Big Data Engineering

Introduction

Adam Hill

April 2023

Contents

- Definitions
- Origins of Big Data
- Case Studies and Motivations

Big Data definition

- Data of a very large size, typically to the extent that its manipulation and management present significant logistical challenges
  - Oxford English Dictionary

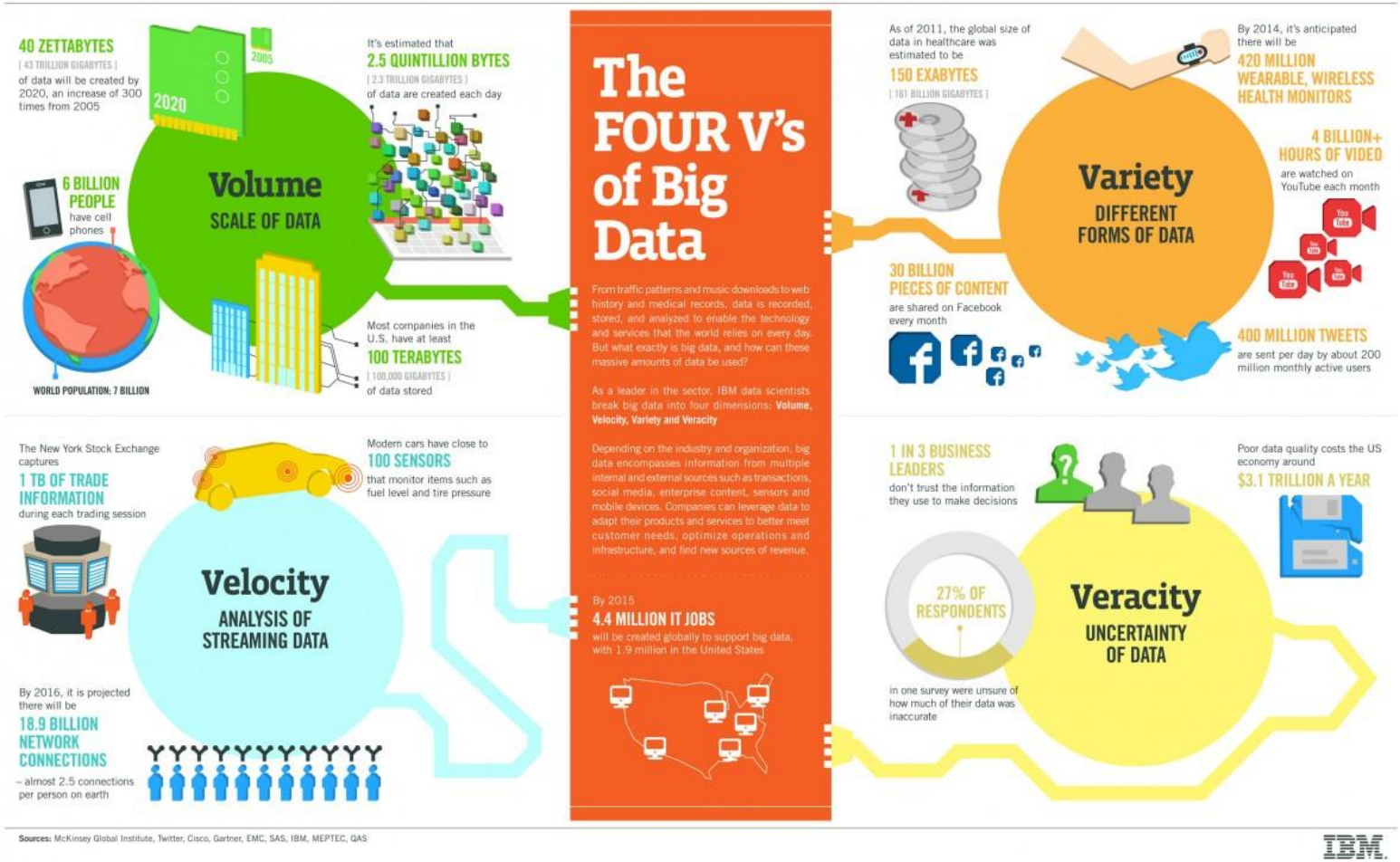
So what is Big Data?

Big data is **high-volume**, **high-velocity** and **high-variety** information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making (source: Gartner glossary)

The three Vs

- Velocity
  - Need to be able to process data faster
  - Handle very large numbers of data elements/sec incoming
- Variety
  - Not just the same old columns
  - New formats, new sources, new details
- Volume
  - Massive volumes are becoming normal
  - Collecting the next level of data
    - E.g. Bank Trades, Website interactions, shopping experiences, etc

The four Vs (IBM)



My Big Data definition

- Any data storage and analysis that:
  - Cannot be processed on a single machine in a timely manner
  - Over time needs more computation and resources than a fixed size system can provide

Origins of Big Data - 1997

Application-Controlled Demand Paging for Out-of-Core Visualization

Michael Cox
MRJ/NASA Ames Research Center
Microcomputer Research Labs, Intel Corporation
<mbc@nas.nasa.gov>

David Ellsworth
MRJ/NASA Ames Research Center
<ellswort@nas.nasa.gov>

Abstract

In the area of scientific visualization, input data sets are often very large. In visualization of Computational Fluid Dynamics (CFD) in particular, input data sets today can surpass 100 Gbytes, and are expected to scale with the ability of supercomputers to generate them. Some visualization tools already partition large data sets into segments, and load appropriate segments as they are needed. However, this does not remove the problem for two reasons: 1) there are data sets for which even the individual segments are too large for the largest graphics workstations, 2) many practitioners do not have access to workstations with the memory capacity required to load even a segment, especially since the state-of-the-art visualization tools tend to be developed by researchers with much more powerful machines. When the size of the data that must be accessed is larger than the size of memory, some form of virtual memory is simply required. This may be by segmentation, paging, or by paged segments. In this paper we

1 Introduction

Visualization provides an interesting challenge for computer systems: data sets are generally quite large, taxing the capacities of main memory, local disk, and even remote disk. We call this the problem of *big data*. When data sets do not fit in main memory (*in core*), or when they do not fit even on local disk, the most common solution is to acquire more resources. This *write-a-check* algorithm has two drawbacks. First, if visualization algorithms and tools are worth developing, then they are worth deploying to more production-oriented scientists and engineers who may have on their desks machines with significantly less memory and disk. Some researchers have noted that their software tools were not used in practice for several years after development because the tools required more power and memory than were available on the average engineer's desk [15]. Second, there may not even be a machine that supports sufficiently large main memory or local disk for the data set one wishes to visualize. We find this in particular in the area of visualization of *Computational Fluid Dynamics (CFD)*

