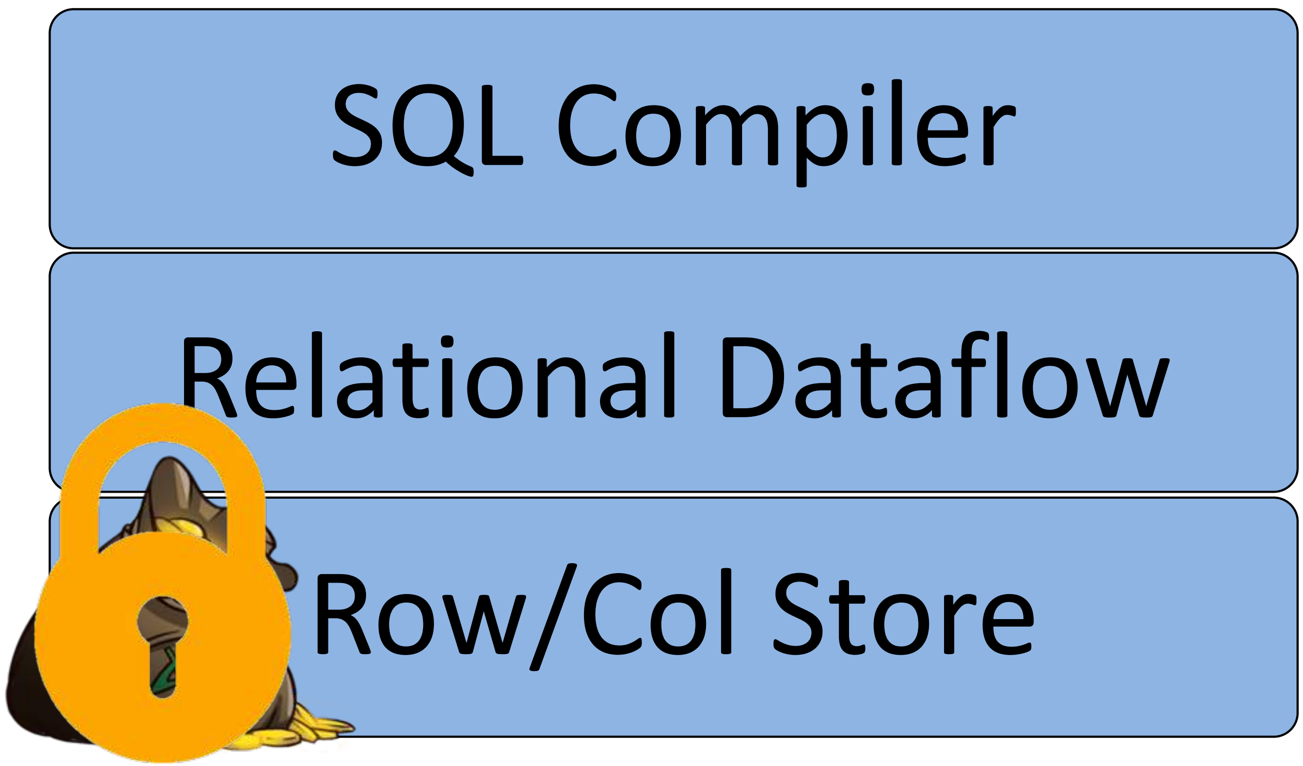
Adapted from Mike Carey, UCI

Database Systems: One way in/out



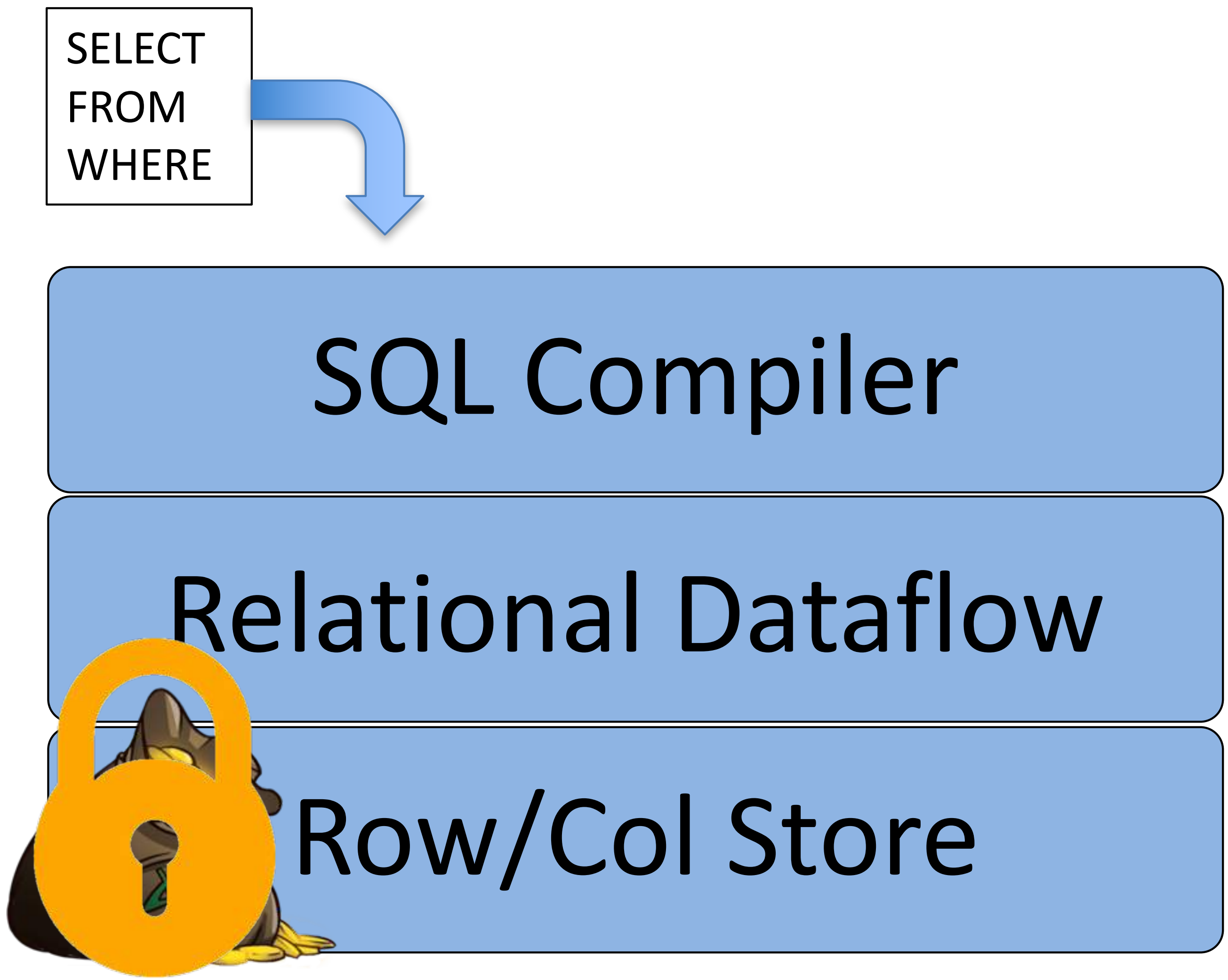
Adapted from Mike Carey, UCI

Database Systems: One way in/out



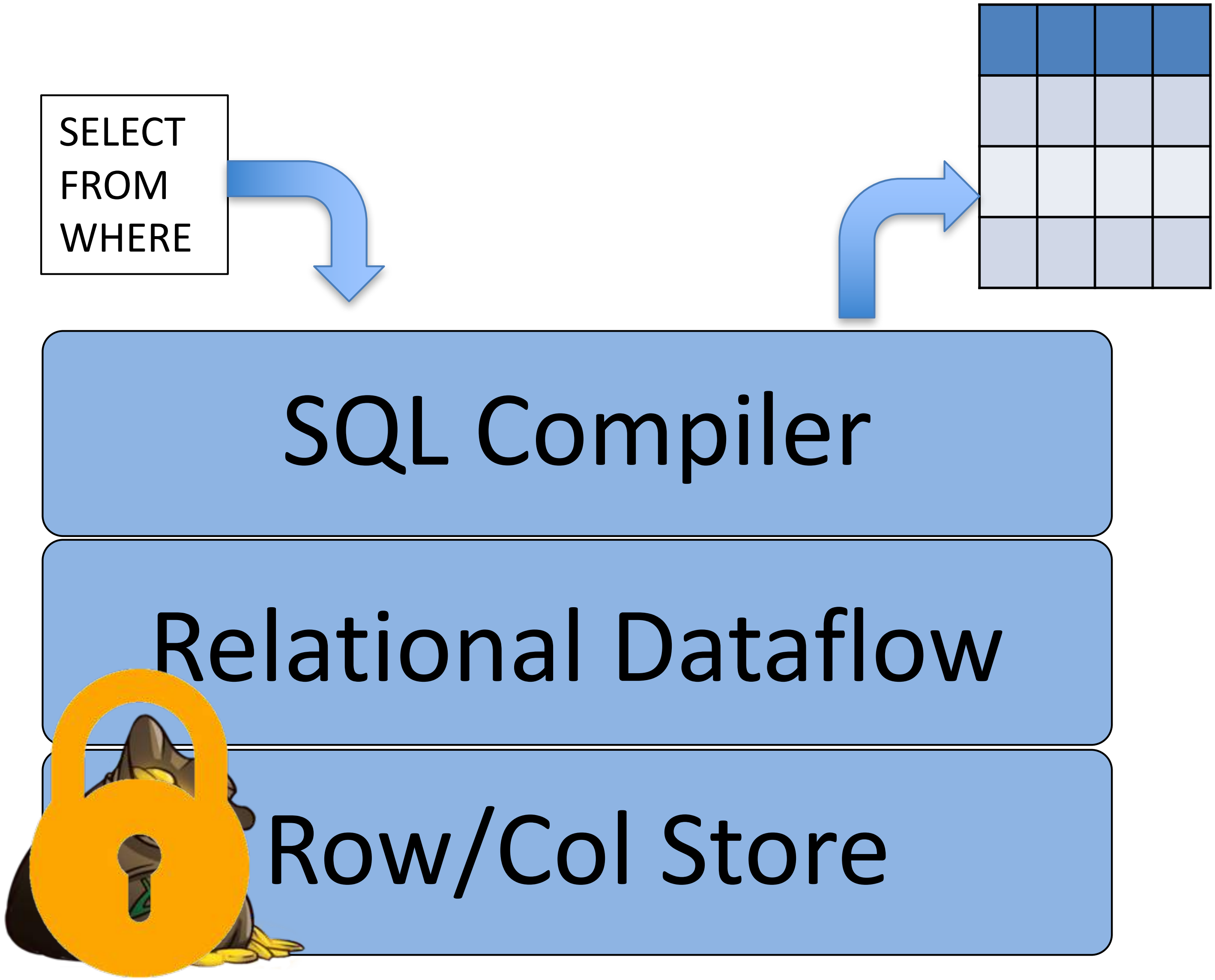
Adapted from Mike Carey, UCI

Database Systems: One way in/out



