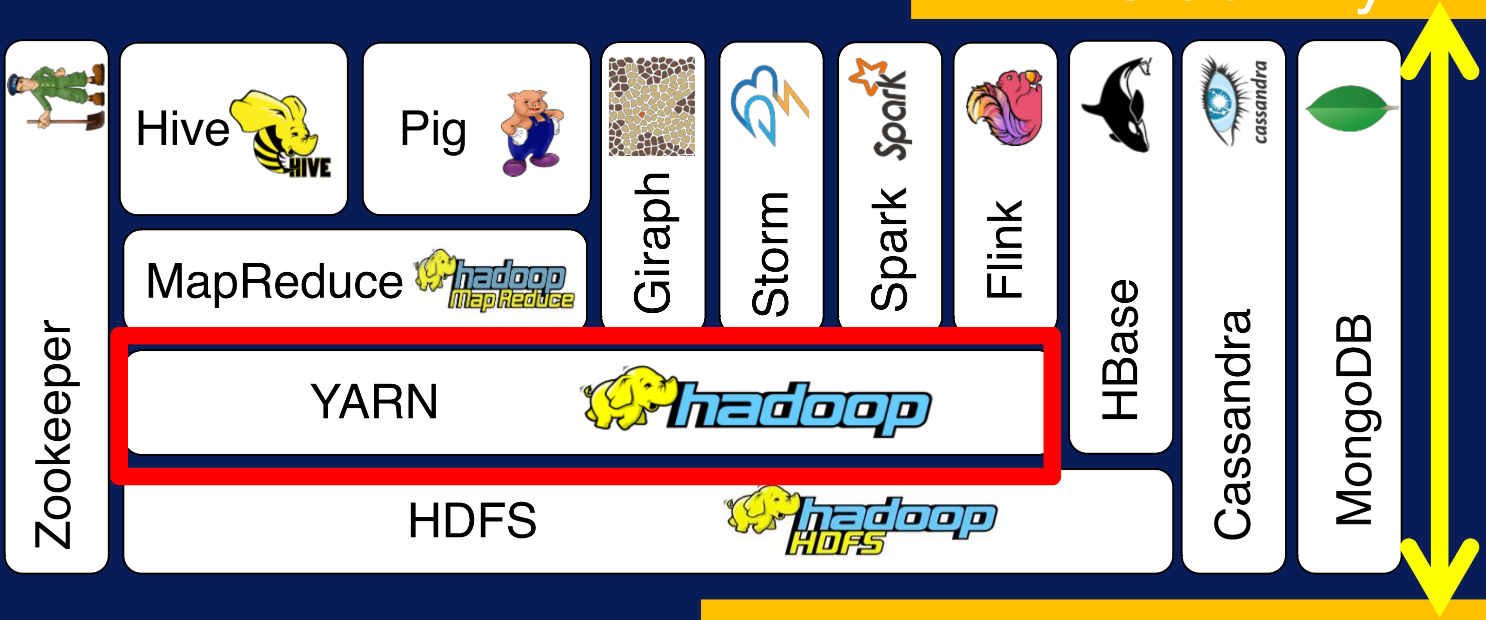
Overview of Big Data Processing Systems

After this video you will be able to..

- Recall the Hadoop Ecosystem
- Draw a layer diagram with three layers for data storage, data processing and workflow management
- Summarize an evaluation criteria for big data processing systems
- Explain the properties of Hadoop, Spark, Flink, Beam and Storm