Map Reduce 2008

MapReduce: Simplified Data Processing on Large Clusters

Jeffrey Dean and Sanjay Ghemawat

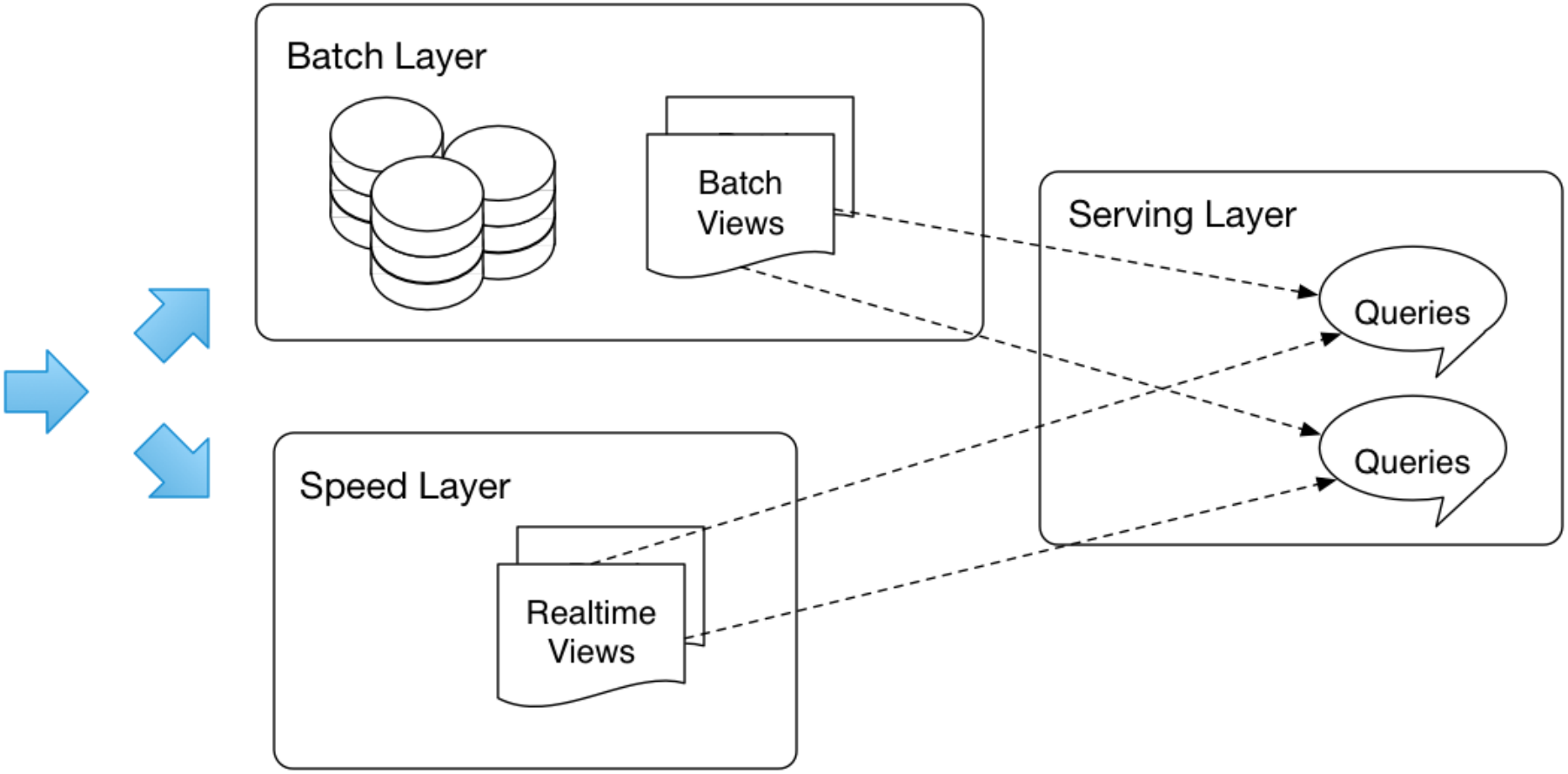
jeff@google.com, sanjay@google.com

Google, Inc.

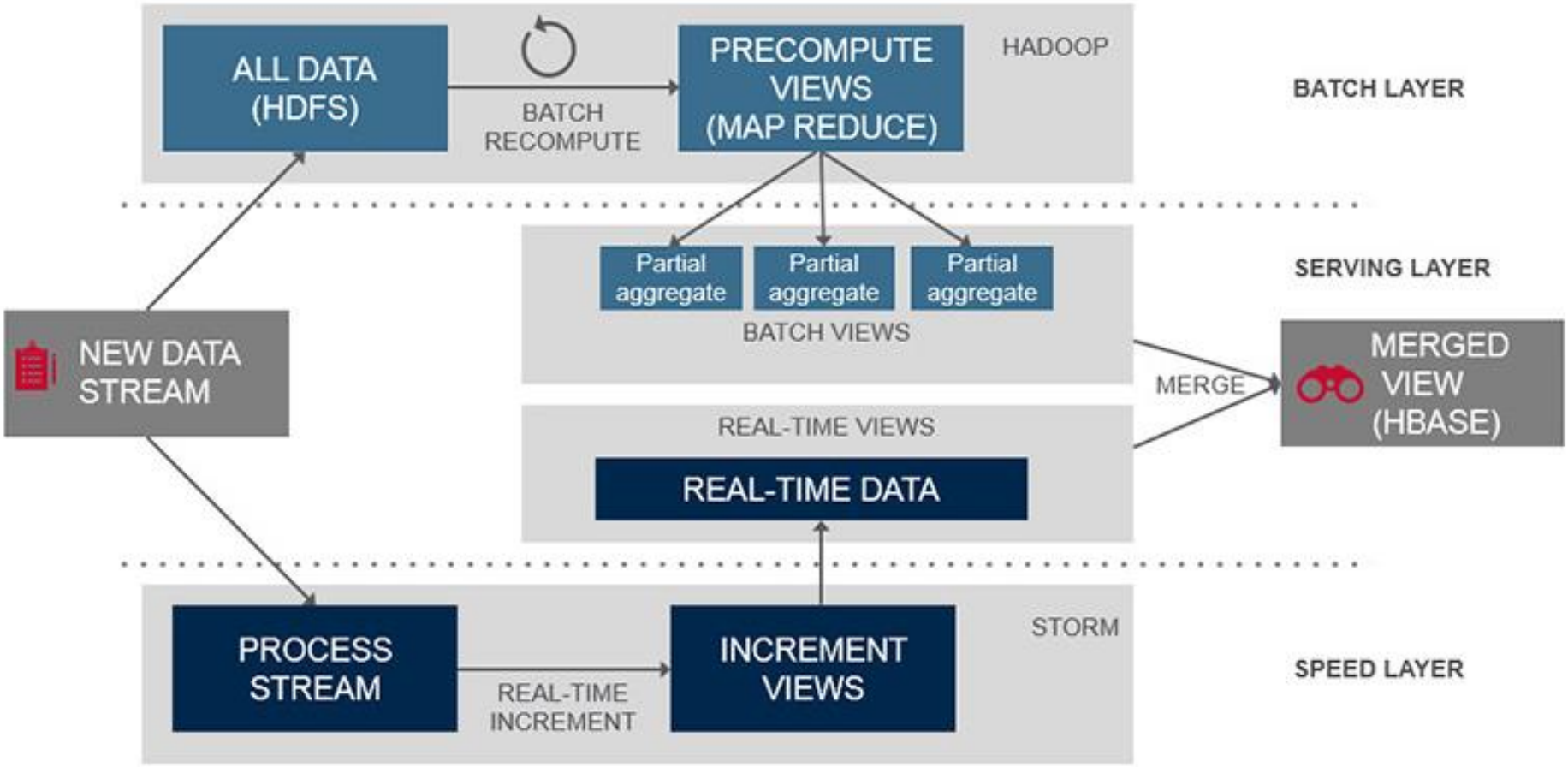
Master Data

- One widely used approach
- You ingest core data and never change it
  - You can create summaries, cleaned data, etc
  - But the original data is immutable
- Cheap disk space…

