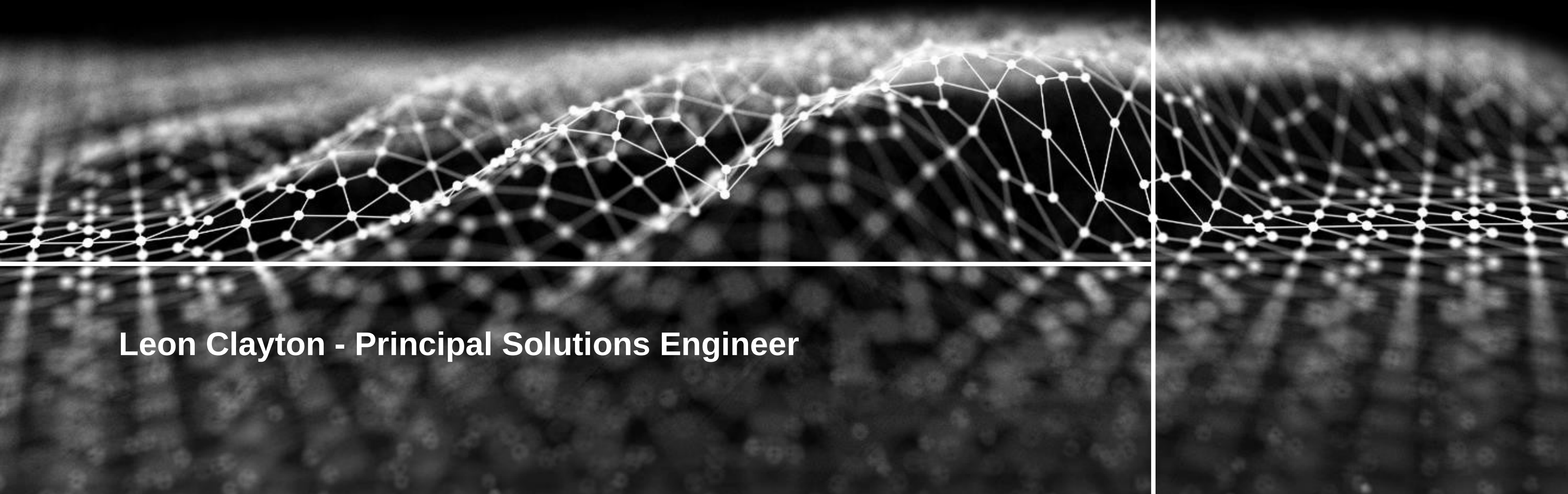
Hadoop Essentials

MAPR®



Leon Clayton - Principal Solutions Engineer

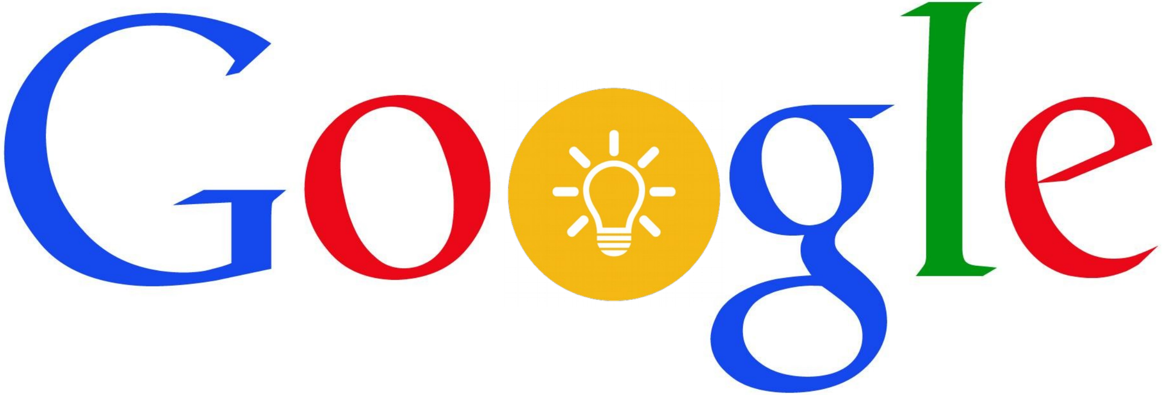
{"about" : "me"}

Leon Clayton

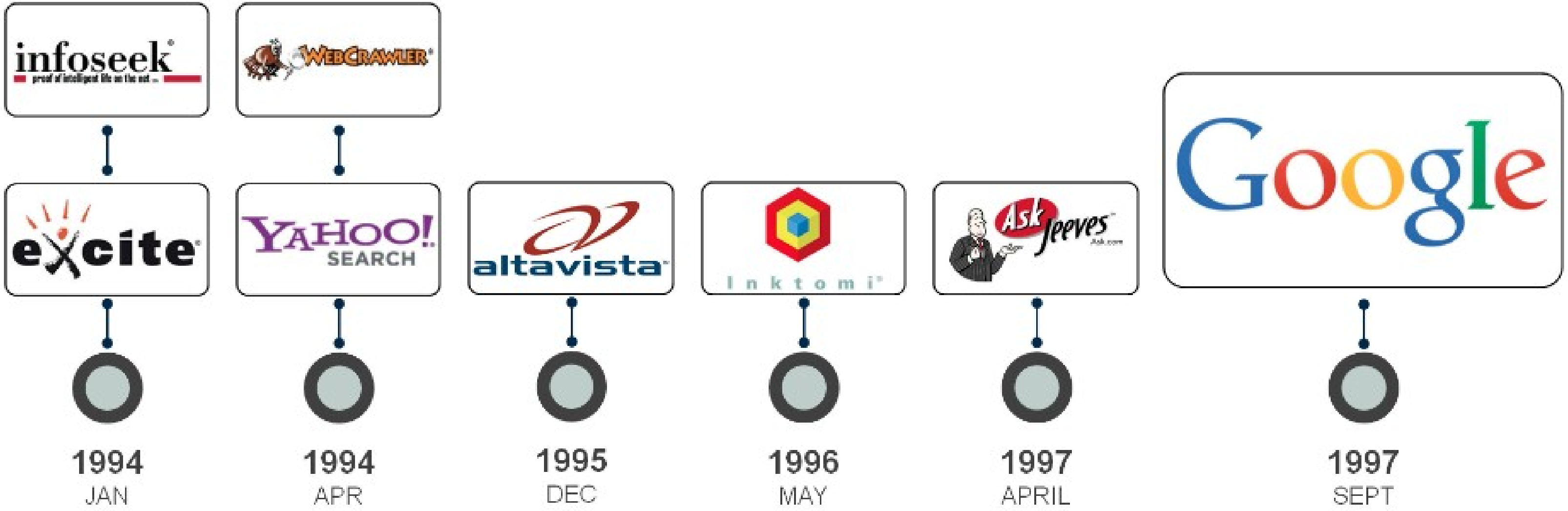
- MapR
  - Solution Architect
- EMC
  - EMEA Isilon Lead
- NetApp
  - Performance Team Lead
- Philips
  - HPC team lead
- Royal Navy
  - Type 42 – Real time systems engineer