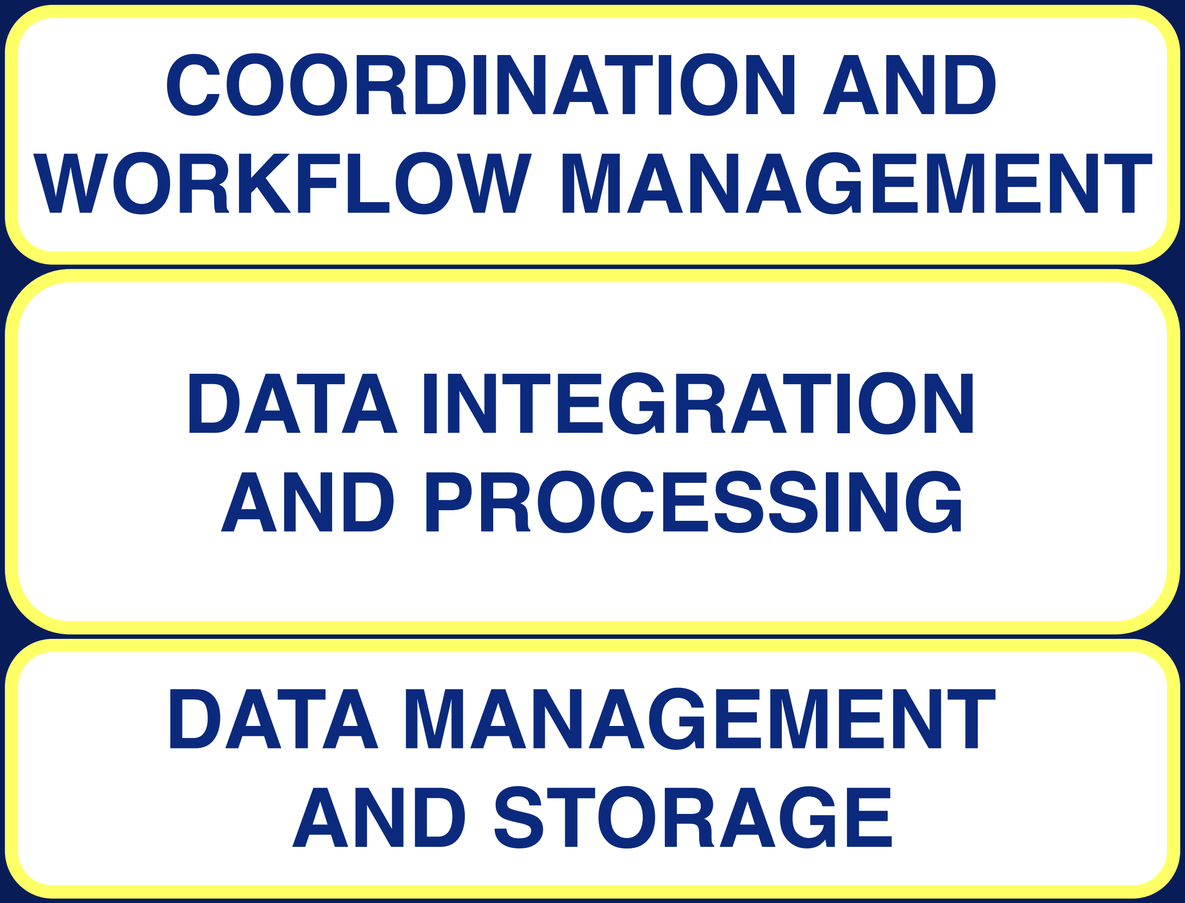
One possible layer diagram for Hadoop tools