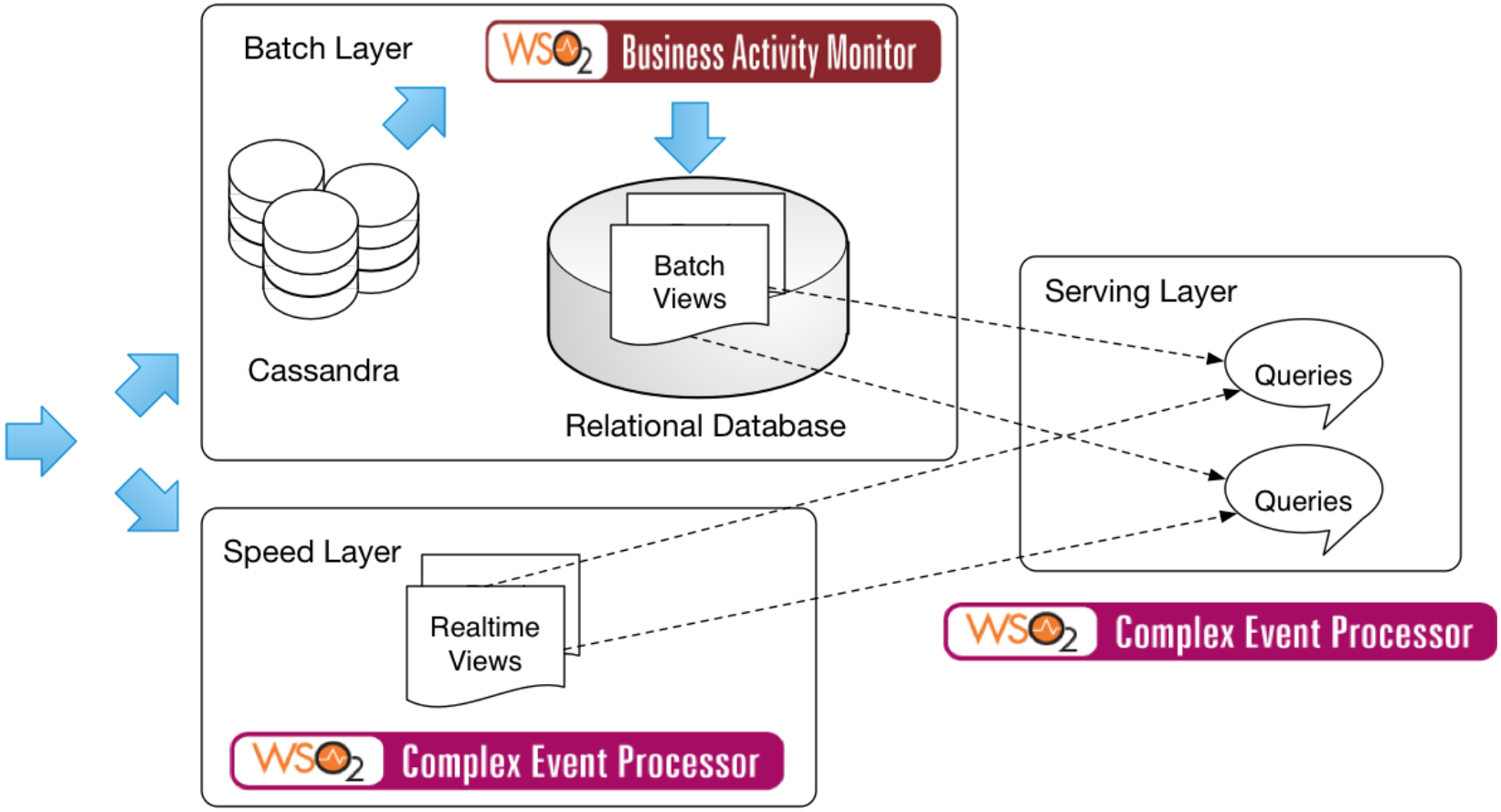
Lambda Architecture



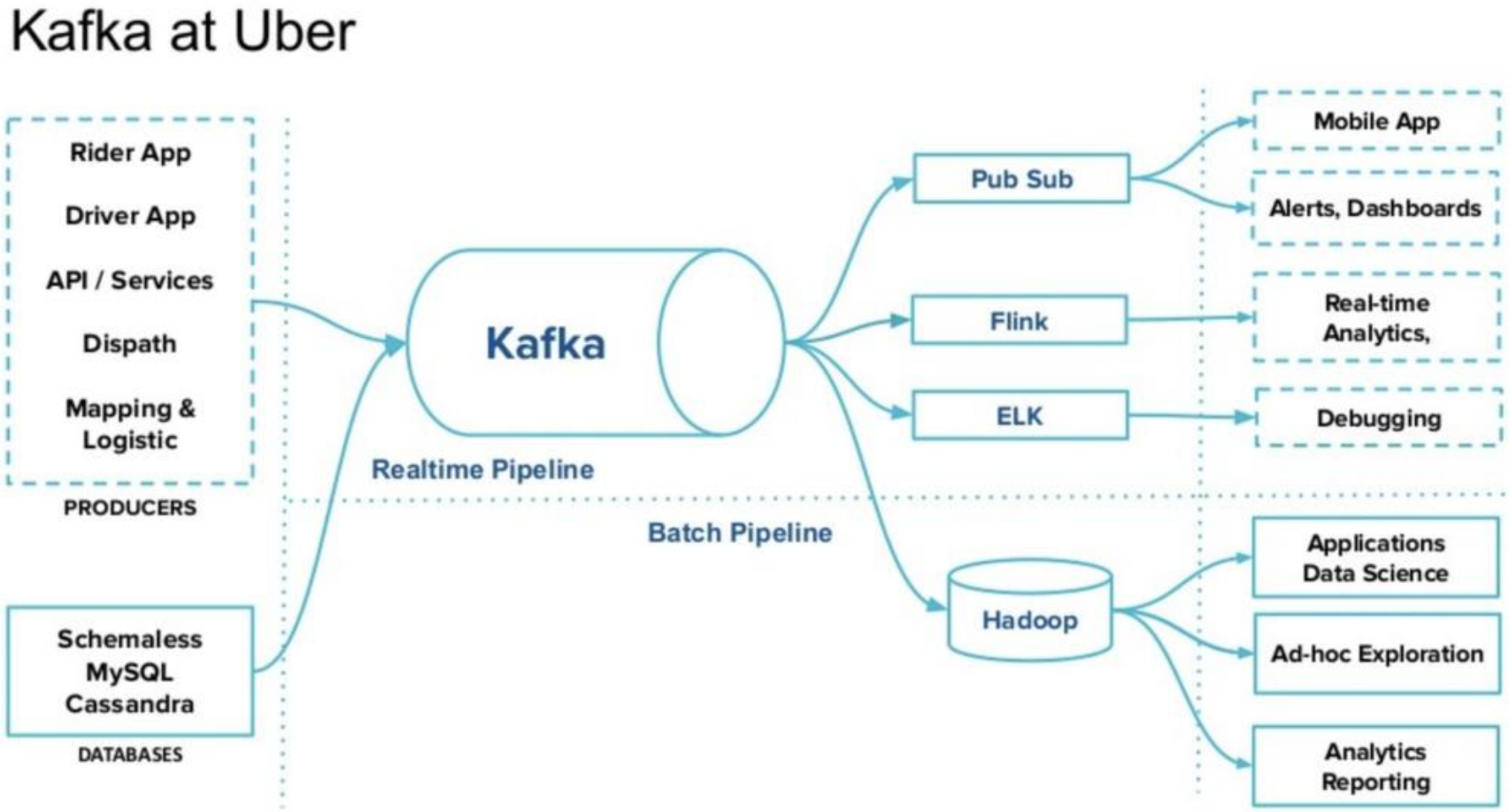
Lambda Architecture (MapR)