- lclayton@mapr.com

Google



History of Hadoop & Map Reduce



1994 JAN

1994 APR

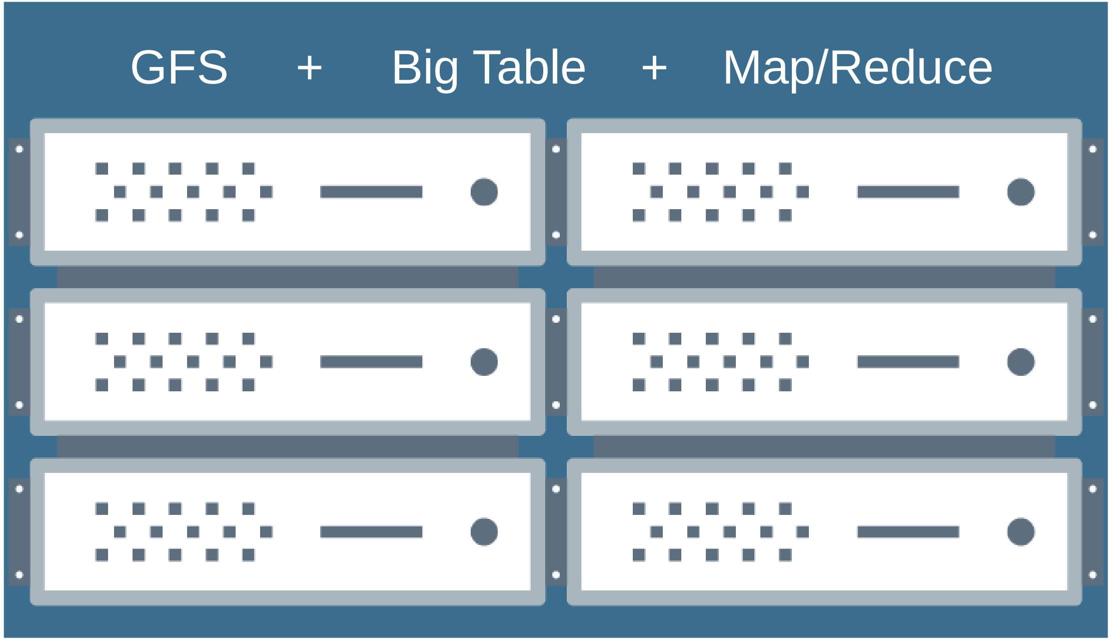
1995 DEC

1996 MAY

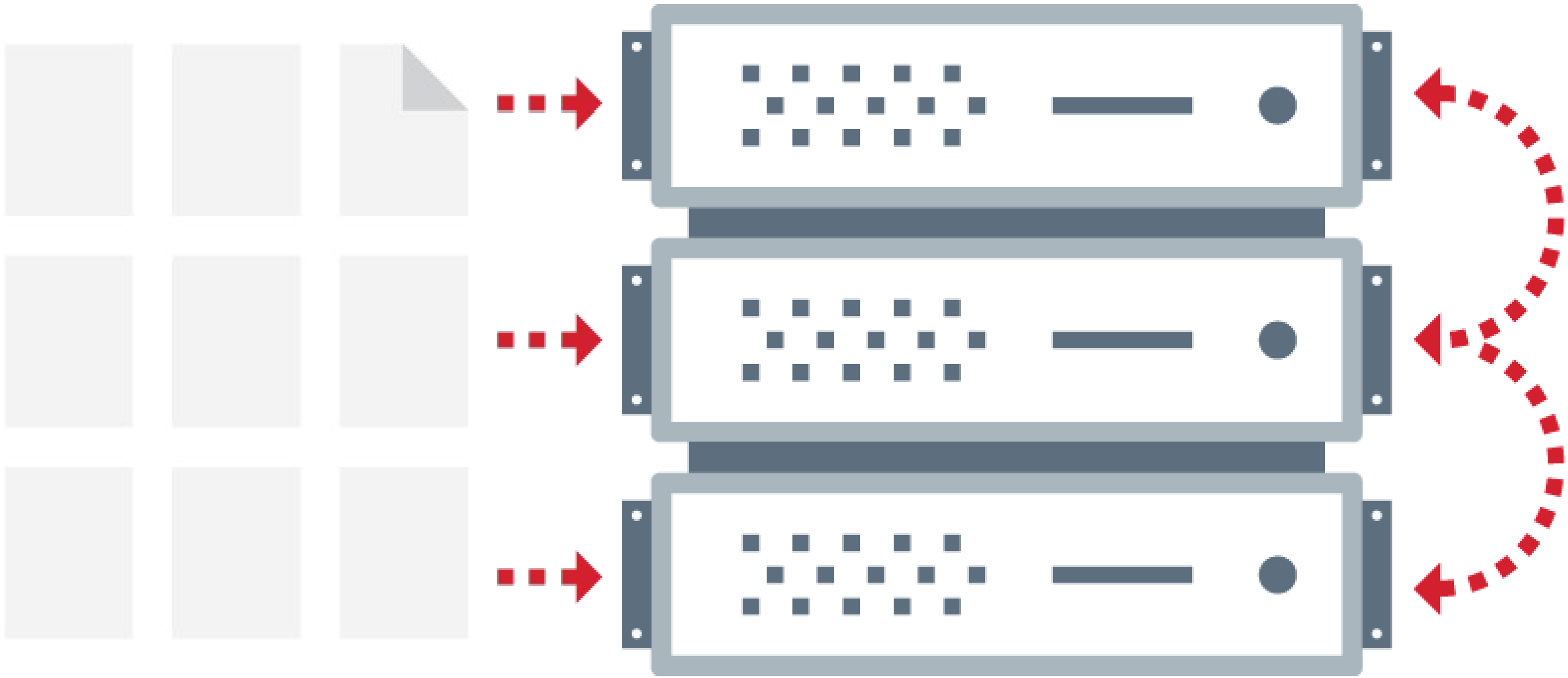
1997 APRIL

1997 SEPT

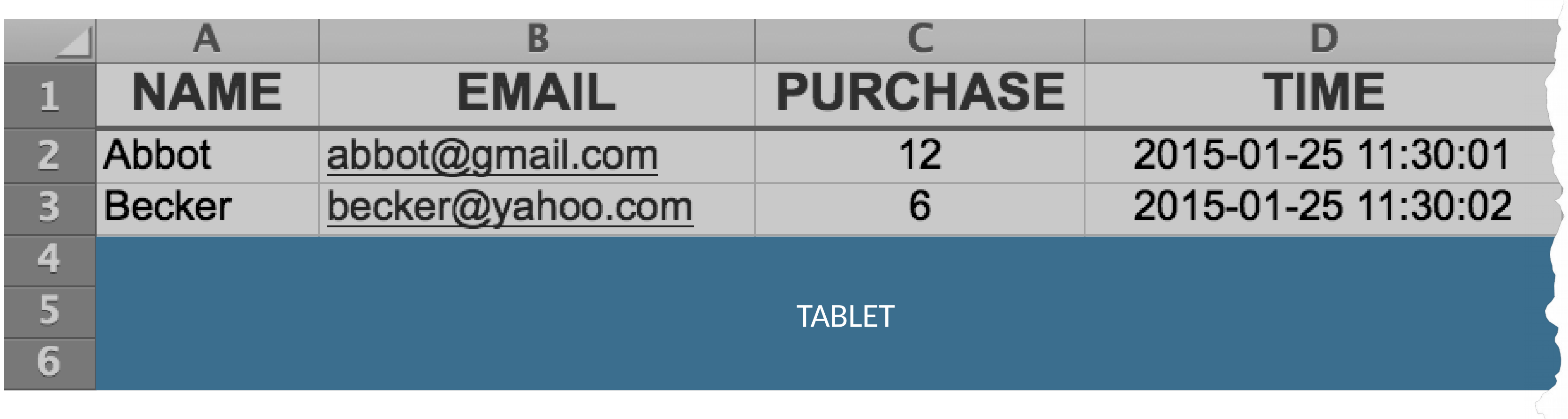
Google Solution



Google File System (GFS)