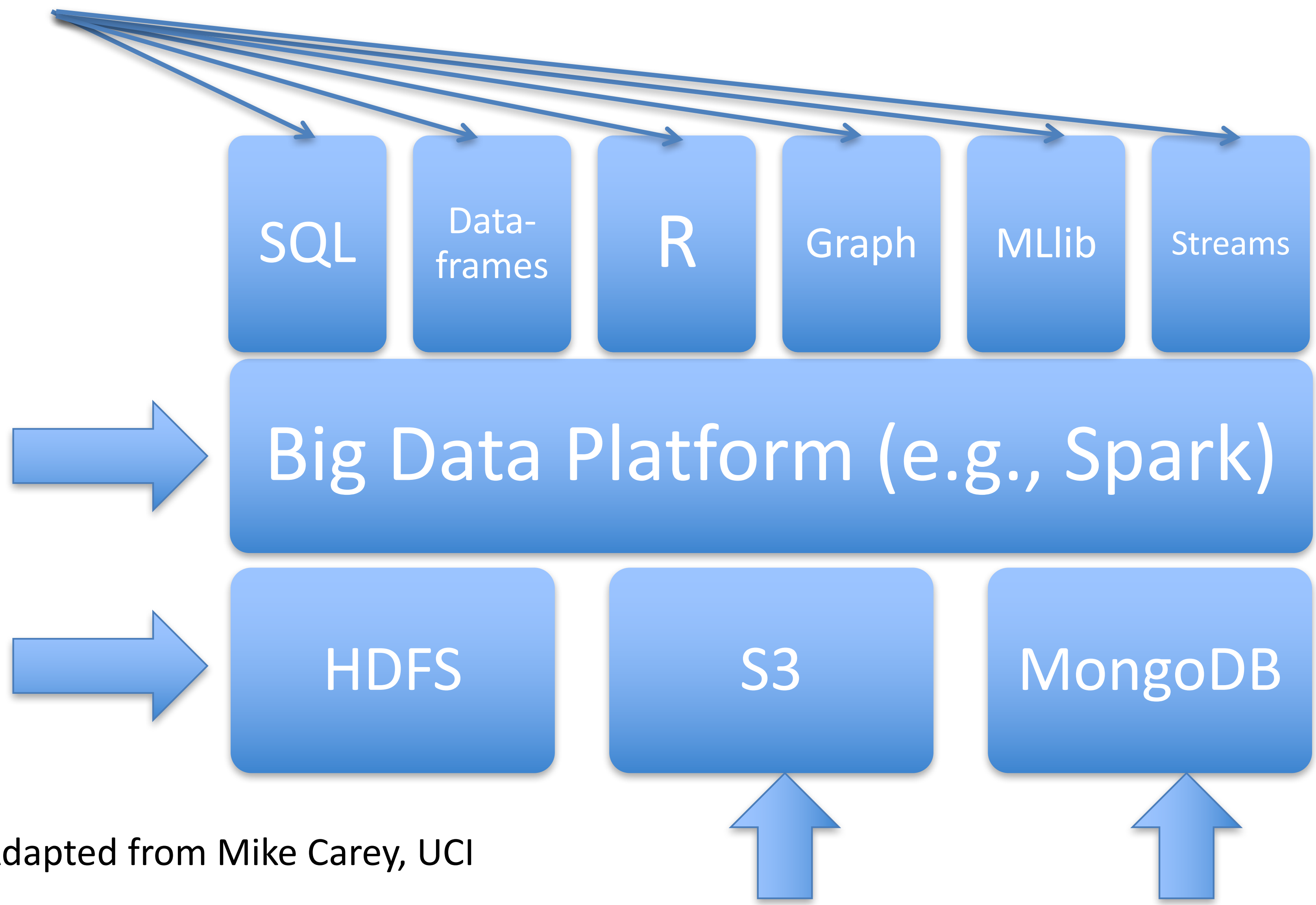
Adapted from Mike Carey, UCI

Database Systems: One way in/out



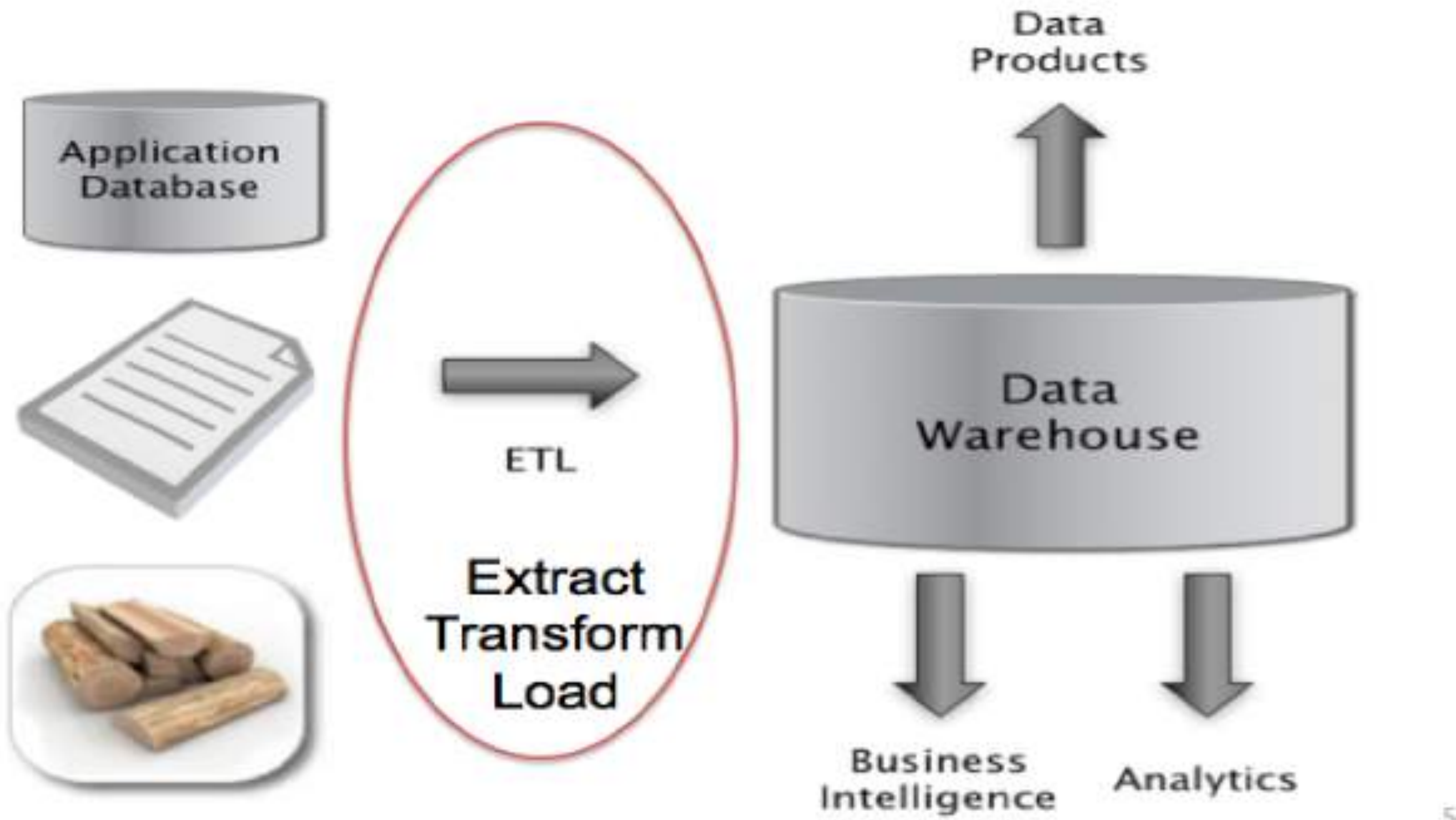
Adapted from Mike Carey, UCI

Reducing Data Friction



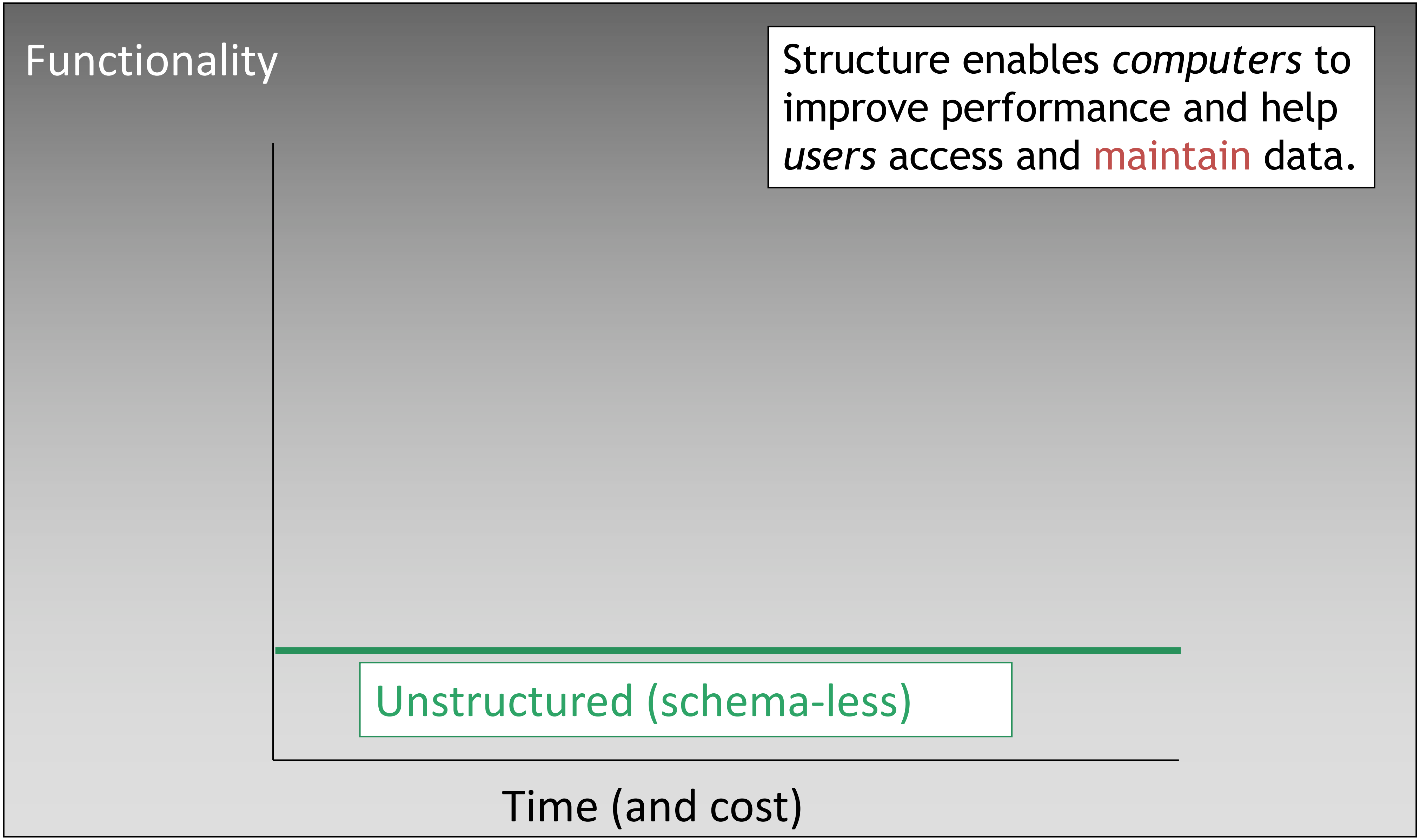
Adapted from Mike Carey, UCI

Reducing Friction – Schema



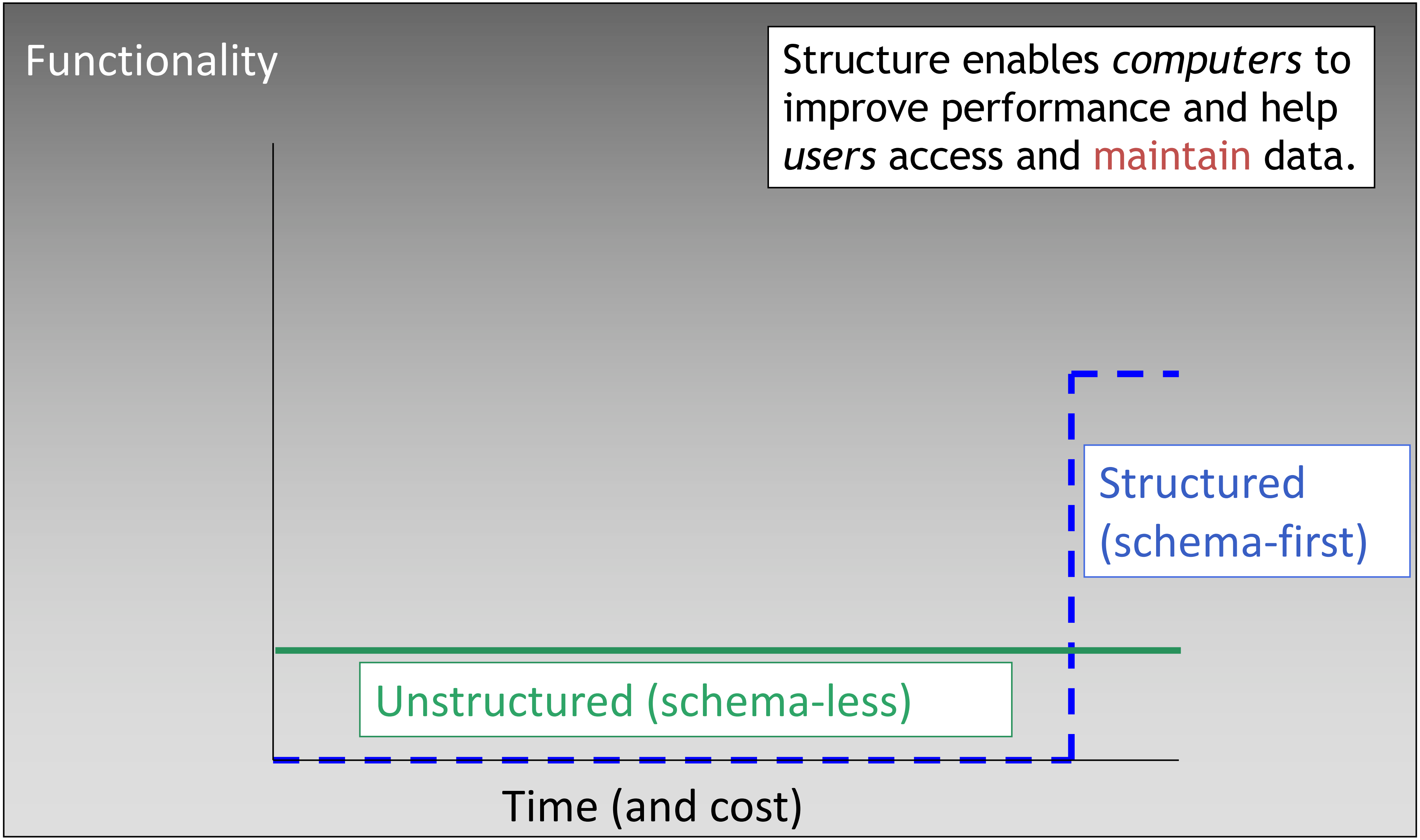