Lambda Architecture instantiation (WSO2)



Kappa Architecture



Big Data technologies

- Map Reduce
  - Hadoop, Spark, etc
- In-Memory Directed Acyclic Graphs
  - Spark, Tez
- Realtime Stream processing
  - Spark, Flink, Kafka
- NoSQL
  - Cassandra, Mongo, Neo4j, etc
- Statistical Analysis
  - R, SparkR, Julia
- Machine Learning
  - PyTorch, MLlib, TensorFlow, Scikit-learn

WHY PYTHON?

Python for Big Data

- Python is a great language for Data Science
  - NumPy, Pandas, many graphic packages
- Python is a great language for Spark
  - Lambdas, concise statements, DataFrames
- Ipython/Jupyter is a great notebook

Other options

- C/C++ are fast to run, but generally slow to develop
- Scala is the original language for Spark
  - But not so strong in wider data science
- Java is too wordy for Data Science!
- R is a great model for both Data Science and Spark, if you are a statistician
- Julia has gained a lot of traction but but not widely adopted yet

Notebooks

- Web-based systems that combine documentation, code and graphics into one place
- Two front runners for Big Data
  - Jupyter (formerly IPython)
    - Based on Python but supporting other languages
  - Apache Zeppelin
    - More language neutral but newer and more buggy (this may be changing of course)

Untitled 2

https://notebooks.azureml-int.net/n/sVUasfucctw/notebooks/Untitled%202.ipynb#

jupyter Untitled 2 Last Checkpoint: 22 minutes ago (autosaved)

File Edit View Insert Cell Kernel Help Python 2

Code Cell Toolbar: None

This is a markdown cell used for documenation

The above cell was written as: "### This is a markdown cell", followed by Shift+Enter

In [12]:
```
import math
print "This is a code cell... and Pi is = ", math.pi
```

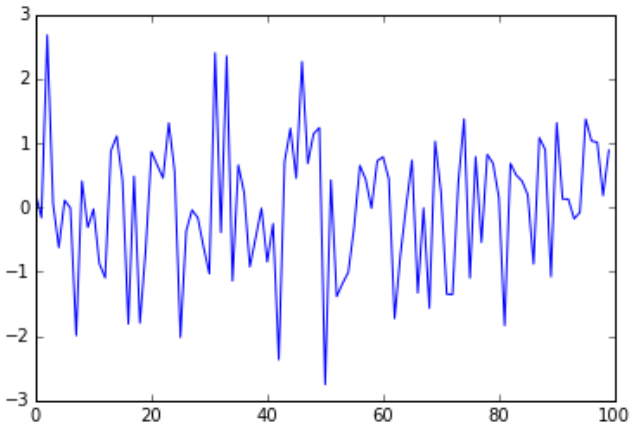
```
This is a code cell... and Pi is =  3.14159265359
```

In [13]:
```
# to enable inline graphs, etc.
%pylab inline

plot(randn(100))
```

```
Populating the interactive namespace from numpy and matplotlib
```

Out[13]:
```
[<matplotlib.lines.Line2D at 0x7fe2a2496cd0>]
```



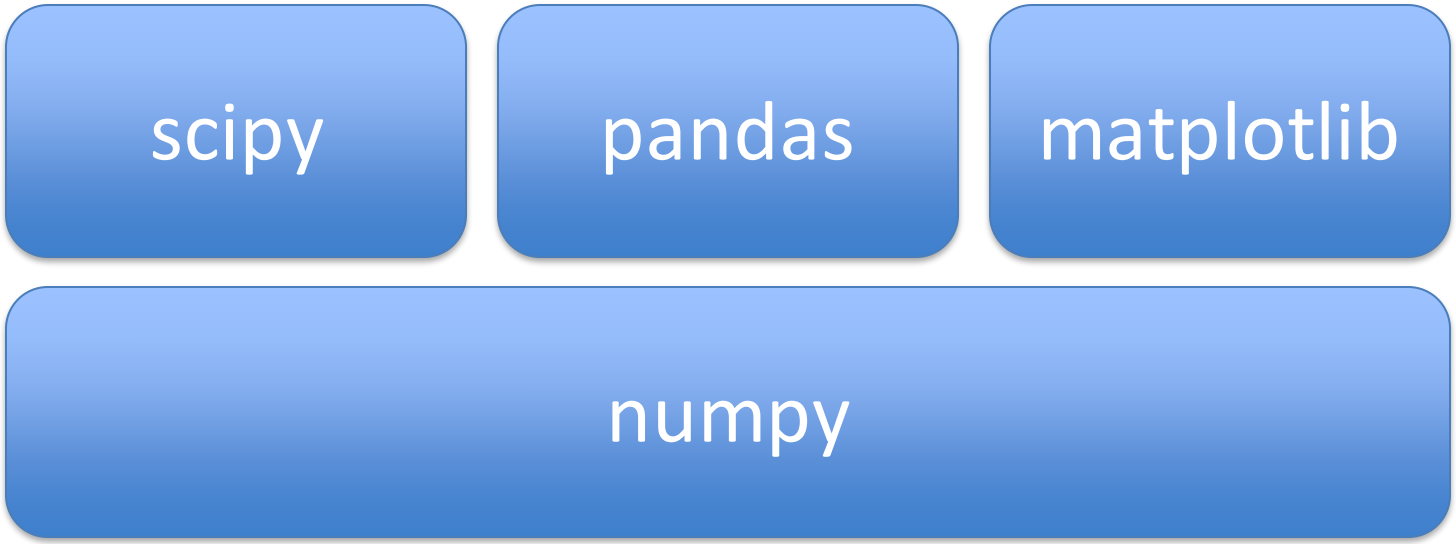
Numpy

```
>conda install numpy
```

***>(sudo) pip install numpy – make sure to manage your env*

- Numerical and scientific analysis library in Python
- Foundation of most data analysis in Python
- Based on arrays of data

Base ecosystem



Pandas

- A rich relational data model built on top of Python's numpy
  - Emerged from the finance industry
  - Like R's data.frame (but maybe better?)