Higher levels: Interactivity

Lower levels: Storage and scheduling

Another way to look at the Hadoop Ecosystem

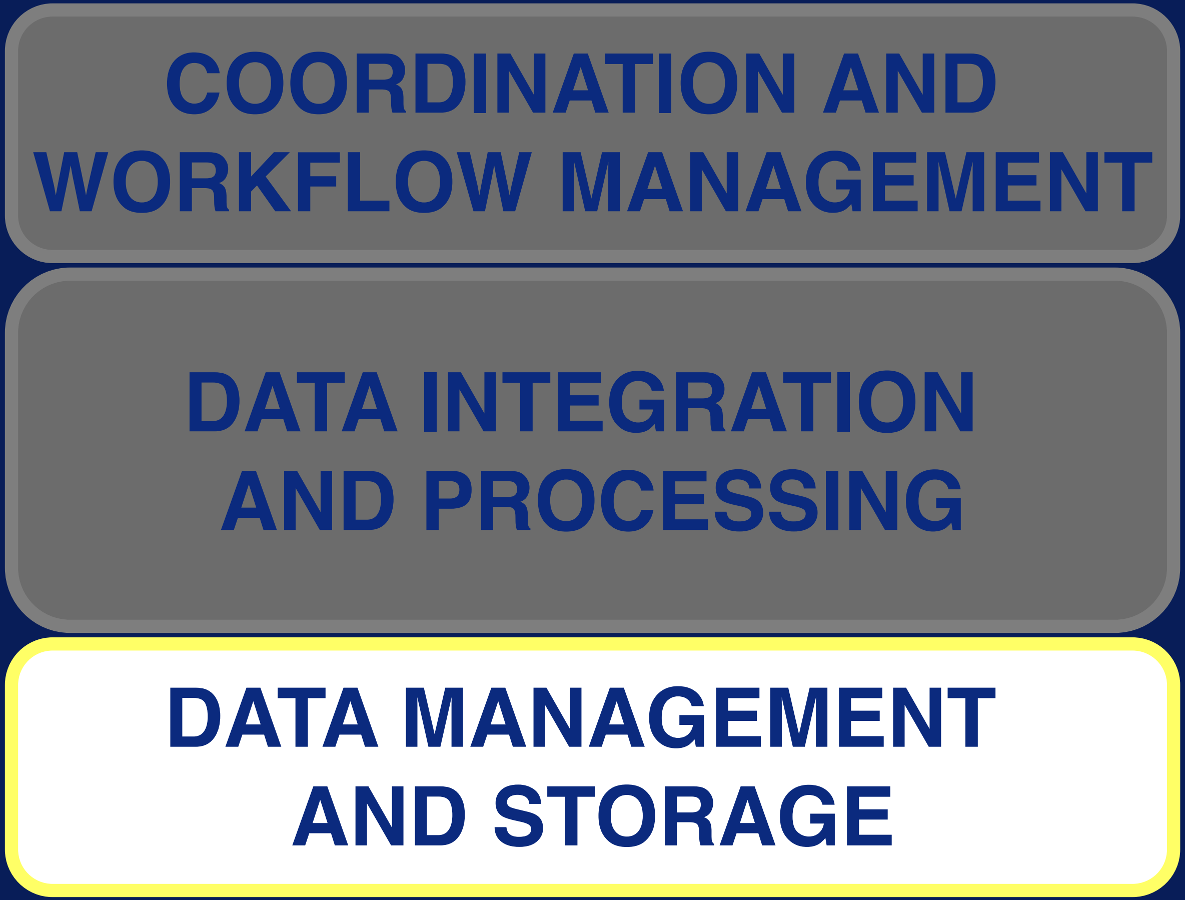


COORDINATION AND WORKFLOW MANAGEMENT

DATA INTEGRATION AND PROCESSING

DATA MANAGEMENT AND STORAGE

Another way to look at the Hadoop Ecosystem



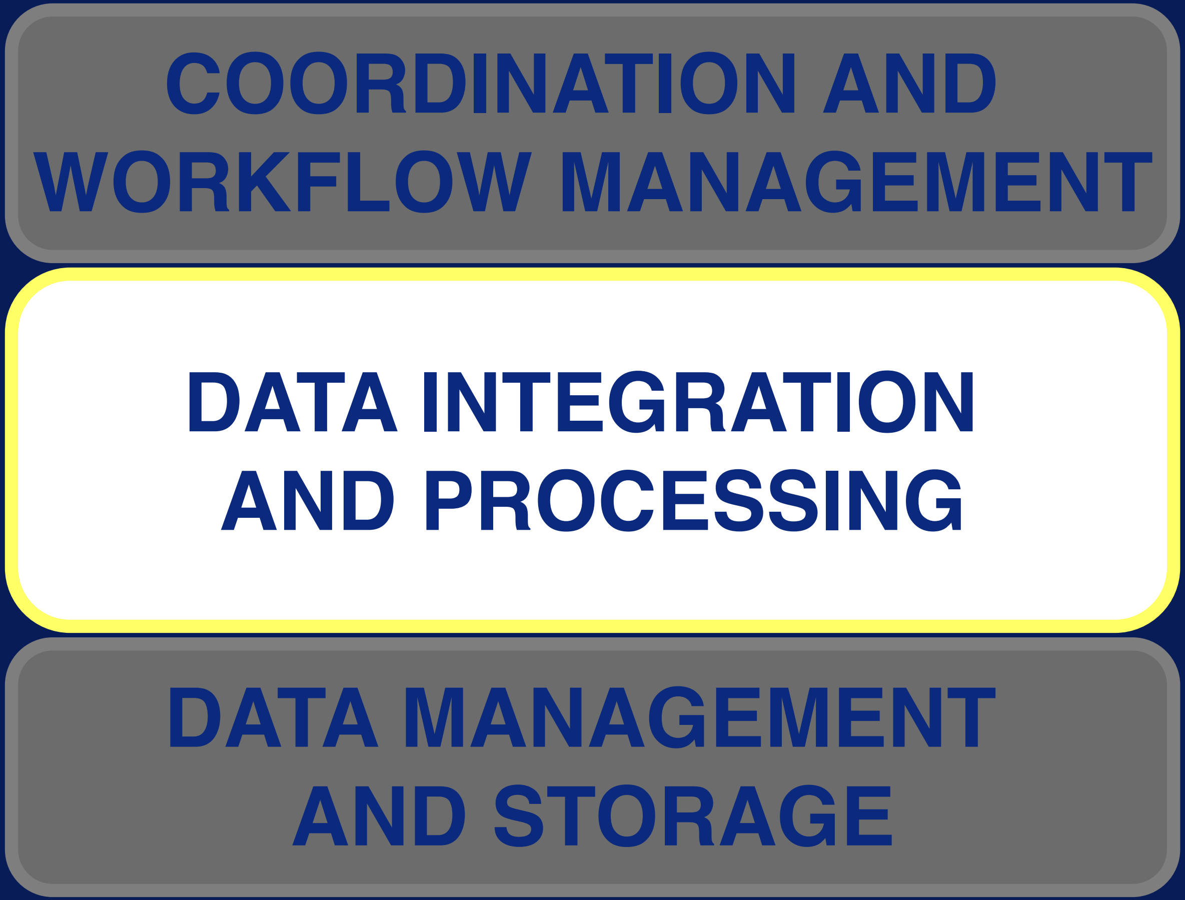
COORDINATION AND WORKFLOW MANAGEMENT

DATA INTEGRATION AND PROCESSING

DATA MANAGEMENT AND STORAGE

