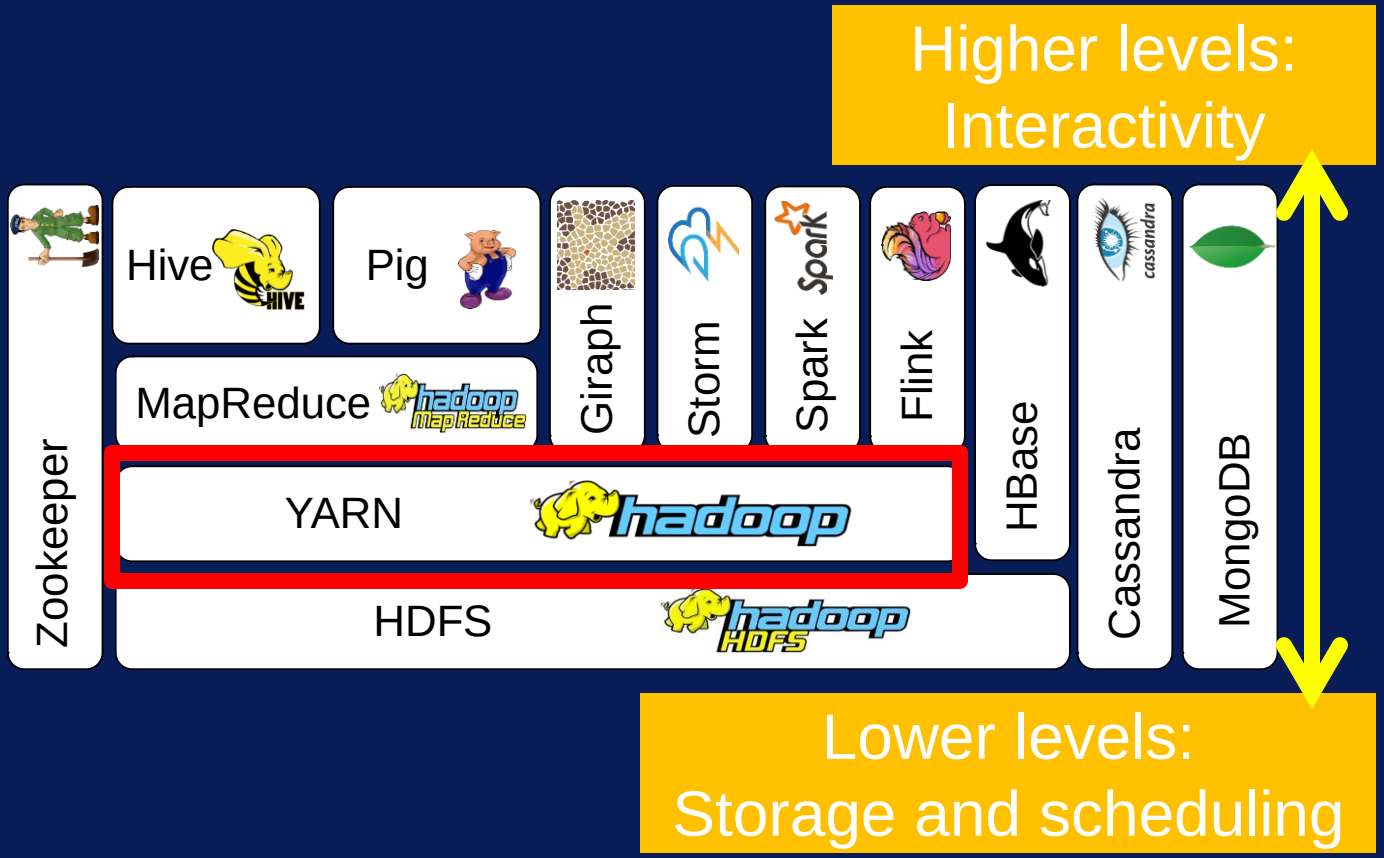
# Overview of Big Data Processing Systems