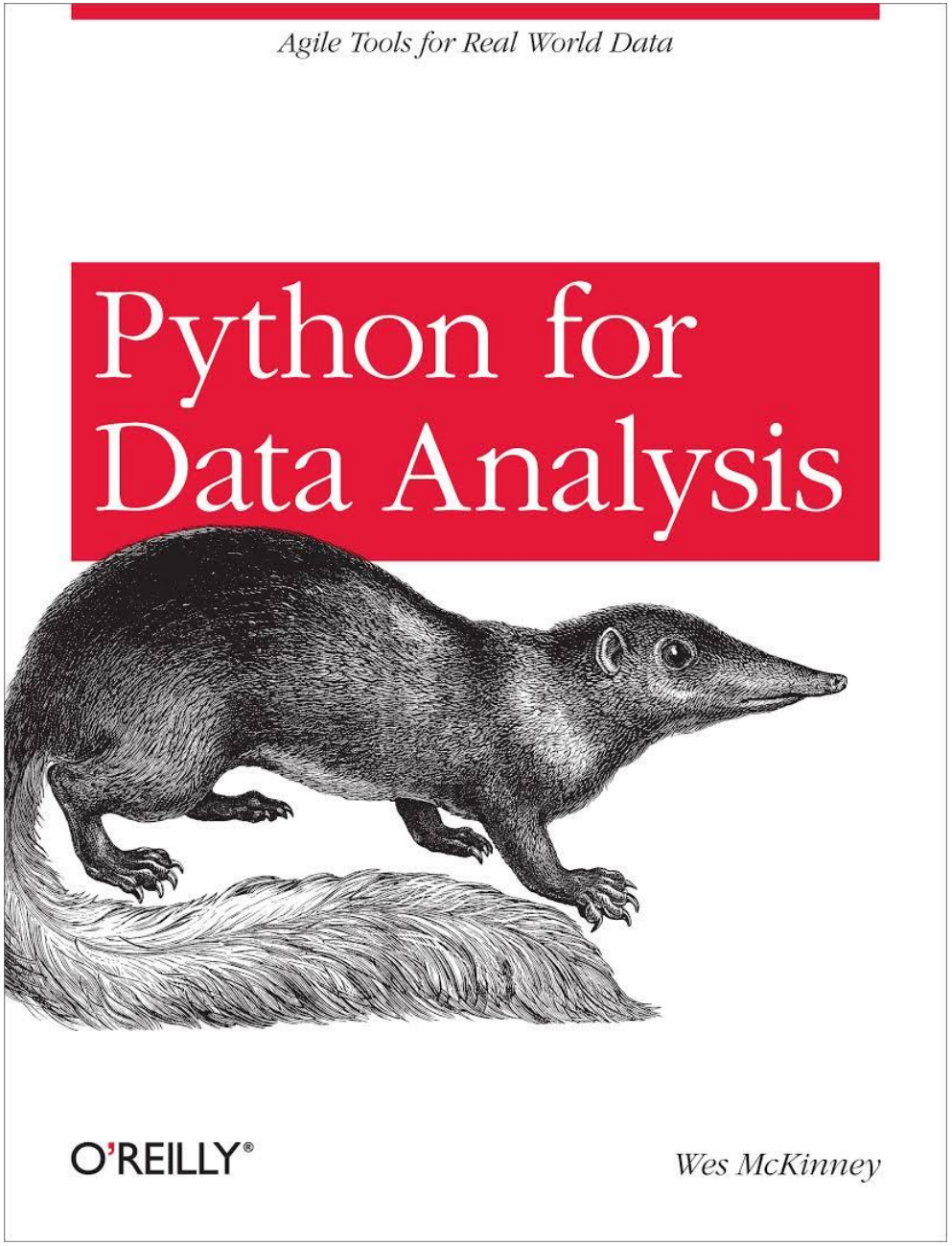
Matplotlib

- A simple graphing library for Python
- Works well with Pandas and Numpy
- Integrated into Jupyter
- There are many alternatives
  - E.g. Bokeh, seaborn, Altair, plotly

This course

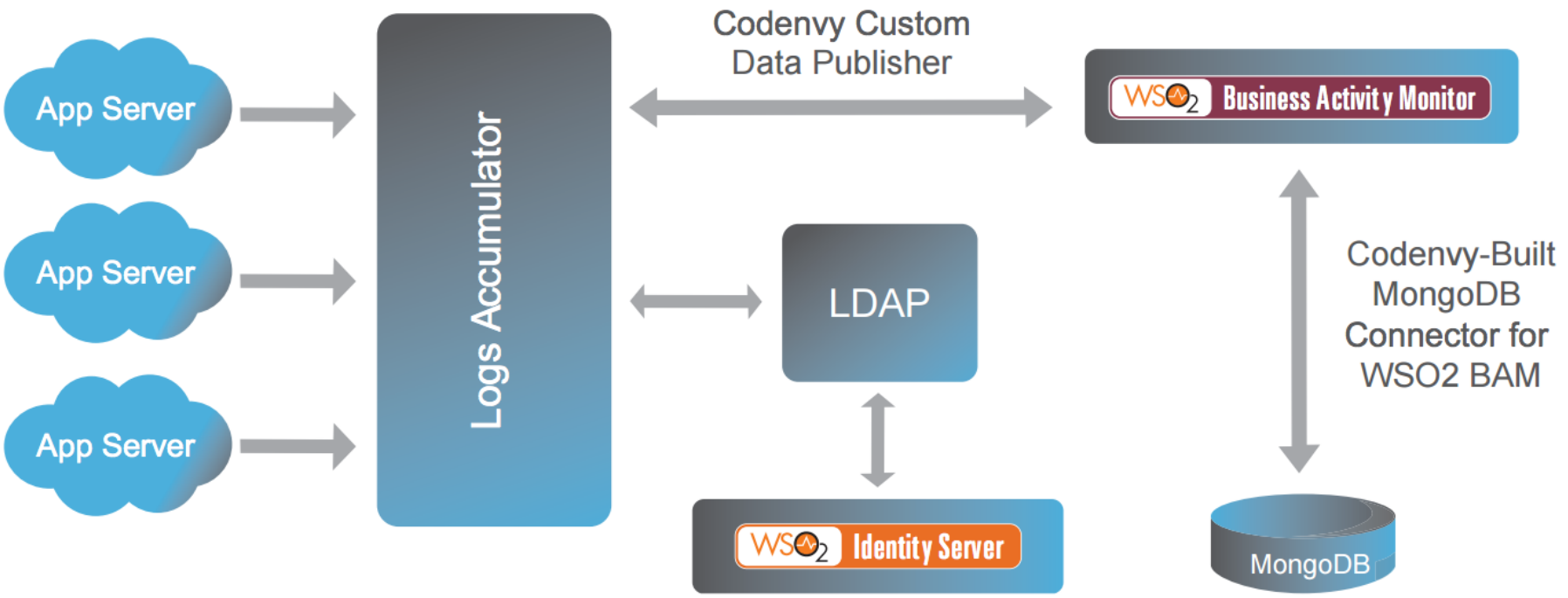
- Python
- pandas
- matplotlib
- pyspark
  - Apache Spark with Python
- Jupyter
- Some other libraries etc as we go

Recommended Reading!