Traditional approach – Schema First
Alternative – "Schema on Read", aka Data Lake or Dataspace
Data Integration remains a "Wicked Problem"

Data Lakes (a.k.a. "Dataspaces")



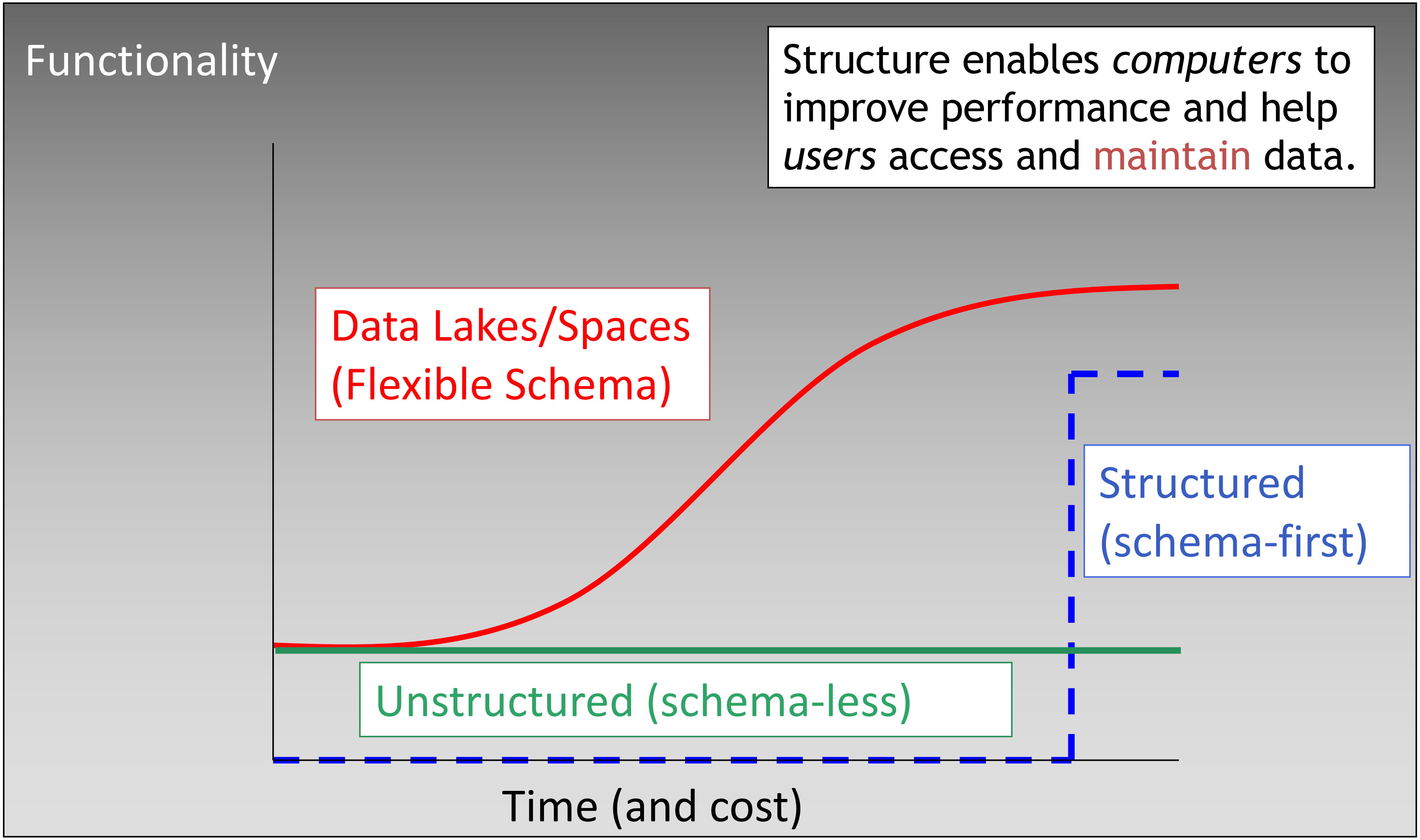
Franklin, Halevy, Maier, "From Databases to Dataspaces: A New Paradigm for Information Management", SIGMOD Record 2005.

Data Lakes (a.k.a. "Dataspaces")



Franklin, Halevy, Maier, "From Databases to Dataspaces: A New Paradigm for Information Management", SIGMOD Record 2005.

Data Lakes (a.k.a. "Dataspaces")



Franklin, Halevy, Maier, "From Databases to Dataspaces: A New Paradigm for Information Management", SIGMOD Record 2005.