DATA MANAGEMENT AND STORAGE

Another way to look at the Hadoop Ecosystem

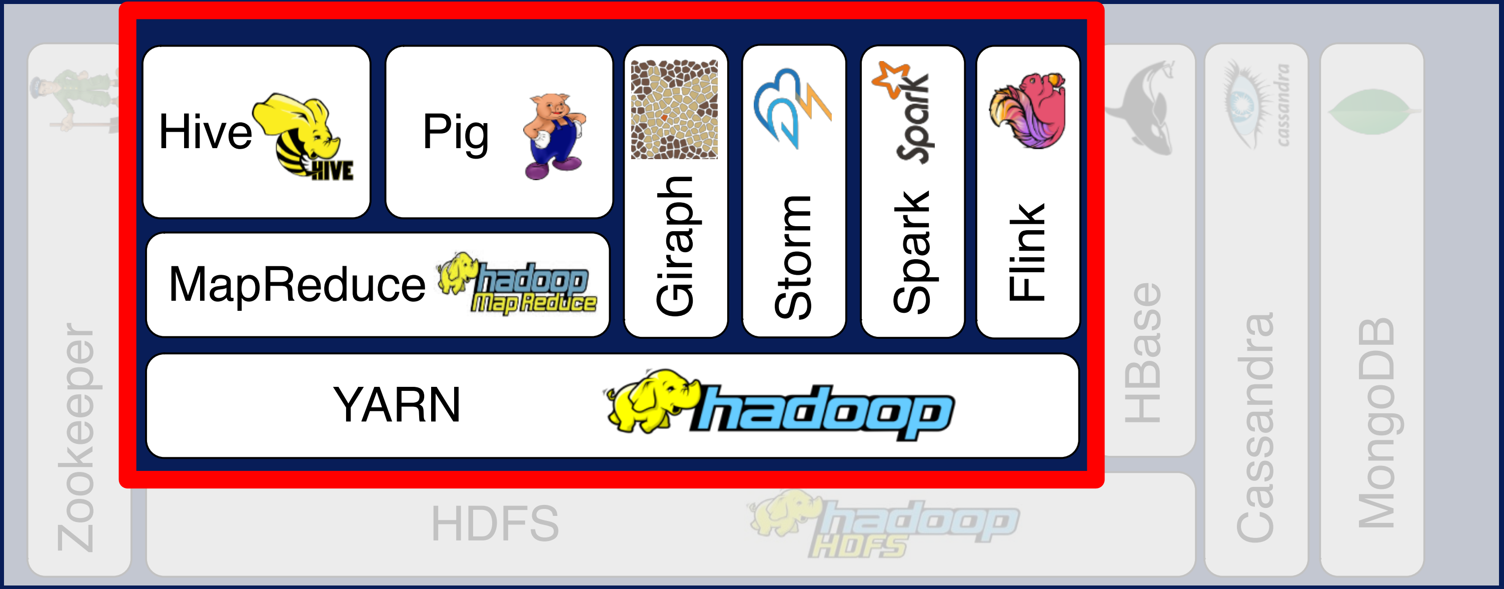


COORDINATION AND WORKFLOW MANAGEMENT

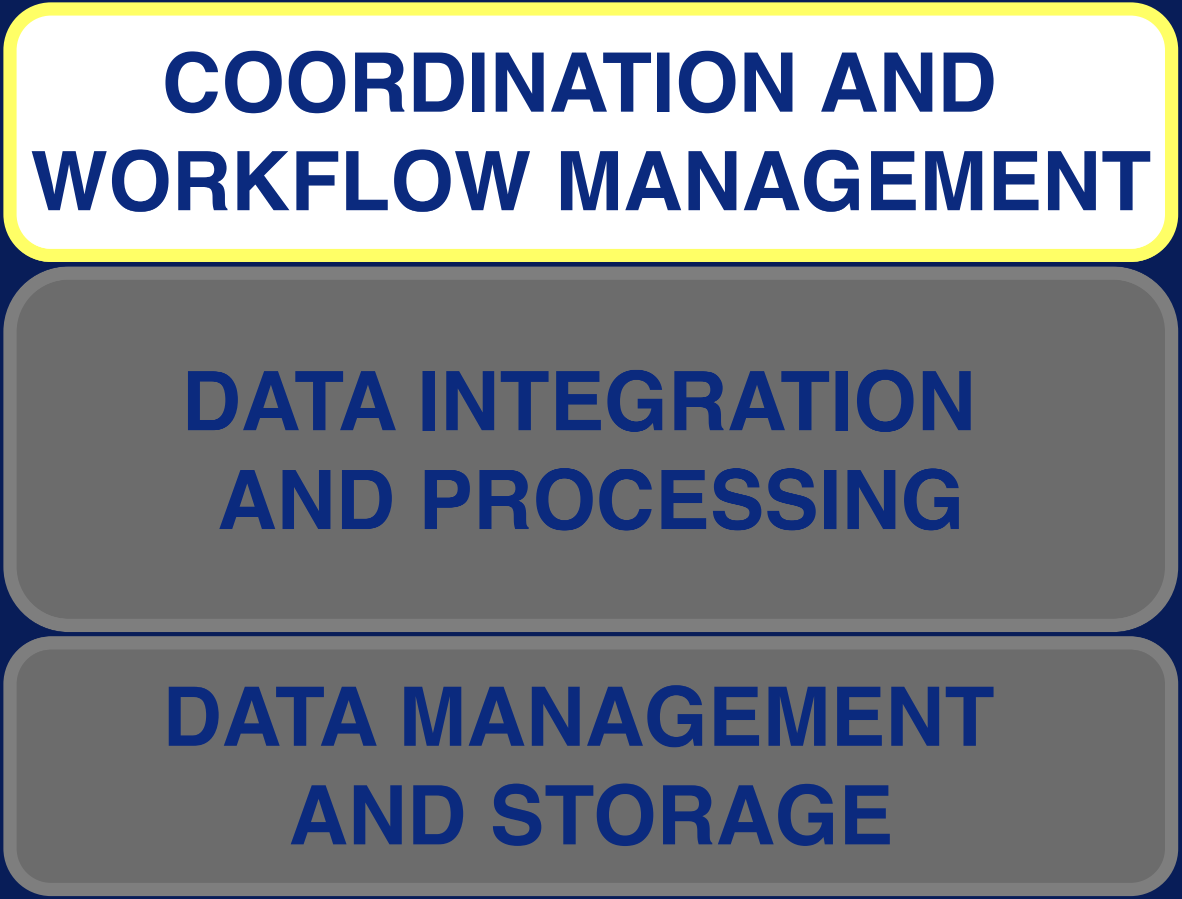
DATA INTEGRATION AND PROCESSING

DATA MANAGEMENT AND STORAGE

DATA INTEGRATION AND PROCESSING



Another way to look at the Hadoop Ecosystem



COORDINATION AND WORKFLOW MANAGEMENT

DATA INTEGRATION AND PROCESSING

DATA MANAGEMENT AND STORAGE

- where integration, scheduling, coordination and monitoring of applications across many tools take place
- where results of big data analysis gets communicated to other programs, websites, visualization tools, and business intelligence tools

COORDINATION AND WORKFLOW MANAGEMENT

develop automated solutions to manage and coordinate the process of combining data management and analytical tasks in big data pipeline ?