CASE STUDIES

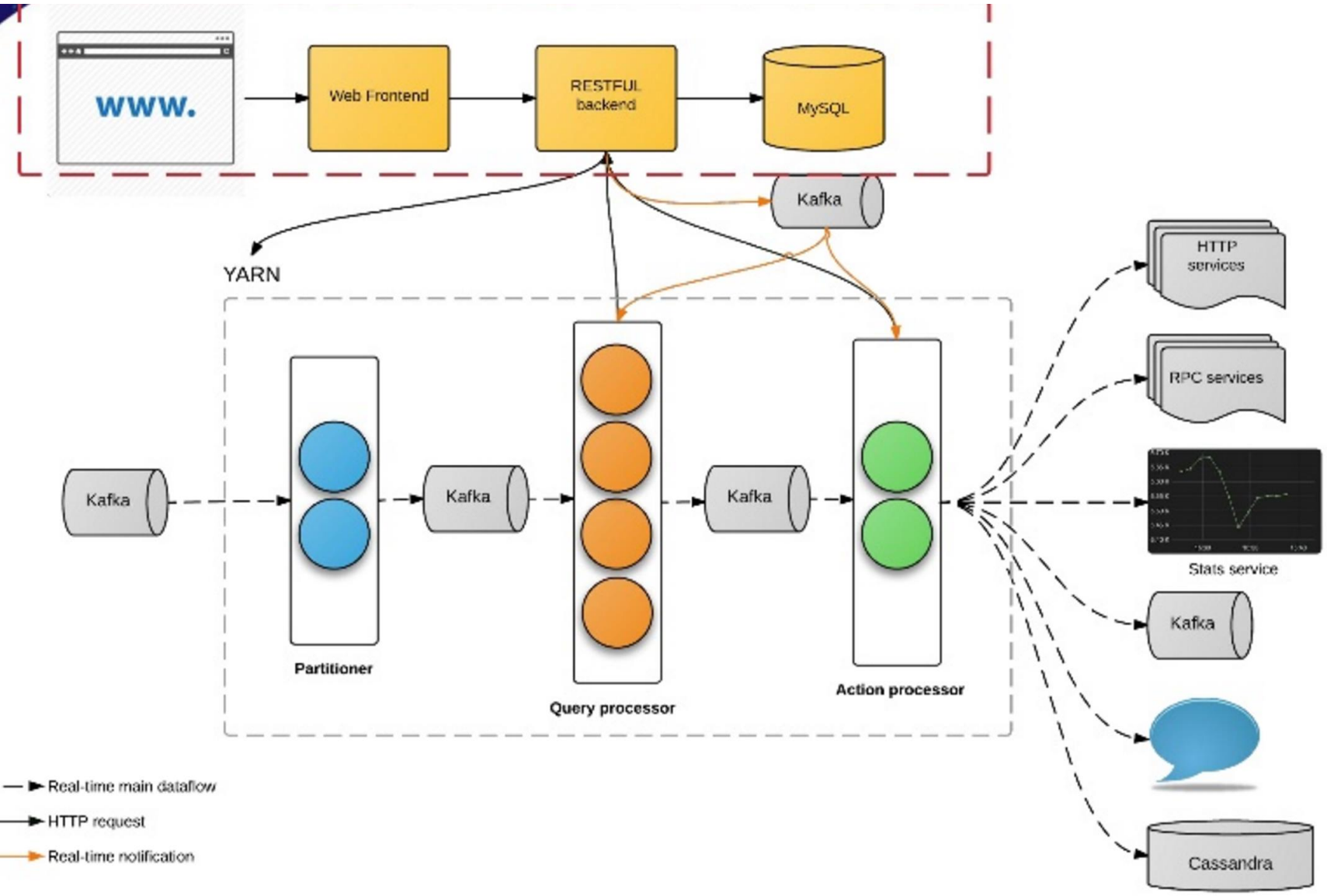
Big Data
Cloud management analytics



Realtime Big Data

- New York-based Bank
- 25 servers in a cluster analysing trading and system data from operational systems
- Siddhi-based engine processing data in realtime
- Handling 10,000s of events/second