# 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



# 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







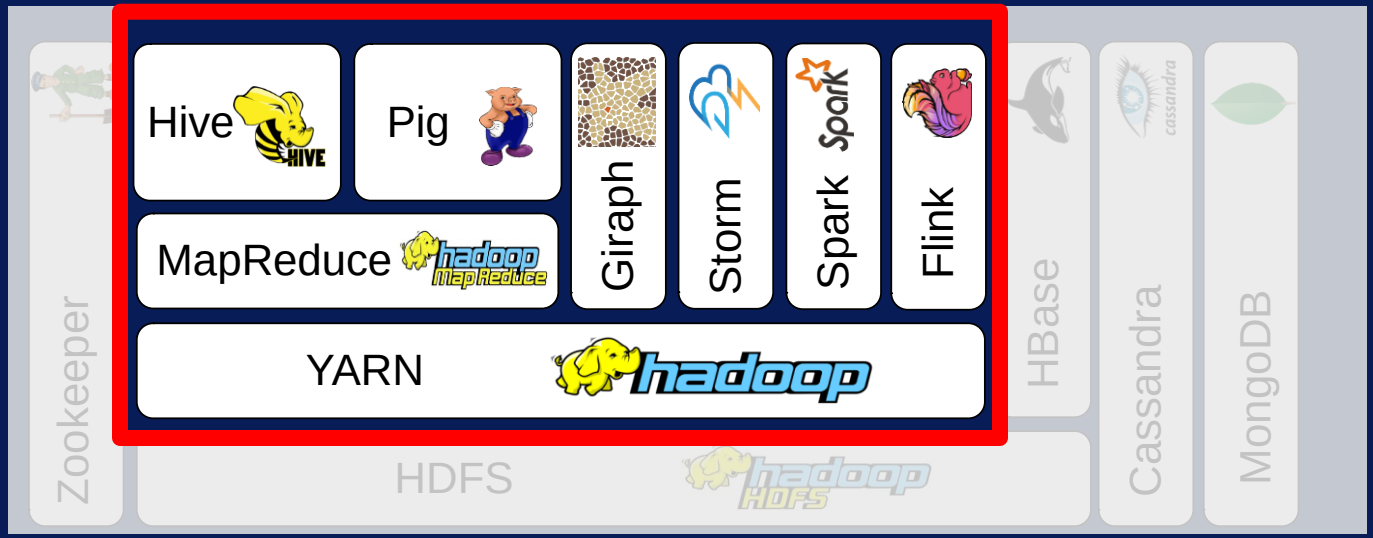

# Another way to look at the Hadoop Ecosystem

**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**

# COORDINATION AND WORKFLOW MANAGEMENT

ACQUIRE → PREPARE → ANALYZE → REPORT → ACT

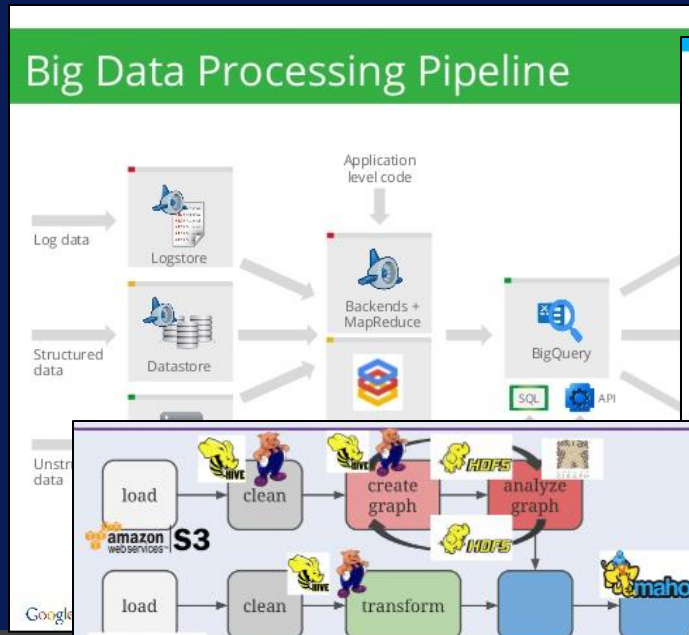

# Another way to look at the Hadoop Ecosystem

**COORDINATION AND WORKFLOW MANAGEMENT**

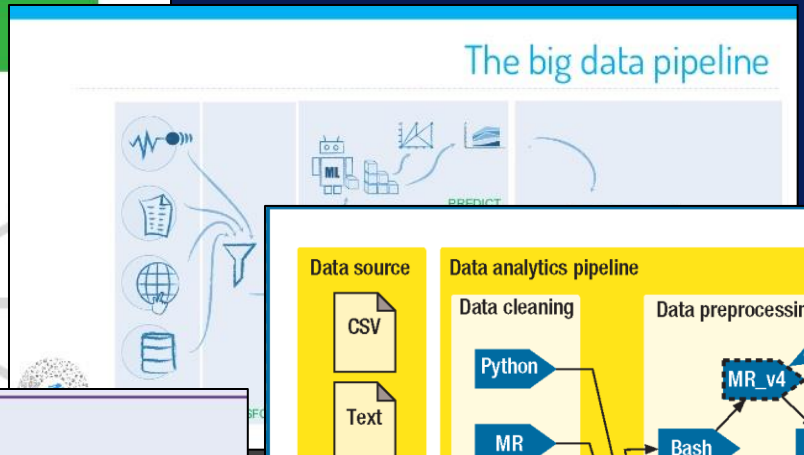