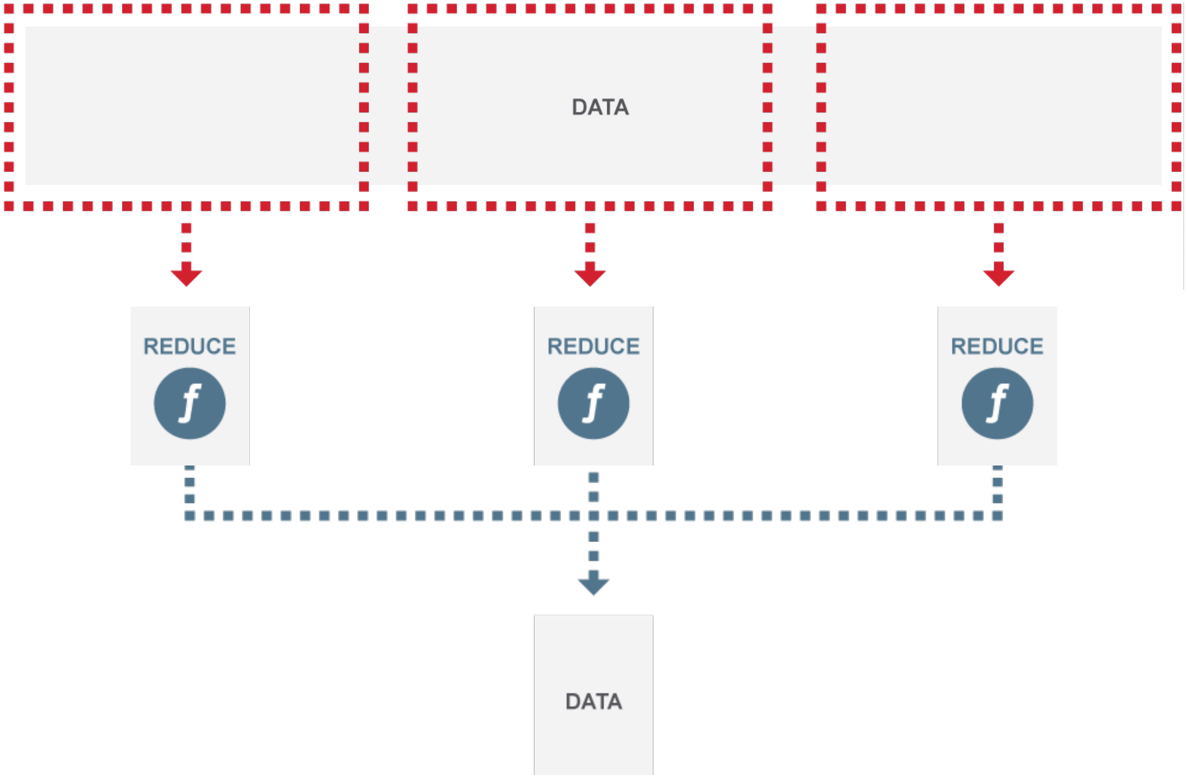
BigTable



	A	B	C	D
1	NAME	EMAIL	PURCHASE	TIME
2	Abbot	abbot@gmail.com	12	2015-01-25 11:30:01
3	Becker	becker@yahoo.com	6	2015-01-25 11:30:02
4				
5				
6				