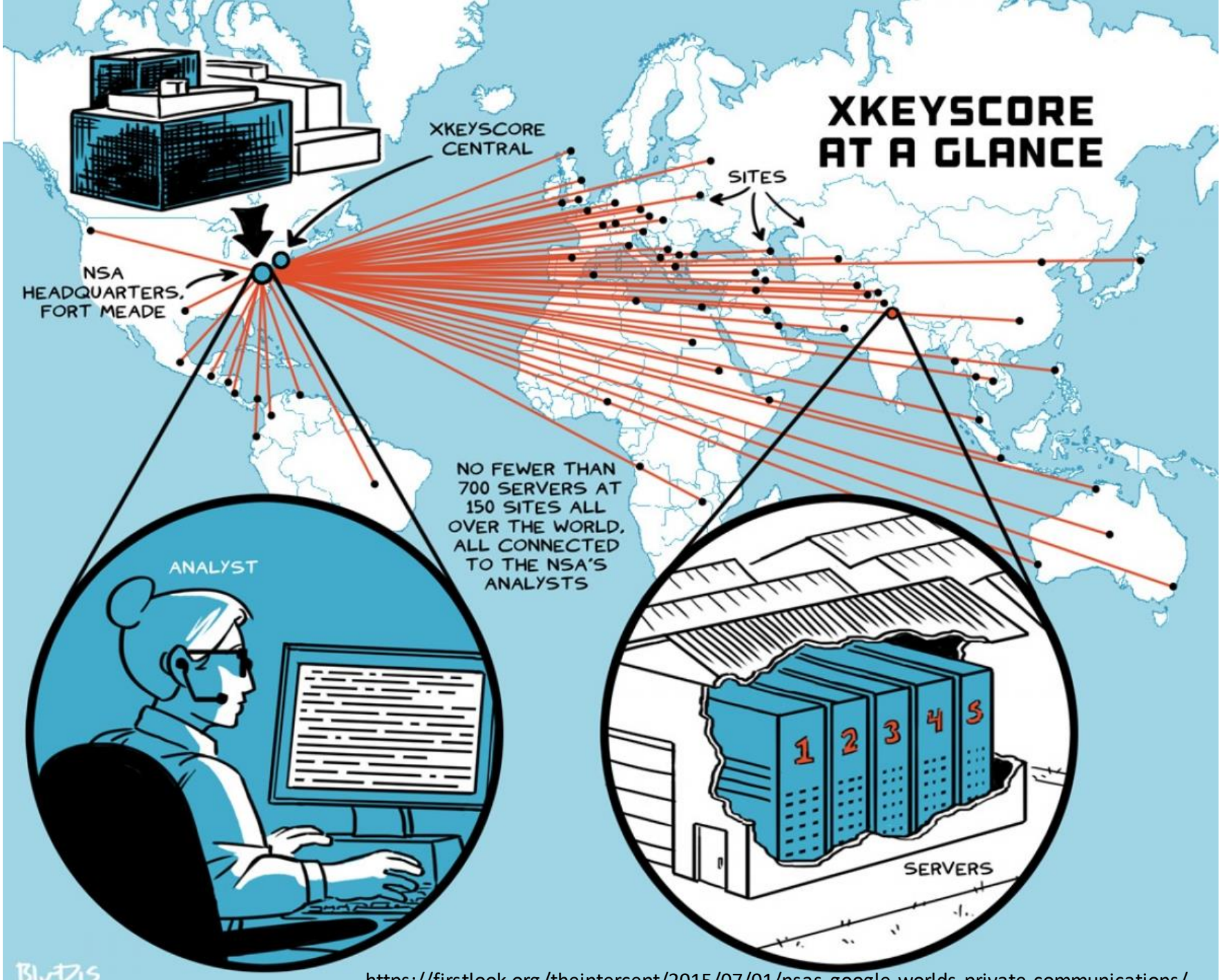
Realtime Big Data at Uber



Realtime Big Data at Uber

- 100+ production apps
- 30 billion messages / day
  - 347,000 messages / second
- Fraud, anomaly detection
- Marketing, promotion
- Monitoring, feedback
- Real time analytics and visualization

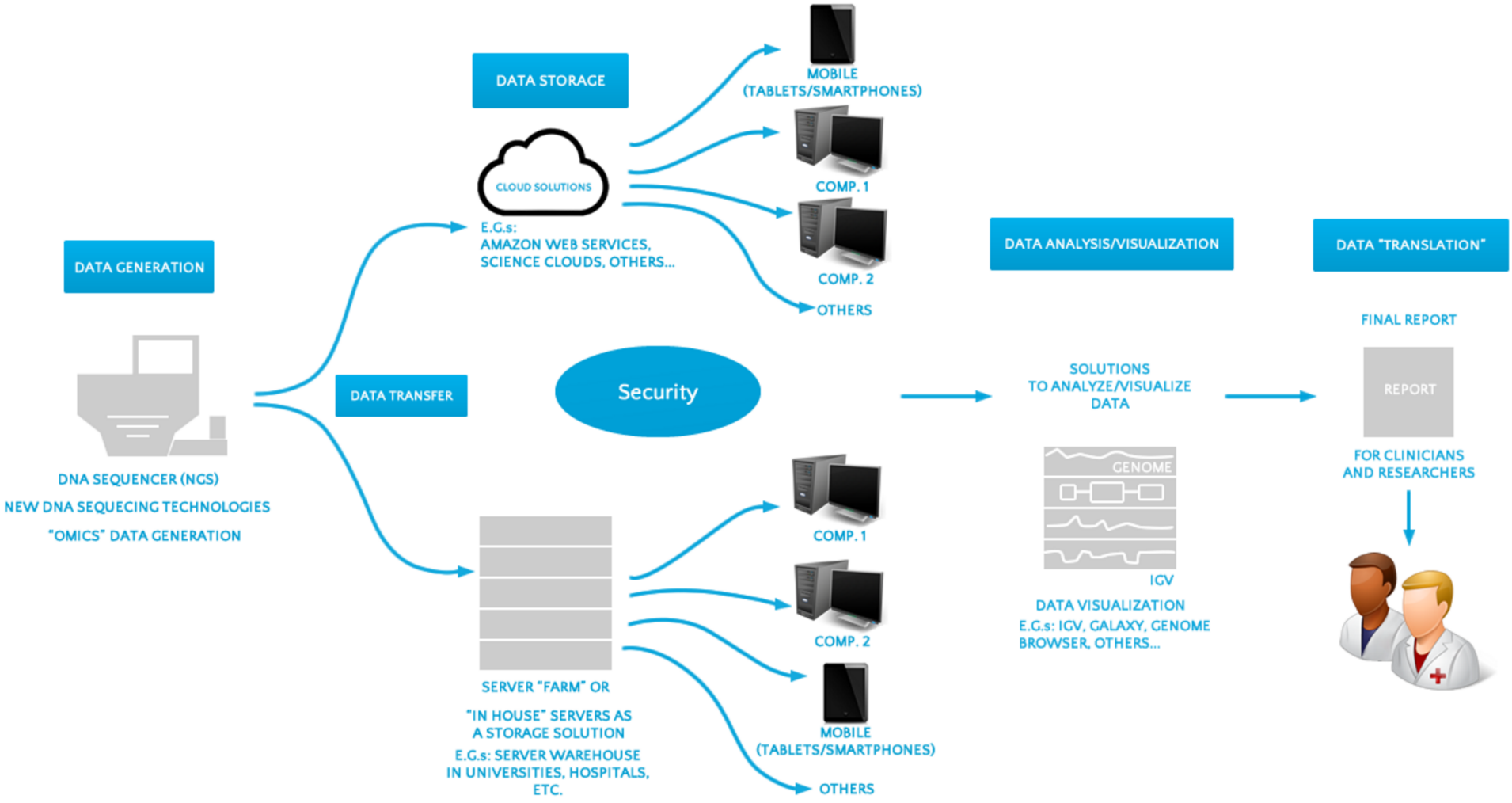
https://freo.me/siddhi-uber



https://firstlook.org/theintercept/2015/07/01/nsas-google-worlds-private-communications/