ACQUIRE

PREPARE

ANALYZE

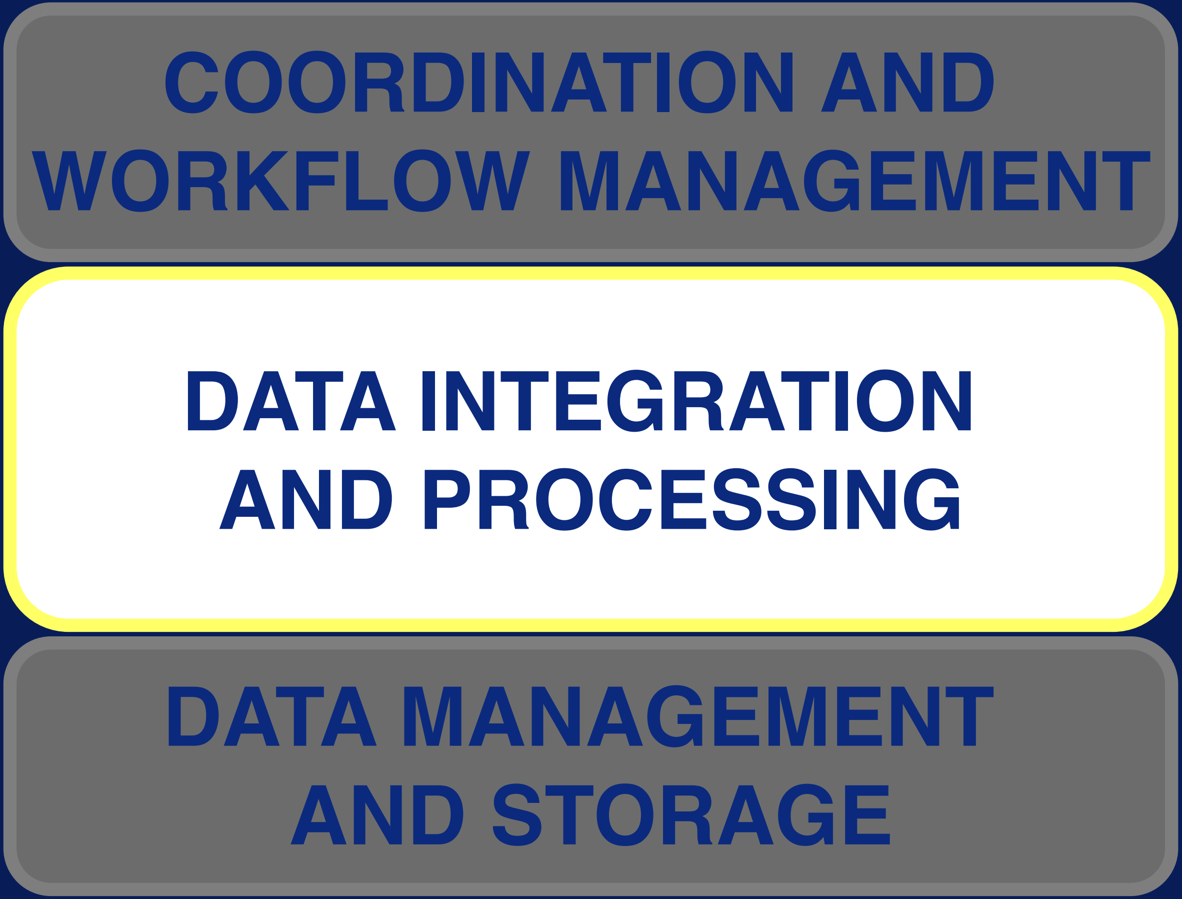
REPORT

ACT

OOZIE

Apache Zookeeper

Another way to look at the Hadoop Ecosystem



COORDINATION AND WORKFLOW MANAGEMENT

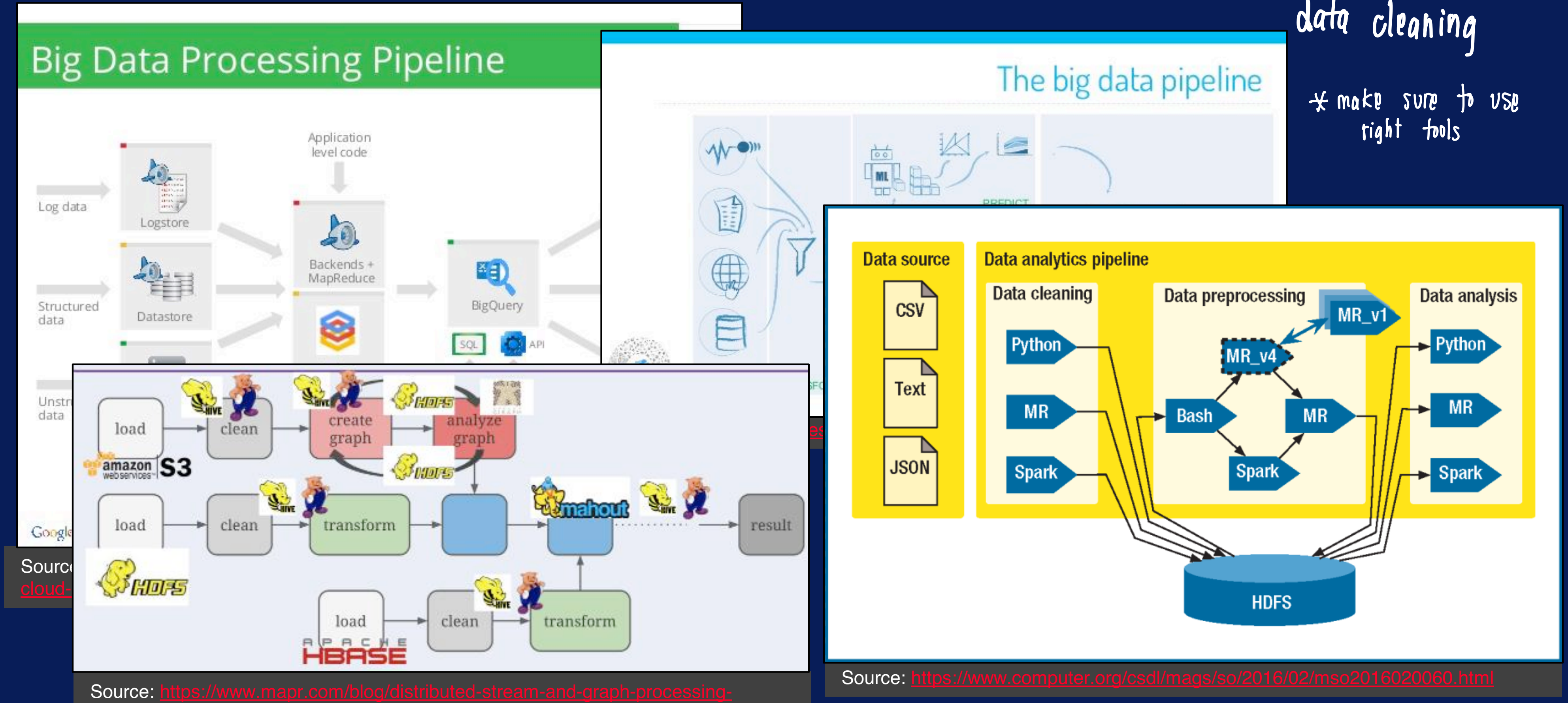
DATA INTEGRATION AND PROCESSING

DATA MANAGEMENT AND STORAGE

Example Big Data Processing Pipelines

data cleaning

* make sure to use right tools



Sourc cloud-

Source: https://www.mapr.com/blog/distributed-stream-and-graph-processing-apache-flink