TABLET

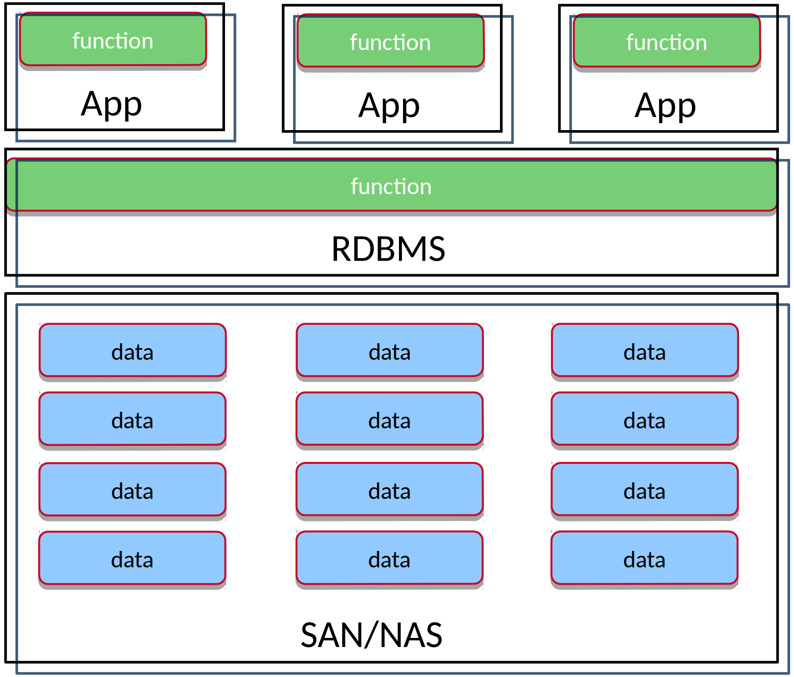
Map/Reduce



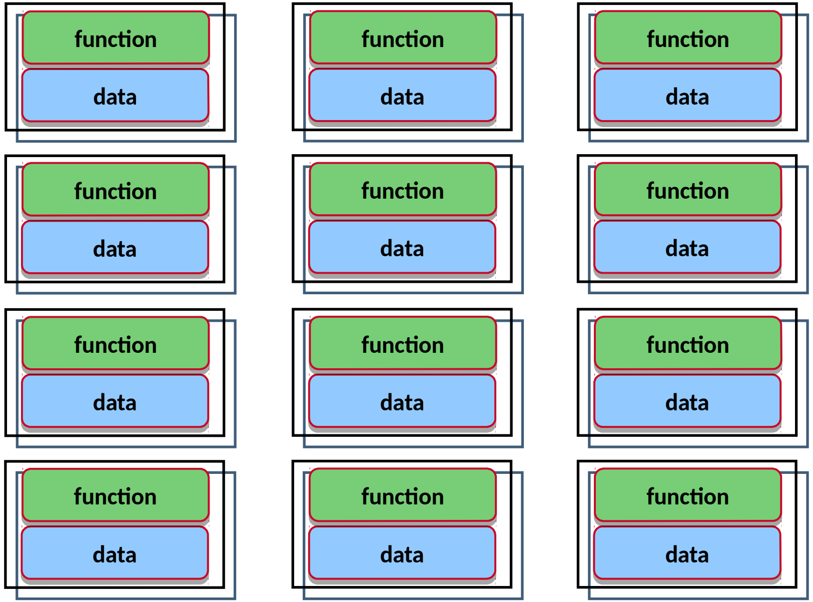
Hadoop

Exploit Locality of Reference

Traditional Architecture



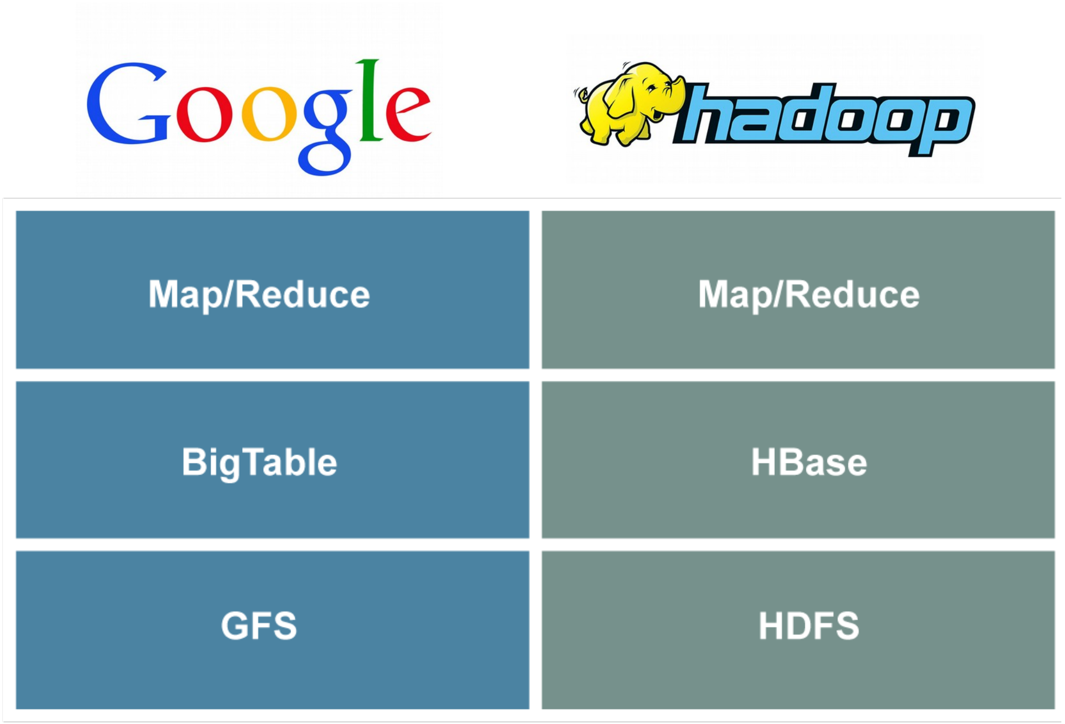
Hadoop Architecture



Hadoop

- MapReduce is a programing model for distributed computing
  - Google popularized it - 2004
  - Apache Hadoop copied it - 2005
  - MapR Technologies improved it - 2009
- Hadoop storage and analysis:
  - 10.000s of disks on
  - 1000s of machines with near linear scaling
  - Commodity hardware (CPU, RAM, disk, etc...)
  - No specialized hardware
  - Handle Big Data – Petabytes and more

Major Components of Hadoop



Google	hadoop
Map/Reduce	Map/Reduce
BigTable	HBase
GFS	HDFS

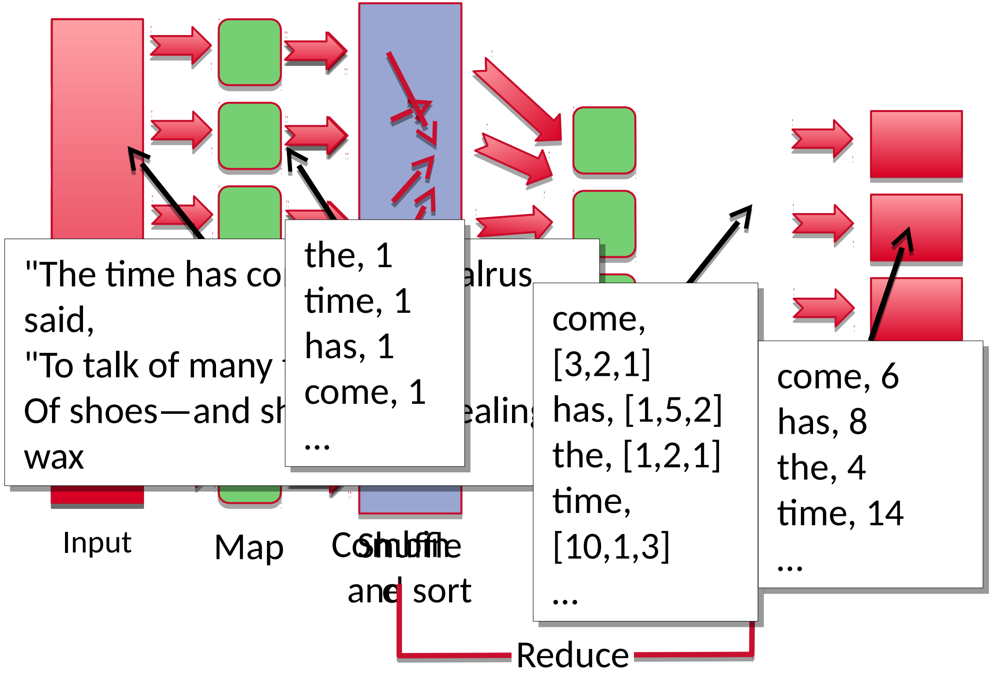
MapR

Google	hadoop	MAPR
Map/Reduce	Map/Reduce	Map/Reduce
BigTable	HBase	MapR/DB
GFS	HDFS	MapR/FS

Phases of MapReduce

- Map
  - Job partitioned into “splits”
- Combine (optional)
  - Makes work easier for the reducer(s)
- Shuffle and sort
  - Map output sent to reducer(s) using a hash
- Reduce

Inside MapReduce



"The time has co alrus
said,
"To talk of many t
Of shoes—and sh ealin
wax

the, 1
time, 1
has, 1
come, 1
…

come,
[3,2,1]
has, [1,5,2]
the, [1,2,1]
time,
[10,1,3]
…

come, 6
has, 8
the, 4
time, 14
…

Hive and Pig

- Hive: data warehousing application in Hadoop
  - Query language is HQL, variant of SQL
  - Tables stored on HDFS as flat files
  - Developed by Facebook, now open source
- Pig: large-scale data processing system
  - Scripts are written in Pig Latin, a dataflow language
  - Developed by Yahoo!, now open source
  - Roughly 1/3 of all Yahoo! internal jobs
- Common idea:
  - Provide higher-level language to facilitate large-data processing
  - Higher-level language “compiles down” to MapReduce jobs

Hive Background

- Started at Facebook
- Data was collected by nightly cron jobs into Oracle DB
- “ETL” via hand-coded python
- Grew from 10s of GBs (2006) to 1 TB/day new data (2007), now 10x that

Source: cc-licensed slide by Cloudera

Hive Data Model

- Tables
  - Typed columns (int, float, string, boolean)
  - Also, list: map (for JSON-like data)
- Partitions
  - For example, range-partition tables by date
- Buckets
  - Hash partitions within ranges (useful for sampling, join optimization)

Source: cc-licensed slide by Cloudera

Hive: Example

- Hive looks similar to an SQL database
- Relational join on two tables:
  - Table of word counts from Shakespeare collection
  - Table of word counts from the bible

```
SELECT s.word, s.freq, k.freq FROM shakespeare s
  JOIN bible k ON (s.word = k.word) WHERE s.freq >= 1 AND k.freq >= 1
  ORDER BY s.freq DESC LIMIT 10;
```