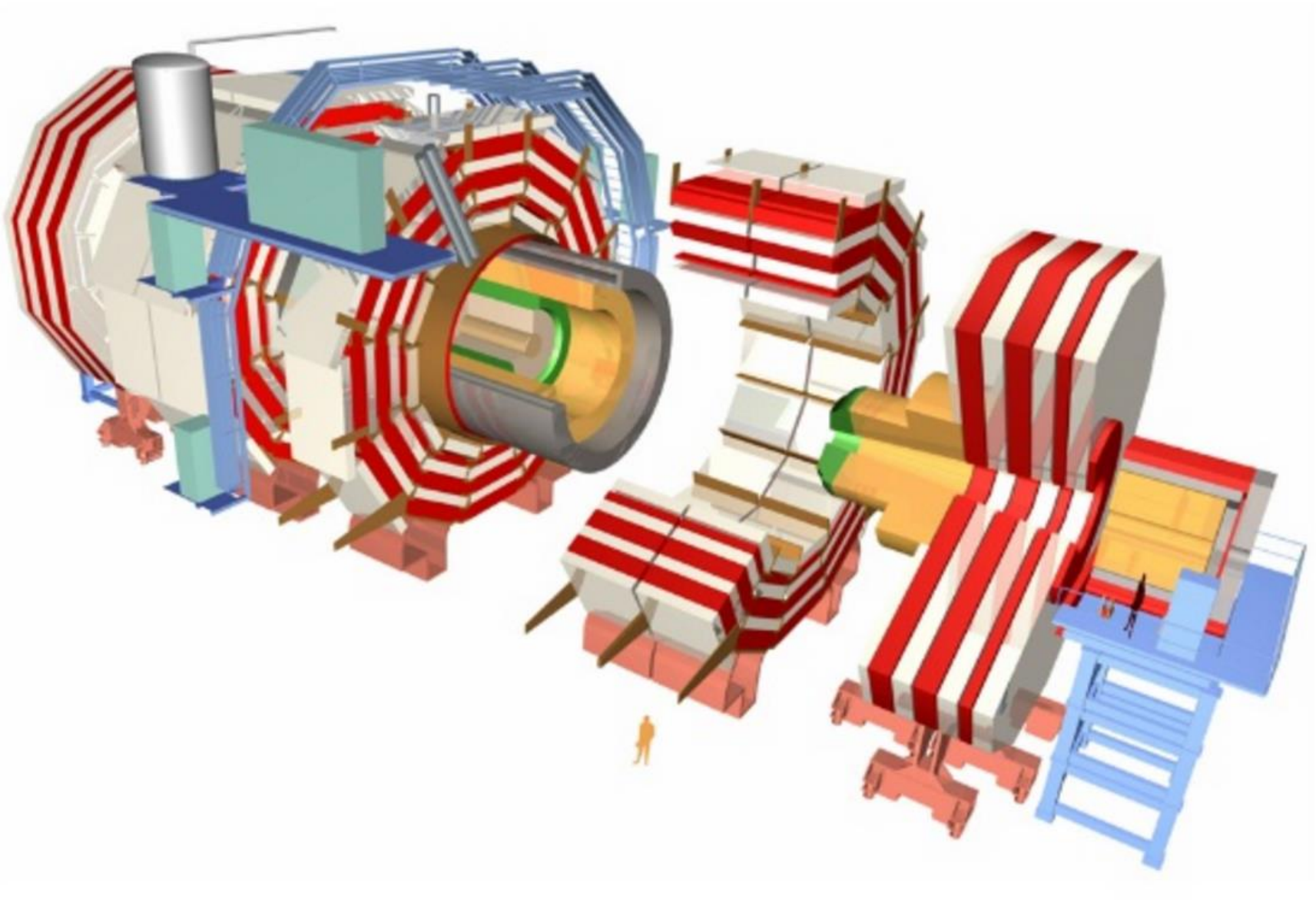
Big Data in Genomics

http://www.laboratory-journal.com/science/information-technology-it/big-data-genomics-challenges-and-solutions



Maclaren Formula 1

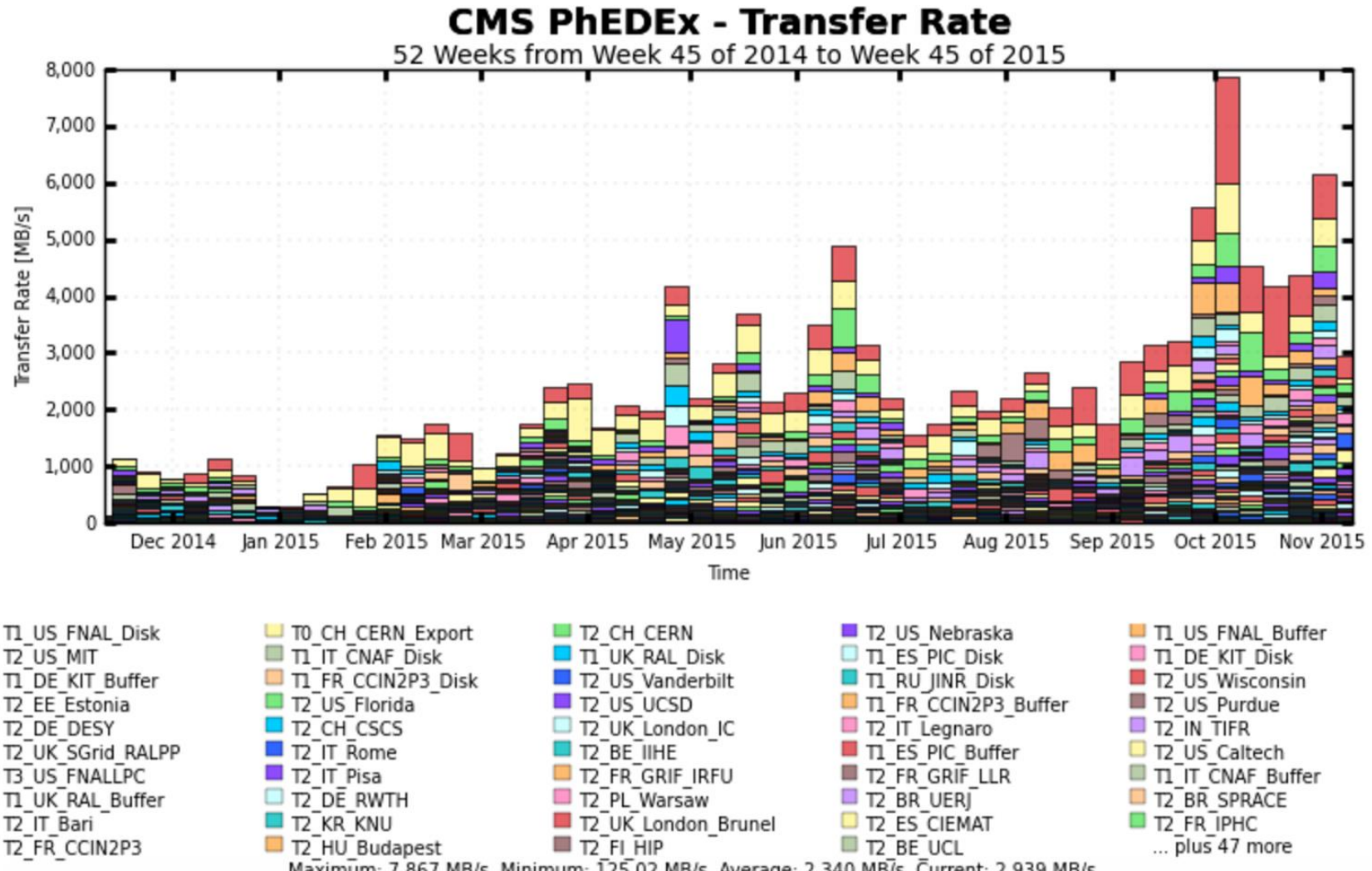
- Collects 1Gb/race
- Analysing in real-time to tune and manage the car



Large Hadron Collider

Compact Muon Solenoid Experiment

Source: http://cmsweb.cern.ch/phedex



Questions?