Source: https://www.computer.org/csdl/mags/so/2016/02/mso2016020060.html

Categorization of Big Data Processing Systems

Execution Model

→ Batch

→ Streaming

→ interactive computing

Latency

Scalability

Programming Language — support for various

Fault Tolerance — how it is handled

Big Data Processing Systems

POWERED BY APACHE™

APACHE STORM™

Distributed · Resilient · Real-time

Flink

hadoop Map Reduce

Spark

beam

MapReduce

Execution Model	Batch processing using disk storage └ data from HDFS gets loaded into mappers before processing
Latency	High-latency └ less scalable execution
Scalability	
Programming Language	Java — others, like python, offer libraries with less efficiency
Fault Tolerance	data Replication └ affects scalability and execution speed further

Spark

to support iterative and
interactive big data processing

Execution Model	Batch and stream processing using disk or memory storage *micro batching
Latency	Low-latency for small micro-batch size
Scalability	
Programming Language	Scala, Python, Java, R
Fault Tolerance	less impact on performance

Flink

└ original version was stratosphere

Execution Model	Batch and stream processing using disk or memory storage
Latency	Low-latency
Scalability	
Programming Language	Java and Scala
Fault Tolerance	

* advantage:
comes from it's optimizer to
pick and apply the best pattern
and execution strategy

Beam

from google

Execution Model	Batch and stream processing
Latency	Low-latency
Scalability	
Programming Language	Java and Scala
Fault Tolerance	

Storm

APACHE STORM™

Distributed · Resilient · Real-time

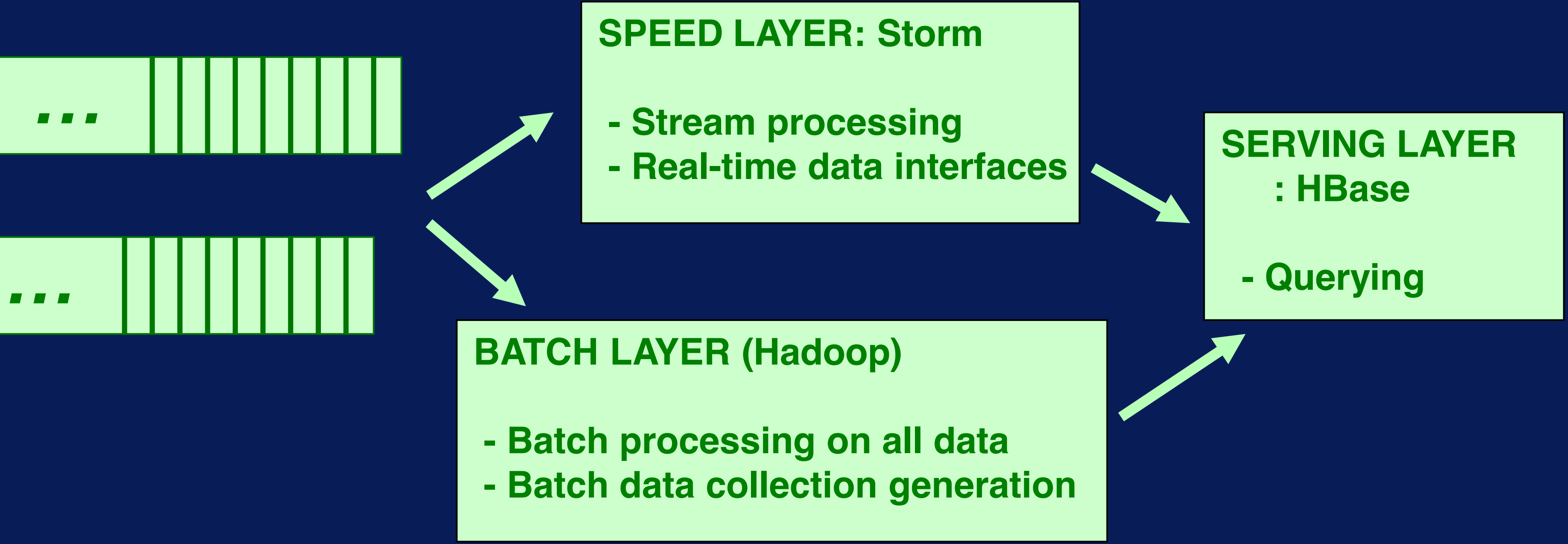
Execution Model	Stream processing
Latency	Very low-latency
Scalability	
Programming Language	Many programming languages
Fault Tolerance	