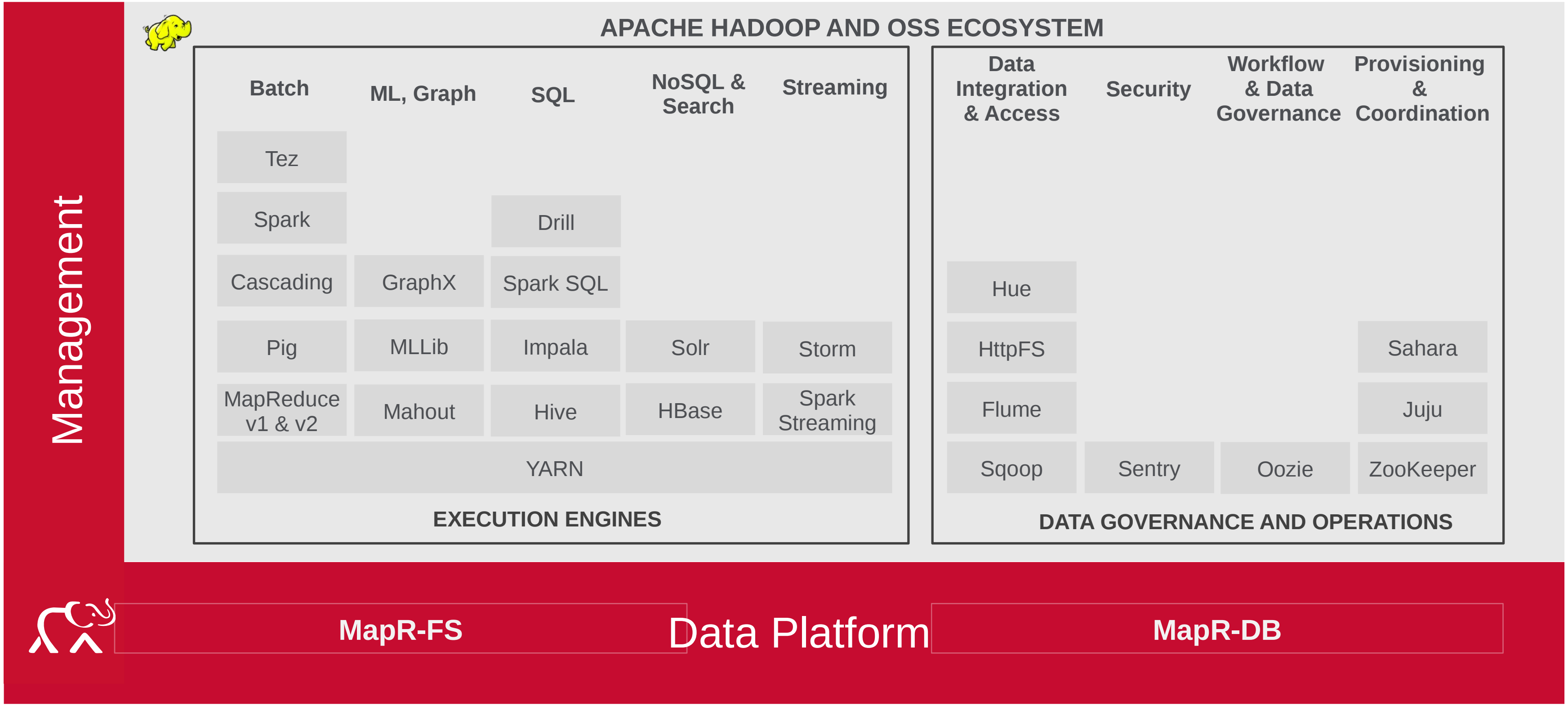
```
the     25848  62394
I       23031  8854
and     19671  38985
to      18038  13526
of      16700  34654
a       14170  8057
you     12702  2720
my      11297  4135
in      10797  12445
is      8882   6884
```

Hadoop Coding Example

– https://www.tutorialspoint.com/hadoop/hadoop_mapreduce.htm

Lets talk through this

The Power of the Open Source Community



APACHE HADOOP AND OSS ECOSYSTEM

Management

Batch	ML, Graph	SQL	NoSQL & Search	Streaming
Tez				
Spark		Drill		
Cascading	GraphX	Spark SQL		
Pig	MLLib	Impala	Solr	Storm
MapReduce v1 & v2	Mahout	Hive	HBase	Spark Streaming
YARN				

EXECUTION ENGINES

Data Integration & Access	Security	Workflow & Data Governance	Provisioning & Coordination
Hue			
HttpFS			Sahara
Flume			Juju
Sqoop	Sentry	Oozie	ZooKeeper