Machine Learning Pipelines

- Data Analytics is a complex process
- Rare to simply run a single algorithm on an existing data set
- Model training is only part of the process
- Emerging systems support more complex workflows:
  - Spark MLPipelines
  - Google TensorFlow
  - KeystoneML and Clipper Model Serving (BDAS)

KeystoneML

Declarative API ➔ Automaton & Optimization

E. Sparks et al., "MLI: An API for Distributed Machine Learning", *ICDM* 2013

E. Sparks et al., "Automating Model Search for Large-Scale Machine Learning", *SOCC 2015*

S. Venkataraman et al., "Ernest: Efficient Performance Prediction for Large-Scale Analytics", *NSDI 2016*

E. Sparks et al., "KeystoneML: Optimizing Pipelines for Large-Scale Advanced Analytics", *ICDE 2017*

KeystoneML

Declarative API ➔ Automation & Optimization

ML operator selection/
hyperparameter tuning

E. Sparks et al., “MLI: An API for Distributed Machine Learning”, *ICDM* 2013

E. Sparks et al., “Automating Model Search for Large-Scale Machine Learning”, *SOCC 2015*

S. Venkataraman et al., ”Ernest: Efficient Performance Prediction for Large-Scale Analytics”, *NSDI 2016*

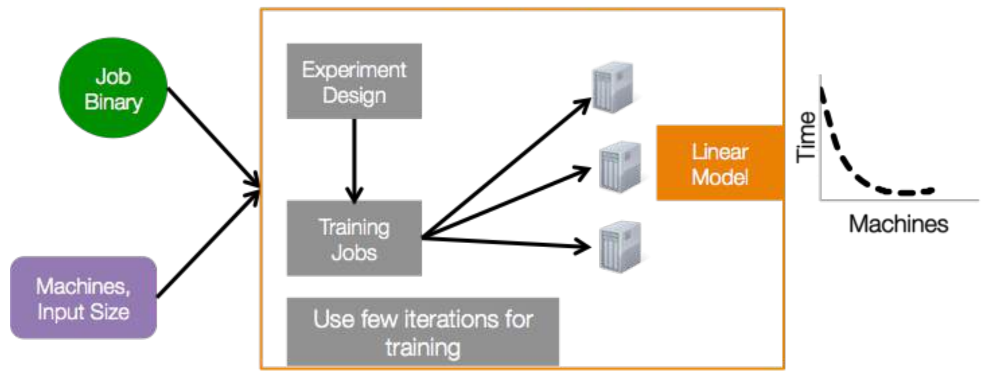
E. Sparks et al., “KeystoneML: Optimizing Pipelines for Large-Scale Advanced Analytics”, *ICDE 2017*

KeystoneML

Declarative API ➔ Automation & Optimization

ML operator selection/ hyperparameter tuning

Auto-provisioning cloud resources



E. Sparks et al., "MLI: An API for Distributed Machine Learning", *ICDM* 2013

E. Sparks et al., "Automating Model Search for Large-Scale Machine Learning", *SOCC 2015*

S. Venkataraman et al., "Ernest: Efficient Performance Prediction for Large-Scale Analytics", *NSDI 2016*

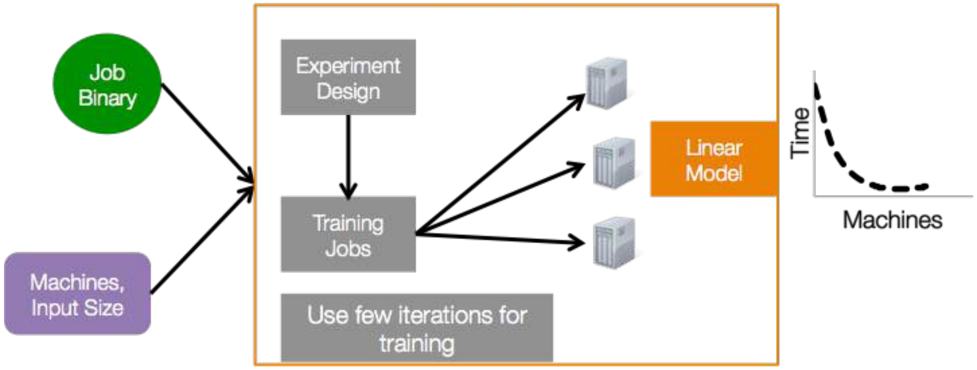
E. Sparks et al., "KeystoneML: Optimizing Pipelines for Large-Scale Advanced Analytics", *ICDE 2017*

KeystoneML

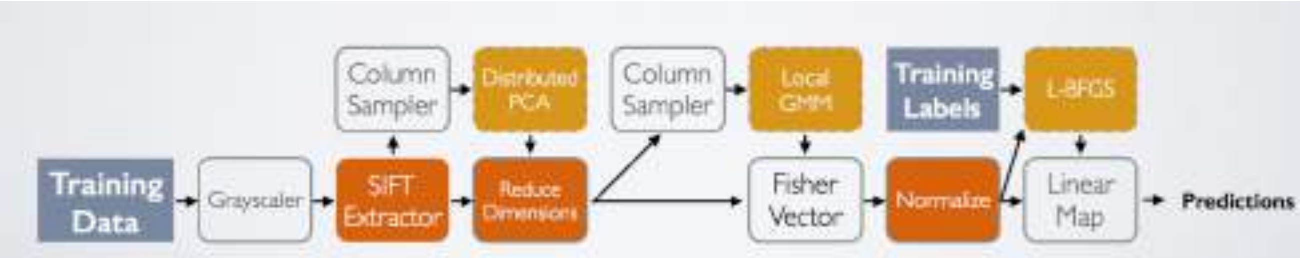
Declarative API ➔ Automation & Optimization

ML operator selection/ hyperparameter tuning

Auto-provisioning cloud resources



Pipeline optimization



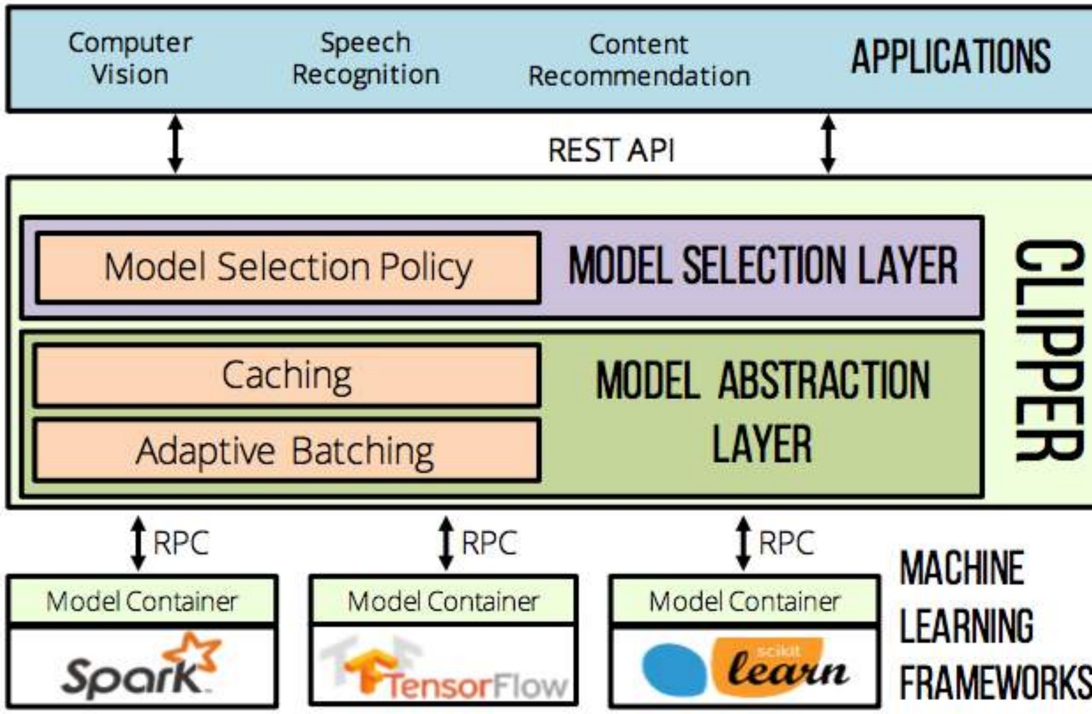
E. Sparks et al., "MLI: An API for Distributed Machine Learning", *ICDM* 2013