pipelined together

input stream interface abstractions: spouts

computation abstractions: bolts

Lambda Architecture:
A Hybrid Data Processing Architecture



SPEED LAYER: Storm

- Stream processing
- Real-time data interfaces

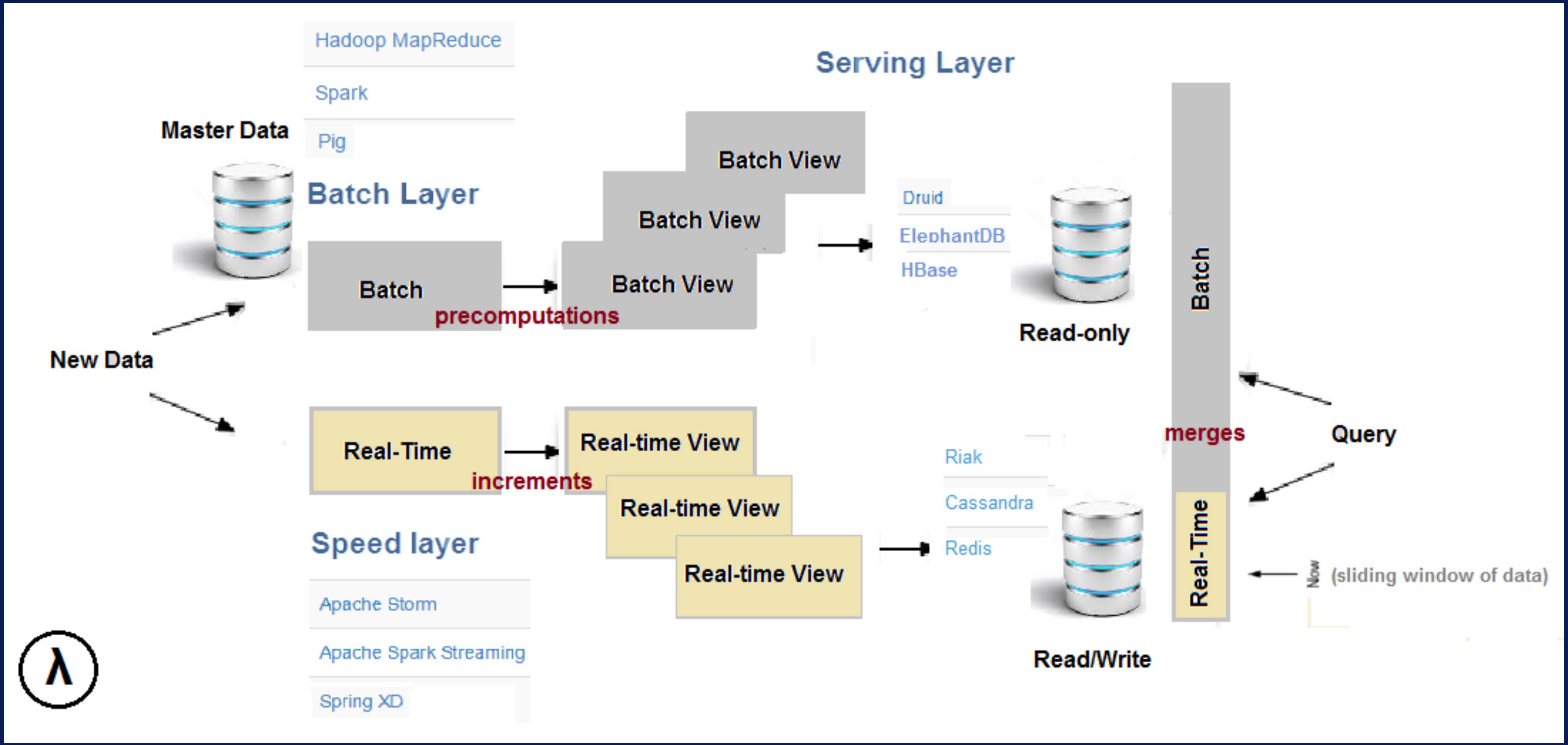
BATCH LAYER (Hadoop)

- Batch processing on all data
- Batch data collection generation

SERVING LAYER
: HBase

- Querying

Lambda Architecture:
A Hybrid Data Processing Architecture


Hadoop MapReduce
Spark
Pig
Master Data
Batch Layer
Batch
precomputations
Batch View
Batch View
Batch View
Serving Layer
Druid
ElephantDB
HBase
Read-only
Batch
New Data
Real-Time
increments
Real-time View
Real-time View
Real-time View
Speed layer
Apache Storm
Apache Spark Streaming
Spring XD
Riak
Cassandra
Redis
Read/Write
merges
Query
Real-Time
Now
(sliding window of data)
λ