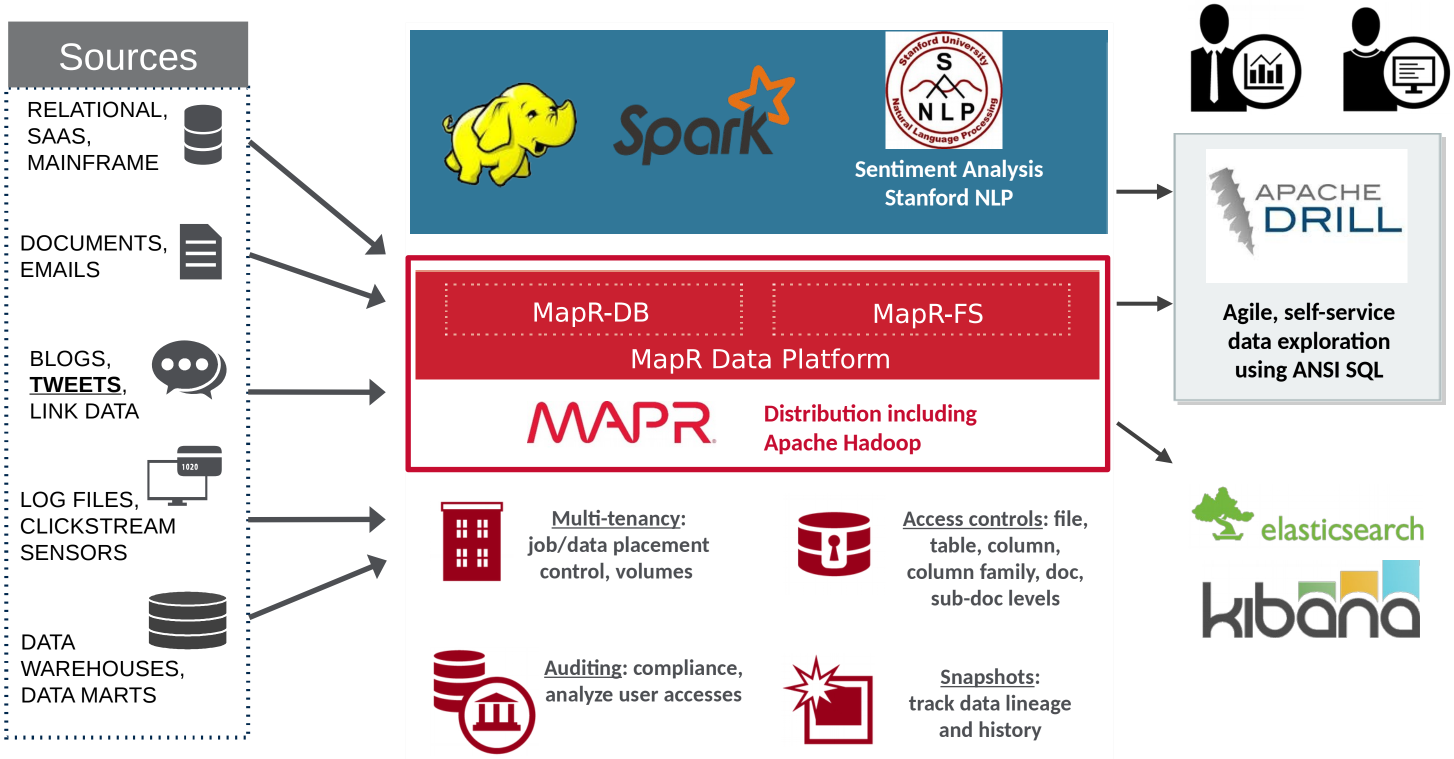
DATA GOVERNANCE AND OPERATIONS

MapR-FS

Data Platform

MapR-DB

Example Big Data Architecture – Hands on architecture



Sources

RELATIONAL, SAAS, MAINFRAME

DOCUMENTS, EMAILS

BLOGS, **TWEETS**, LINK DATA

LOG FILES, CLICKSTREAM SENSORS

DATA WAREHOUSES, DATA MARTS

Sentiment Analysis
Stanford NLP

MapR-DB

MapR-FS

MapR Data Platform

Distribution including
Apache Hadoop

Multi-tenancy: job/data placement control, volumes

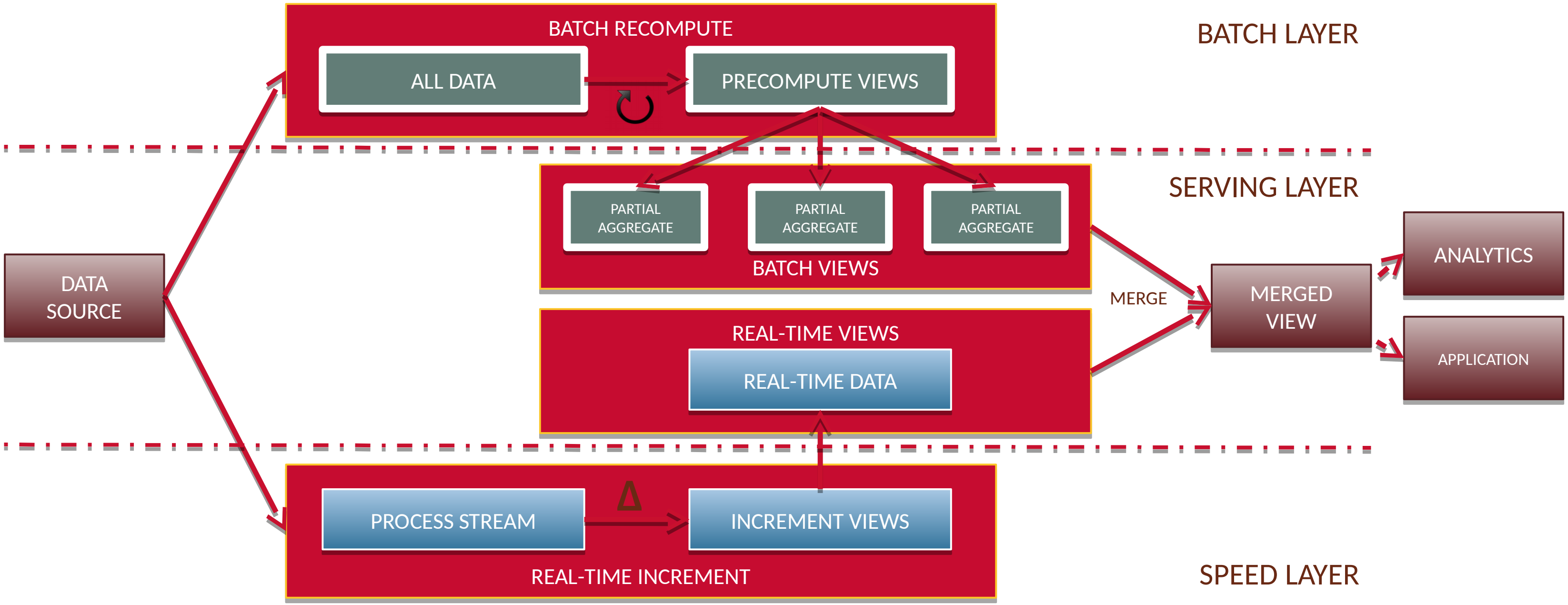
Access controls: file, table, column, column family, doc, sub-doc levels

Auditing: compliance, analyze user accesses

Snapshots: track data lineage and history

Agile, self-service data exploration using ANSI SQL

Lambda Architecture

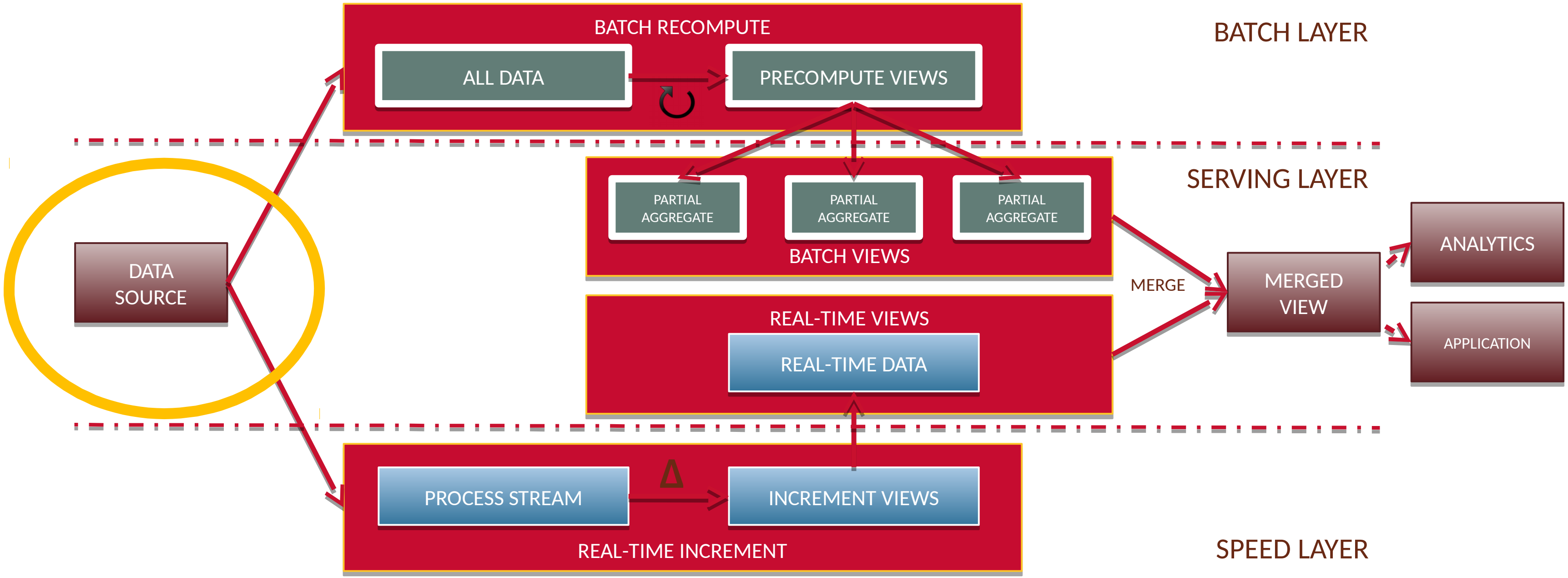


Lambda Architecture

Overview

- Invented by Nathan Marz at Twitter
- Very fault tolerant, because of recomputation
- Combines the complete history with the newest data
- Real-time results