E. Sparks et al., "Automating Model Search for Large-Scale Machine Learning", *SOCC 2015*

S. Venkataraman et al., "Ernest: Efficient Performance Prediction for Large-Scale Analytics", *NSDI 2016*

E. Sparks et al., "KeystoneML: Optimizing Pipelines for Large-Scale Advanced Analytics", *ICDE 2017*

Deployment: Model Serving



Clipper: A prediction serving system that spans multiple ML frameworks

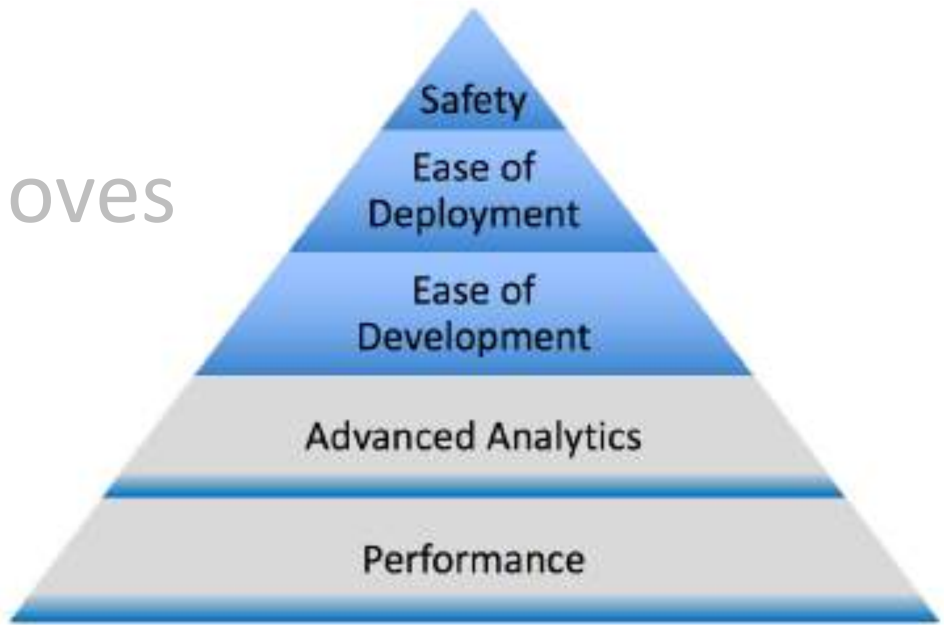
- Simplifies model serving
- Bounds latency and increases prediction throughput
- Enables real-time learning and personalization across machine learning frameworks
- Can be extended to support edge processing

D. Crankshaw et al., "Clipper: A Low-Latency Online Prediction Serving System", *NSDI Conf.,* March 2017

https://github.com/ucbrise/clipper

as we Move Up the Hierarchy…

a new set of concerns moves to the fore:



1) Reducing Friction:

   Ease of Development and Deployment

2) Data Science/Analytics Full Lifecycle Concerns

3) "Safe" Data Science and Human Factors

The Data Science Lifecycle

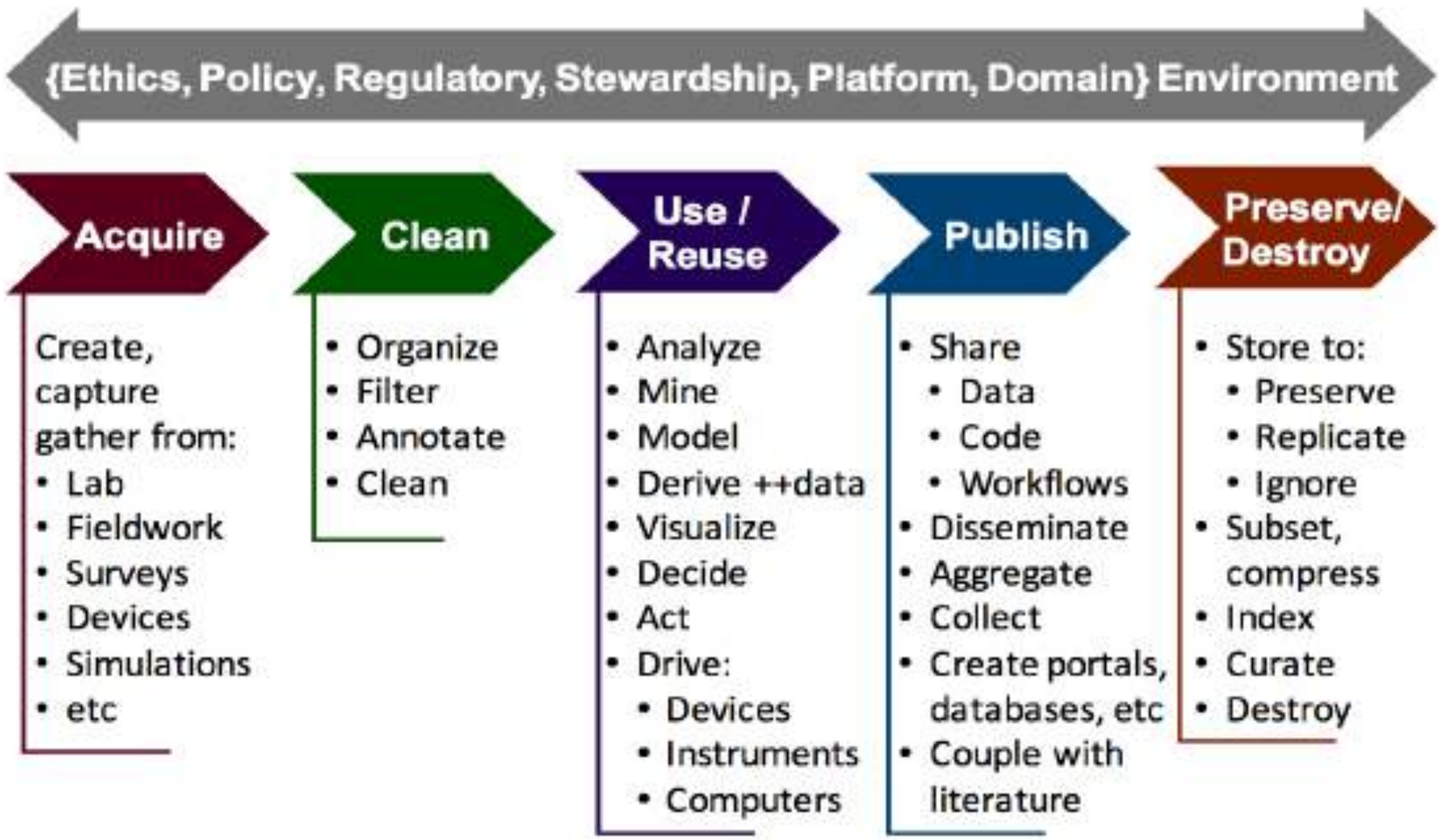
CISE AC Data Science Report

December 2016

REALIZING THE POTENTIAL OF DATA SCIENCE

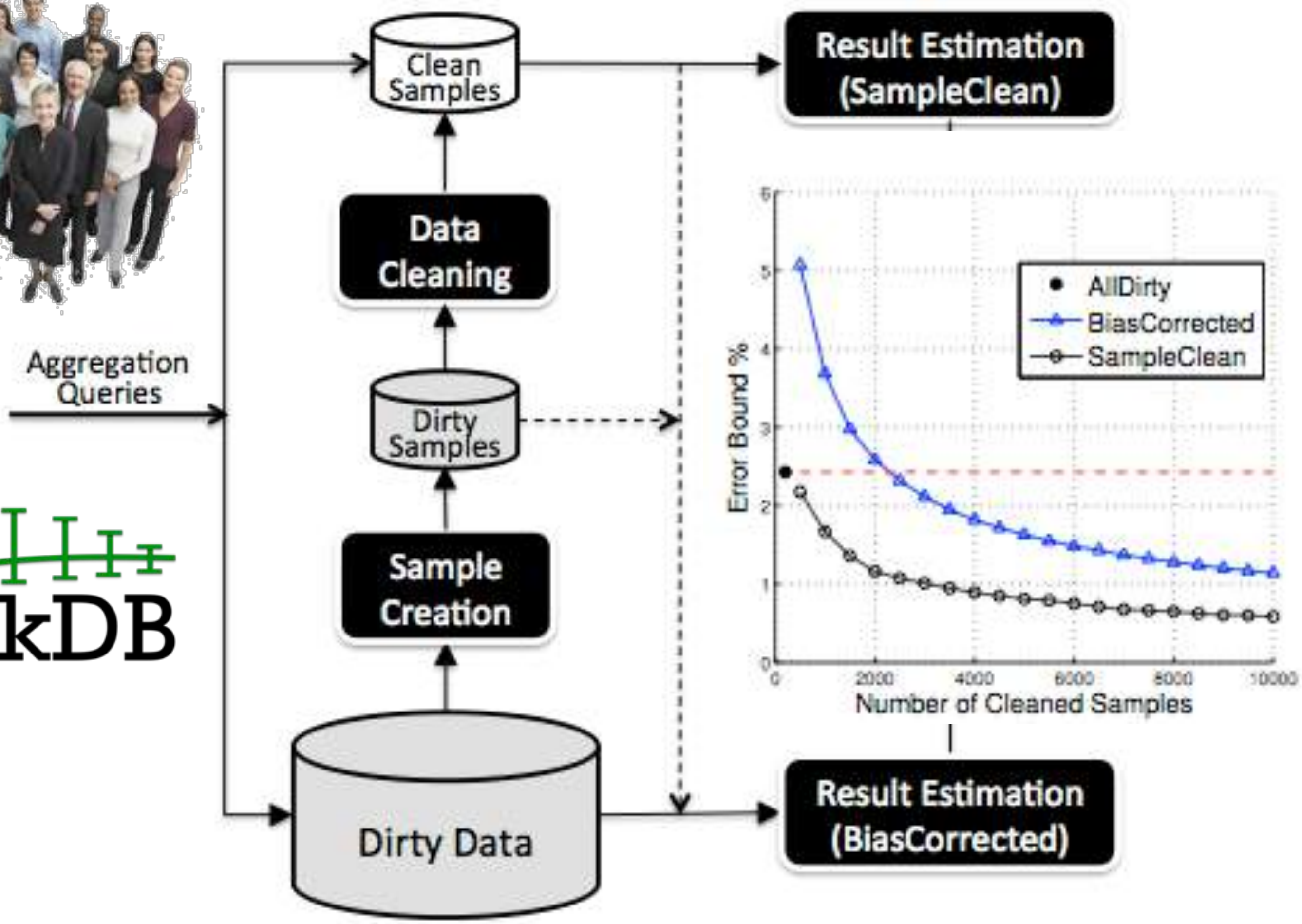
Final Report from the National Science Foundation Computer and Information Science and Engineering Advisory Committee Data Science Working Group

Francine Berman and Rob Rutenbar, co-Chairs
Henrik Christensen, Susan Davidson, Deborah Estrin, Michael Franklin, Brent Hailpern, Margaret Martonosi, Padma Raghavan, Victoria Stodden, Alex Szalay



Data Cleaning: SampleClean

Key Systems Issues – how to deal with latency and cost of the crowd?



J. Wang, S. Krishnan, *et al.*, A Sample-and-Clean Framework for Fast and Accurate Query Processing on Dirty Data, *SIGMOD 2014*

Curation and Reproducibility

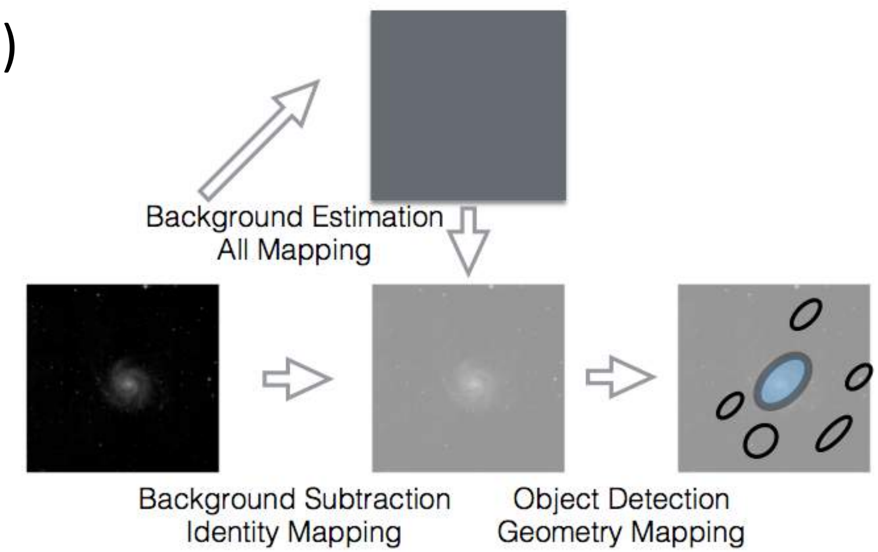
Data outlives any particular application:

“[database systems] let you use one set of data in multiple ways, including ways that are unforeseen at the time the database is built and the 1st applications are written.” (Curt Monash, analyst/blogger)

Curation and Reproducibility

Data outlives any particular application:

"[database systems] let you use one set of data in multiple ways, including ways that are unforeseen at the time the database is built and the 1st applications are written." (Curt Monash, analyst/blogger)



Z. Zhang et al., Hippo, HPDC 17:

- Efficient fine-grained lineage for machine learning and advanced analytics pipelines
- Supports code debugging, result analysis, data anomaly removal and computation replay
- Provides interactive answers to queries over lineage

Bias, Privacy and Ethical Issues


SALON VIDEO NEWS POLITICS ENTERTAINMENT LIFE INNOVATION

SUNDAY, MAY 14, 2017 09:00 AM HKT

Why big-data analysis of police activity is inherently biased

One predictive policing algorithm targeted black neighborhoods at roughly twice the rate of white neighborhoods


sign in become a supporter subscribe search jobs dating more International edition

theguardian

UK world sport football opinion culture business lifestyle fashion environment tech travel all sections

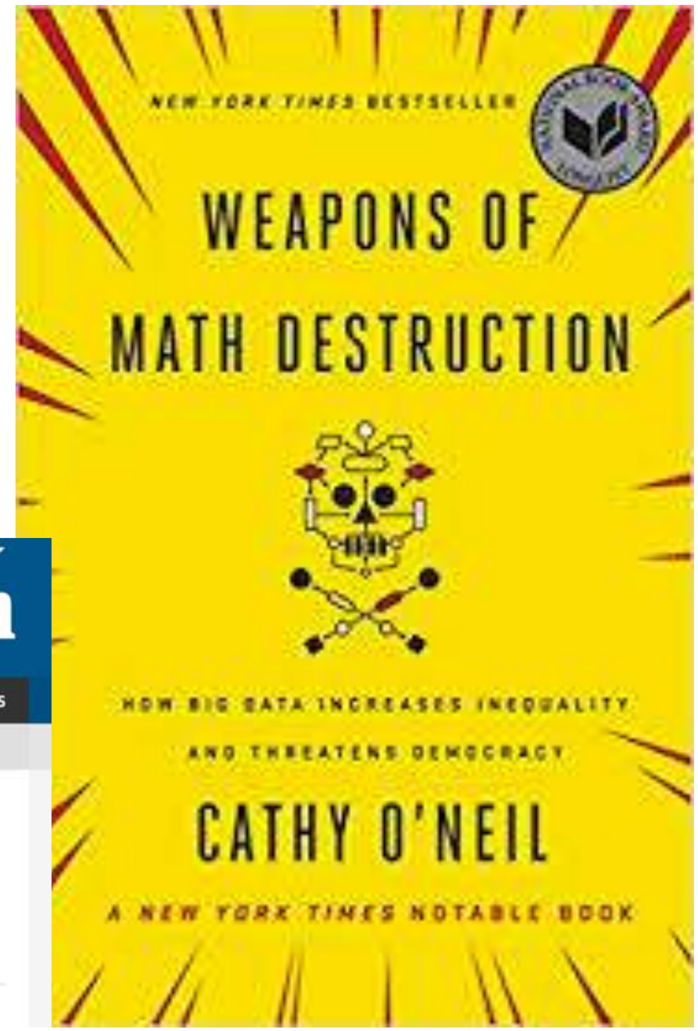
home › tech

Big data
Big data development (series)

Little privacy in the age of big data

With massive amounts of our personal data now being routinely collected and stored, privacy breaches are almost inevitable

Most popular



"With Big Data comes Big Responsibility"

Humans in the loop

Data Consumers

Predictions
Decisions

Data Generators

Data

Citizen Science

Data Scientists

Commands

Data Processors

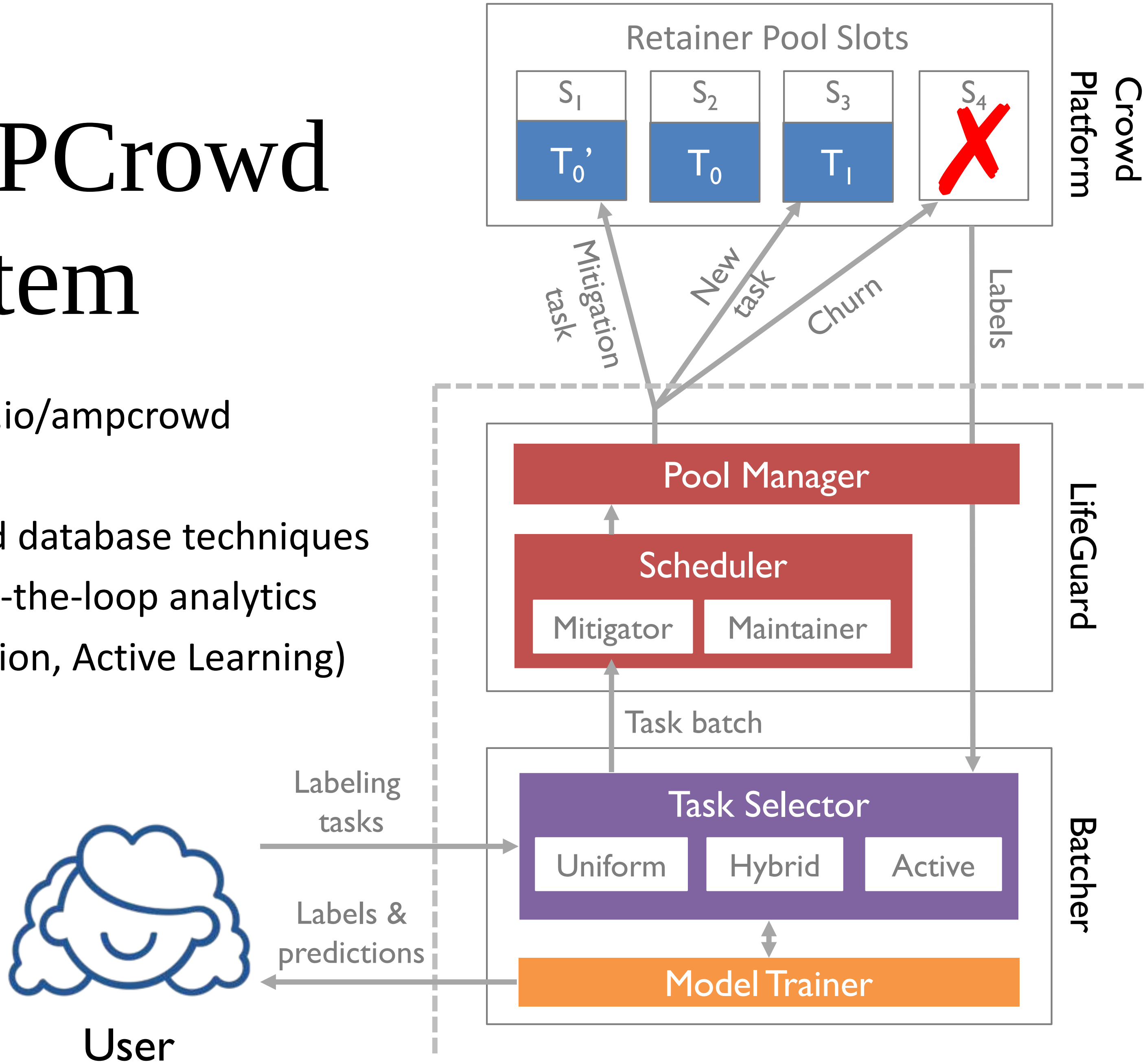
Computation

People Icons created by Clara Joy from Noun Project

The AMPCrowd System

amplab.github.io/ampcrowd

Leveraging systems and database techniques for hybrid human-in-the-loop analytics (e.g. Straggler Mitigation, Active Learning)



D. Haas, *et al.*, Clamshell: Scaling Up Crowds for Low Latency Data Labeling, *PVLDB 9(4)*
Haas & Franklin, Cioppino: Multi-tenant Crowdsourcing, *HCOMP 2017*

New Challenges Summary

Performance, Scalability, and Functionality remain important, but we face new challenges, <u>including</u>:

Ease of Development and Deployment

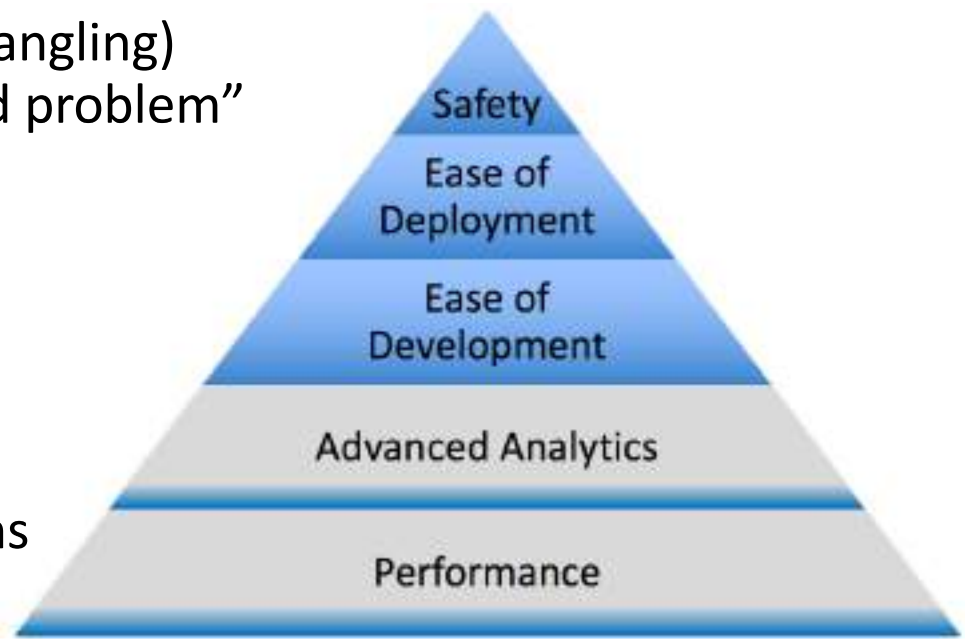
- Leverage database-style abstractions (e.g., declarative query optimization)
- Make ML and AI pipelines easier to build
- New components for “model serving” and “model management”

Data Science Lifecycle

- Data Acquisition, Cleaning (i.e., wrangling)
- Data Integration remains a “wicked problem”
- Communicating results, Curation,
- “Translational Data Science”

“Safe” Data Science

- end-to-end Bias Mitigation
- Security, Ethics and Data Privacy
- Explaining and influencing decisions
- Human-in-the-loop



Acknowledgements

The work described here is due to an amazing group of AMPLab students, staff, faculty and sponsors and to the open source community.

Thanks and for More Info

Mike Franklin

mjfranklin@uchicago.edu