**DATA INTEGRATION AND PROCESSING**

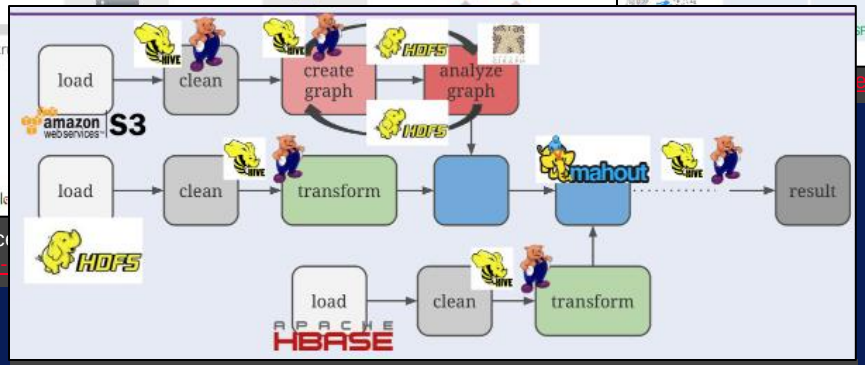
**DATA MANAGEMENT AND STORAGE**

# Example Big Data Processing Pipelines

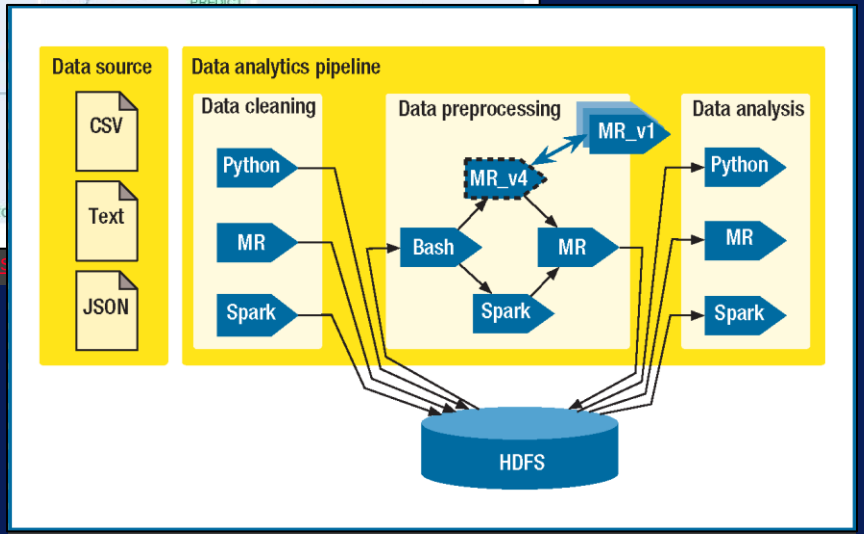



Source: cloud-



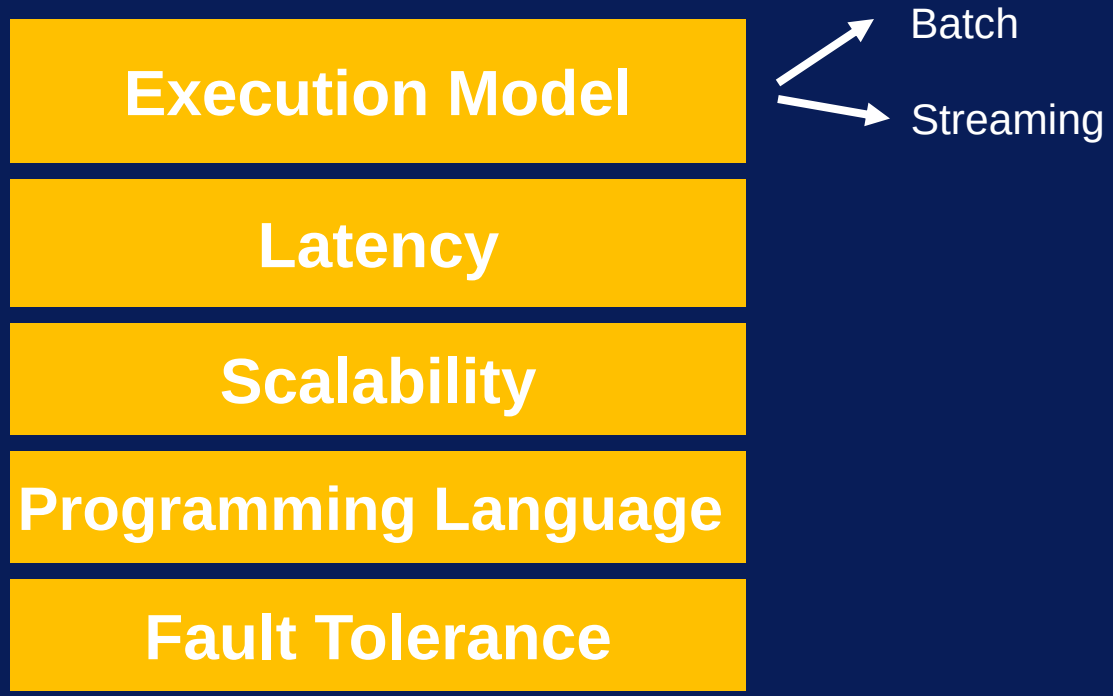




Source: https://www.mapr.com/blog/distributed-stream-and-graph-processing-apache-flink




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

# Categorization of Big Data Processing Systems



# Big Data Processing Systems






# MapReduce


| | |
|---|---|
| **Execution Model** | Batch processing using disk storage |
| **Latency** | High-latency |
| **Scalability** | |
| **Programming Language** | Java |
| **Fault Tolerance** | Replication |

# Spark


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

# Flink


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

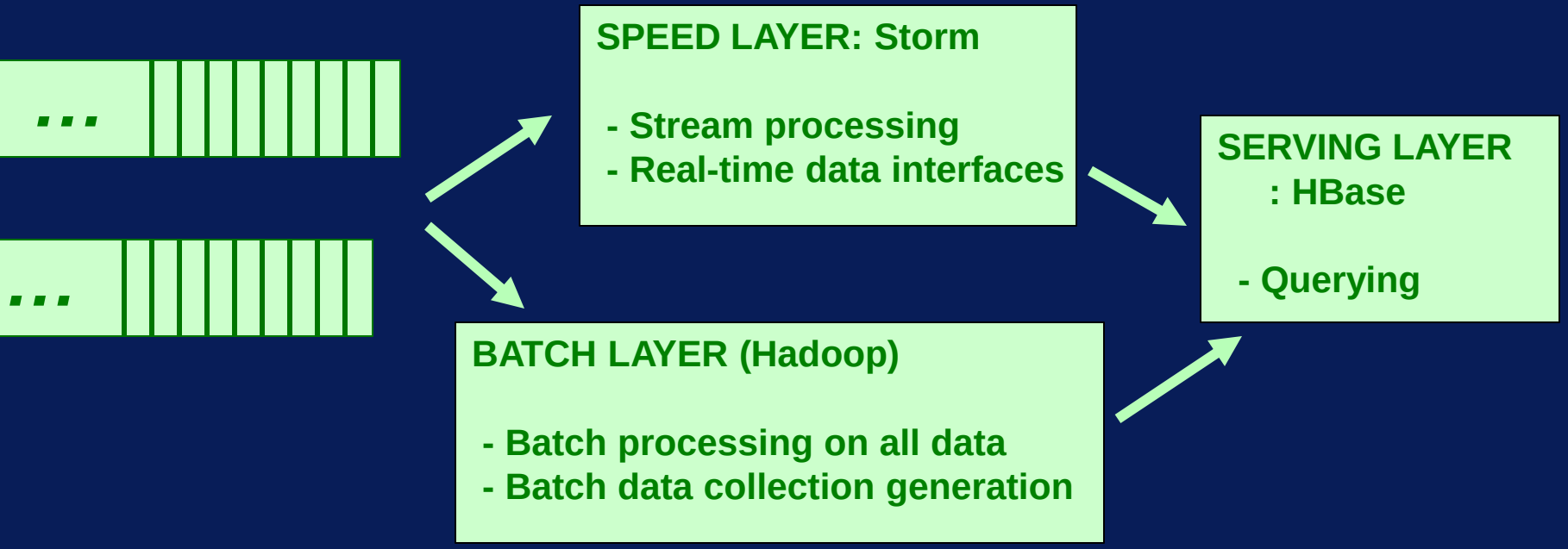
# Beam


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

# Storm


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

# Lambda Architecture:
## A Hybrid Data Processing Architecture



# Lambda Architecture:
## A Hybrid Data Processing Architecture