Lambda Architecture

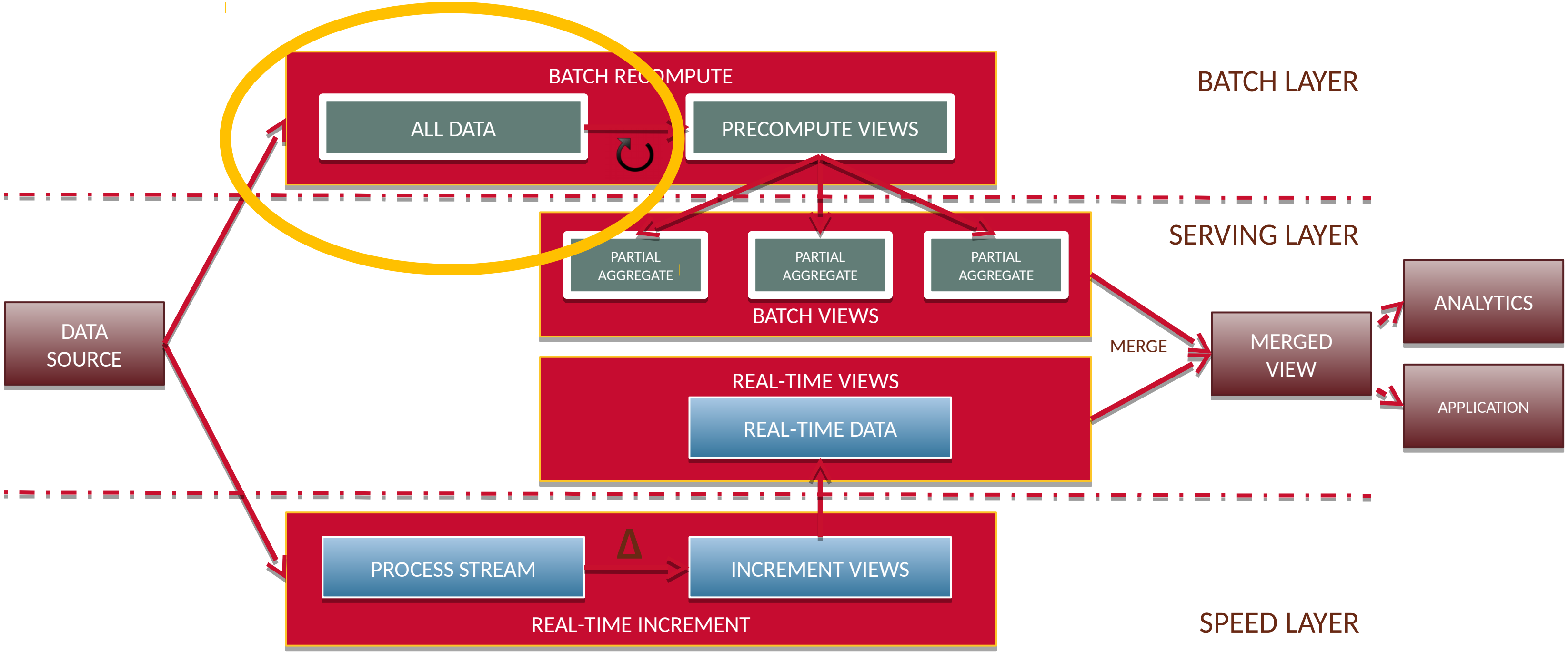


Lambda Architecture

Data ingest

- A message queue that feeds the system
- Apache Kafka

Lambda Architecture

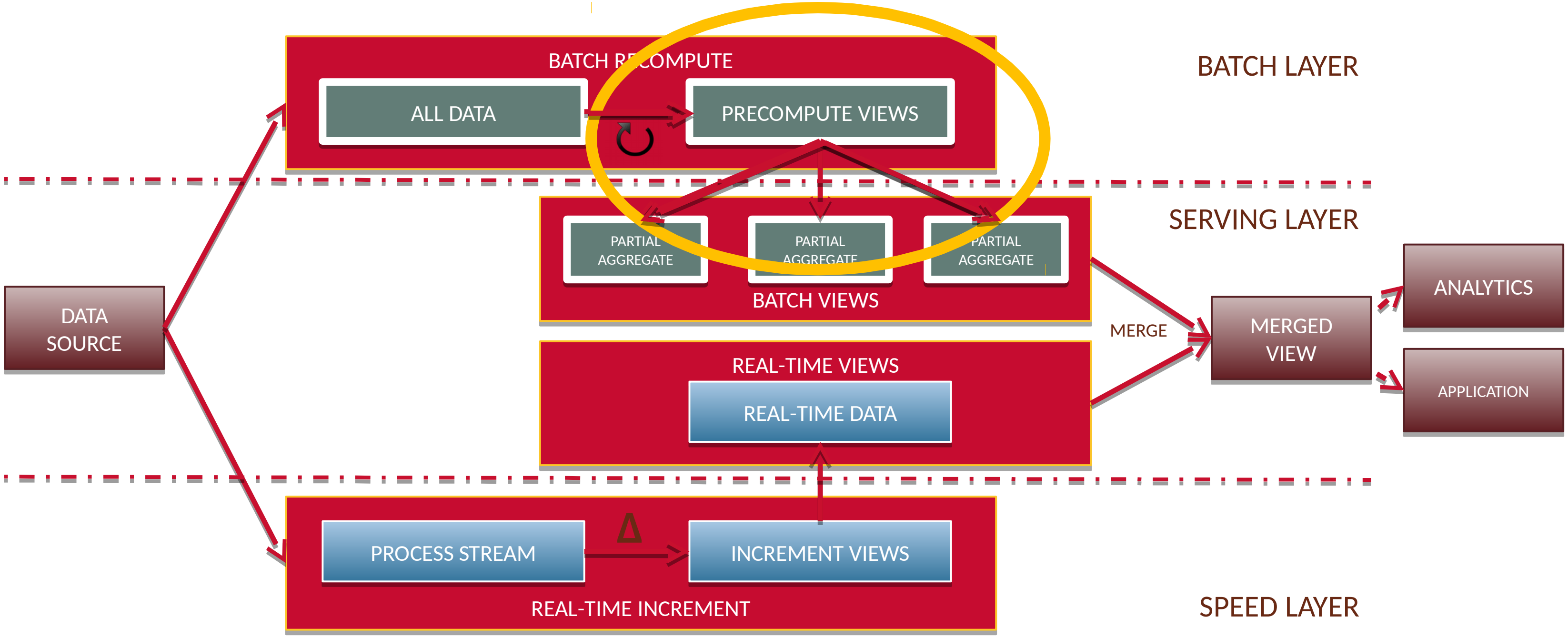


Lambda Architecture

Master Data Set

- Append only
- Data is stored raw
- MapR-FS / HDFS
- Different users for appending and reading

Lambda Architecture

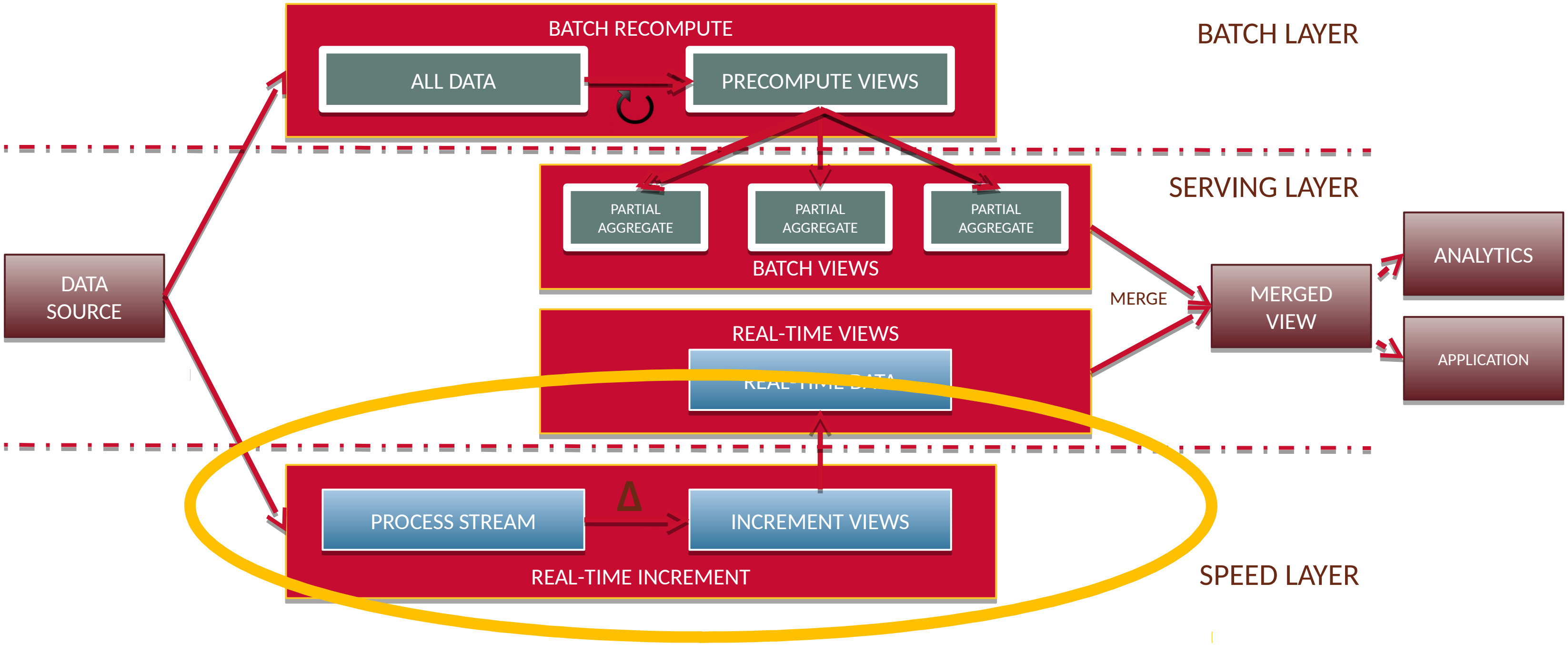


Lambda Architecture

Batch processes

- Runs periodically
- If a bug wreaks havoc, just fix + recompute
- Data that comes in while a new batch is processing goes via the speed layer
- Typically compute aggregates or train models

Lambda Architecture

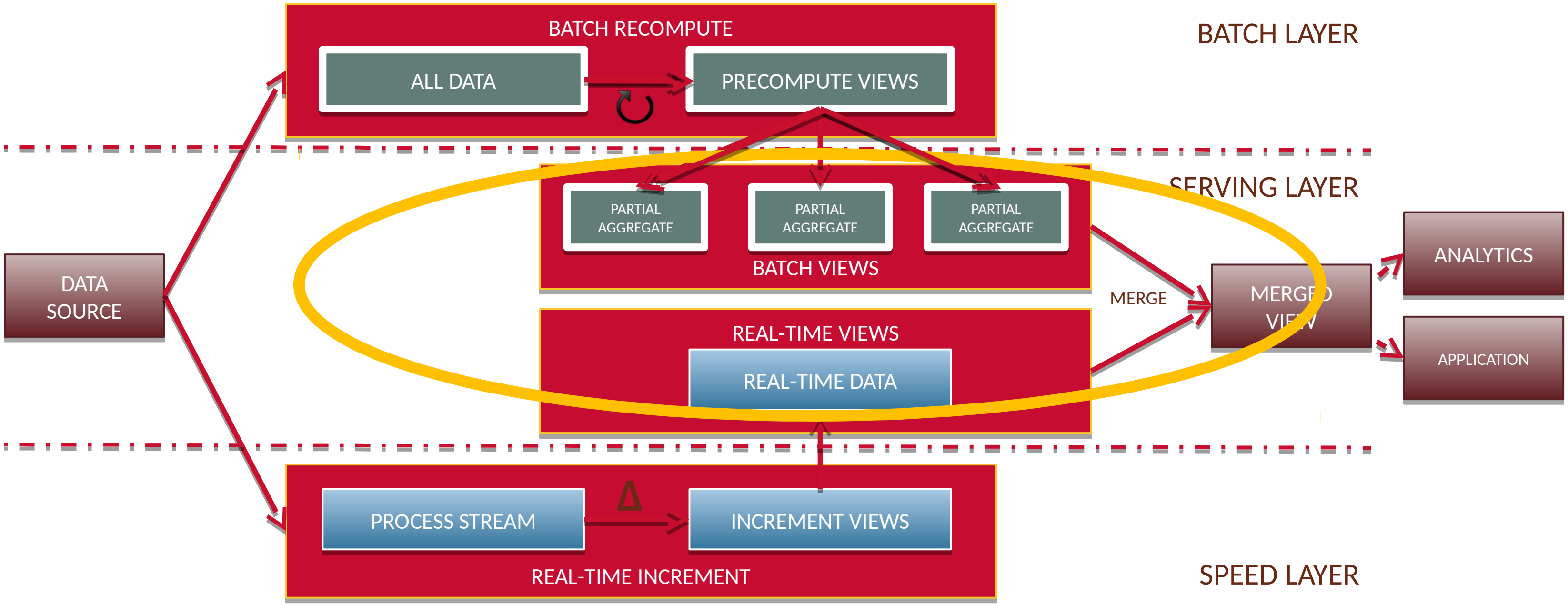


Lambda Architecture

Streaming

- Process events immediately as they arrive
- Produces incremental updates or predictions
- E.g. “+1 view for URL *x*” or “Customer *y* belongs to cluster *z*”

Lambda Architecture



Lambda Architecture

Views

- This is very application specific
  - Aggregates might fit in a regular SQL table
  - The merged view might need some custom work
- Implementations range from SQL to NoSQL
  - Postgres, MS-SQL
  - Hbase (with Phoenix)
  - Cassandra
  - Elastic
  - …

Lambda Architecture

Lambda + Spark = profit

- “Code duplication” biggest criticism
- Spark offers batch and streaming paradigms
- At least event interpretation can be shared

Learn more: