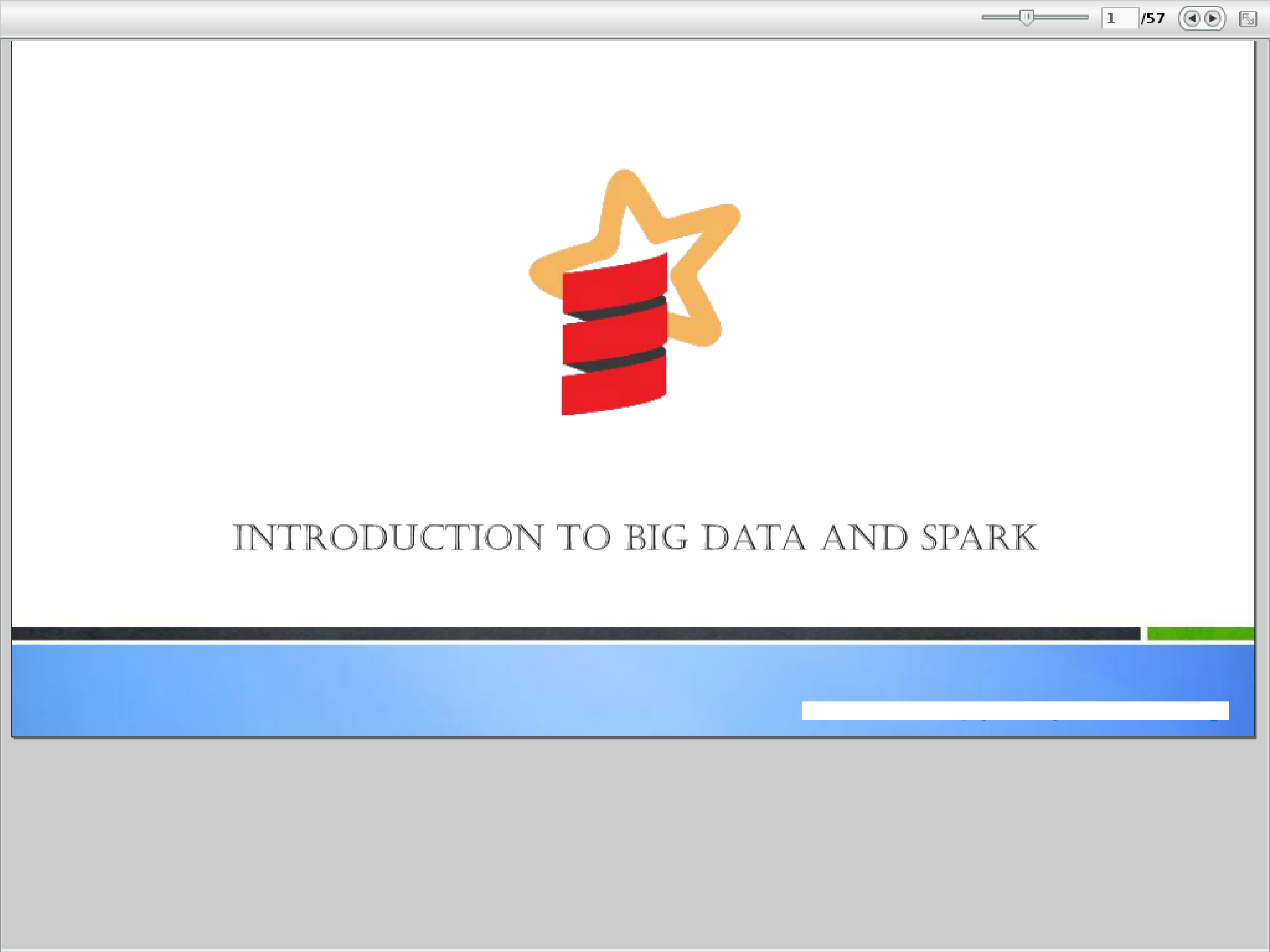
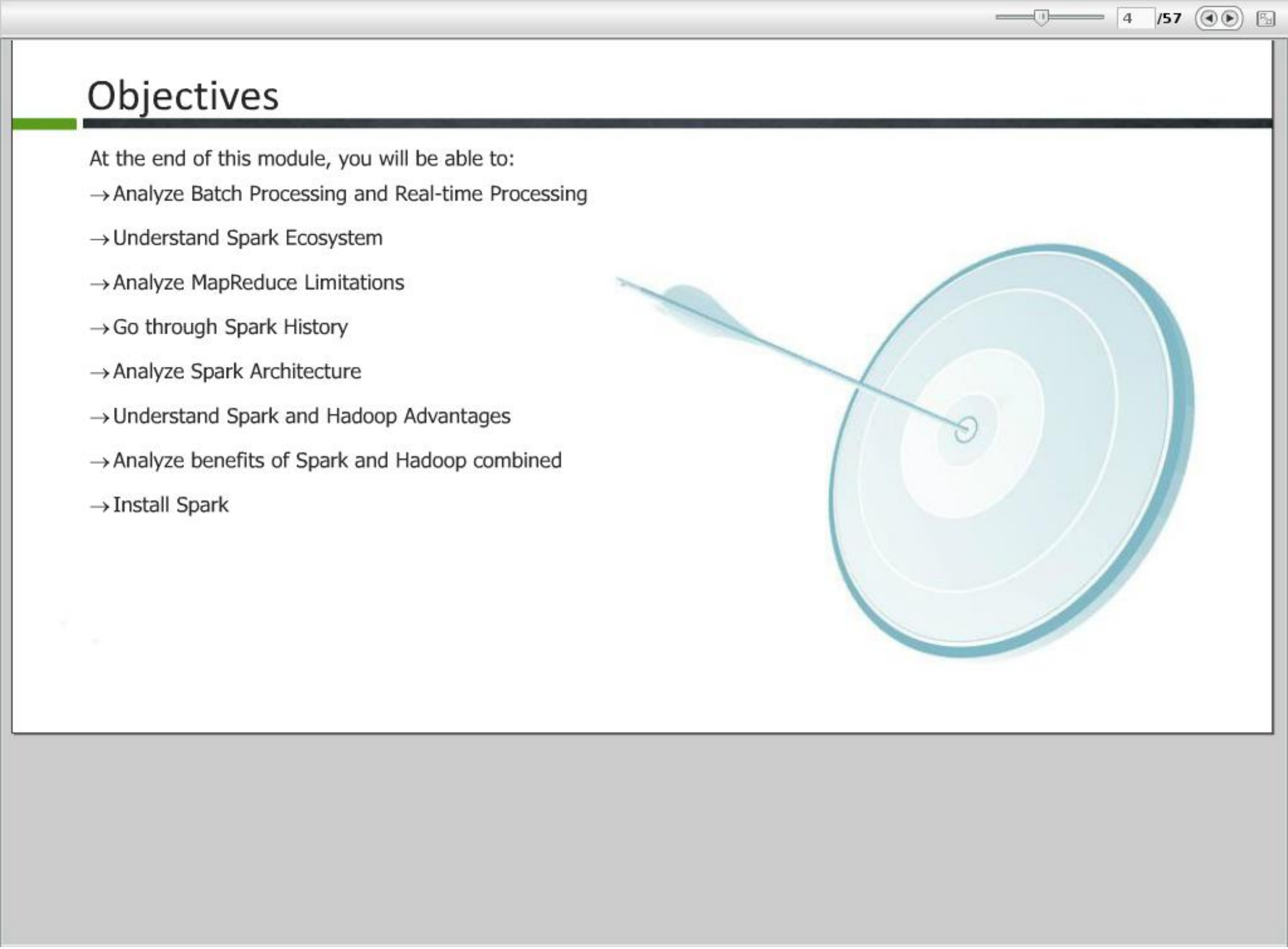


# INTRODUCTION TO BIG DATA AND SPARK



# Objectives

At the end of this module, you will be able to:

→ Analyze Batch Processing and Real-time Processing

→ Understand Spark Ecosystem

→ Analyze MapReduce Limitations

→ Go through Spark History

→ Analyze Spark Architecture

→ Understand Spark and Hadoop Advantages

→ Analyze benefits of Spark and Hadoop combined

→ Install Spark

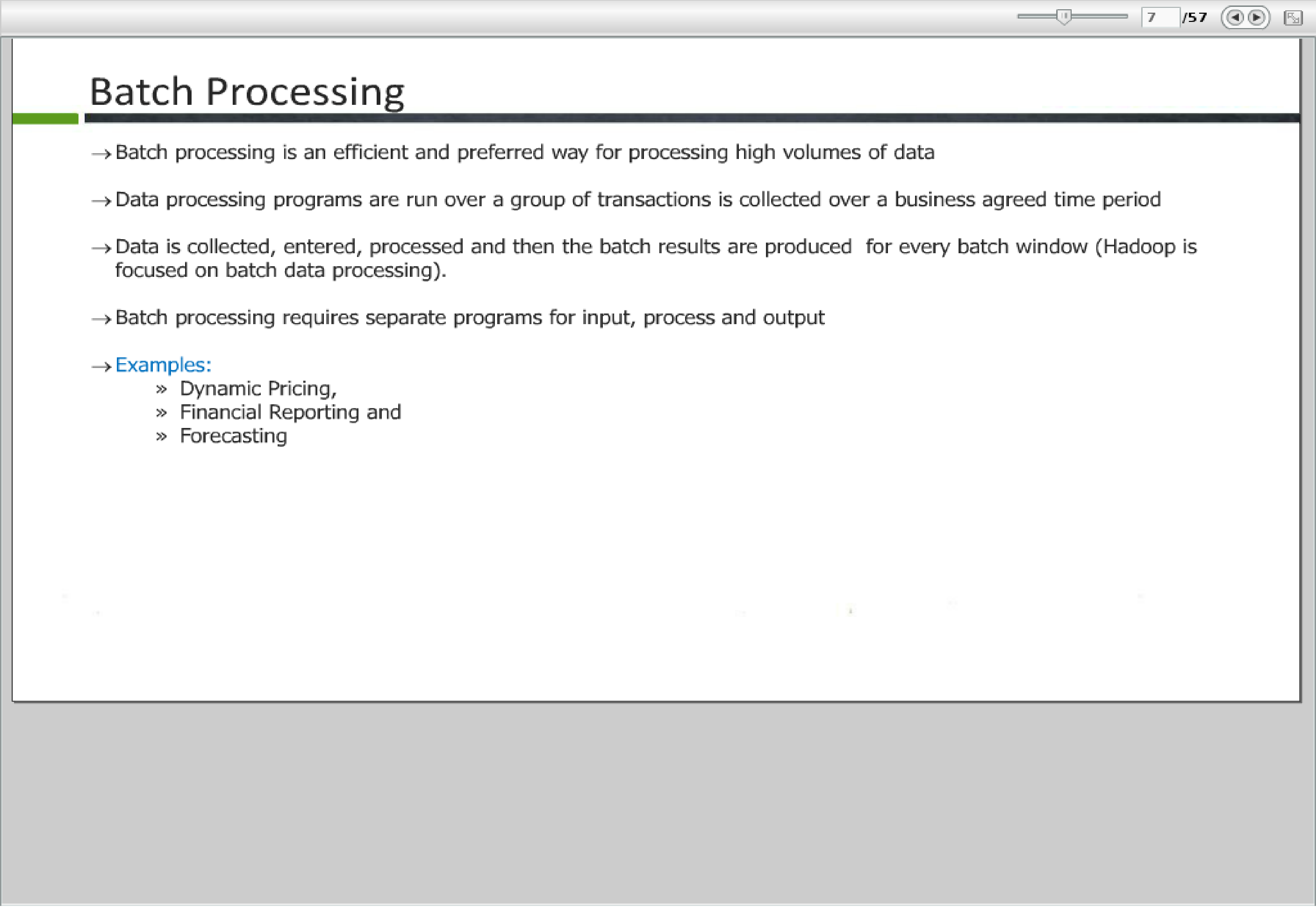
# Big Data and Associated Challenges

→ NYSE broadcasts several levels of data, including trade prices, sizes

→ NYSE Technologies receives four to five terabytes of a data in a day and uses it to do complex analytics, market surveillance, capacity planning and monitoring

NYSE generates about one terabyte of new trade data per day to perform stock trading analytics to determine trends for optimal trades

Refer: http://www.forbes.com/sites/tomgroenfeldt/2013/02/14/at-nyse-the-data-deluge-overwhelms-traditional-databases/



# Batch Processing

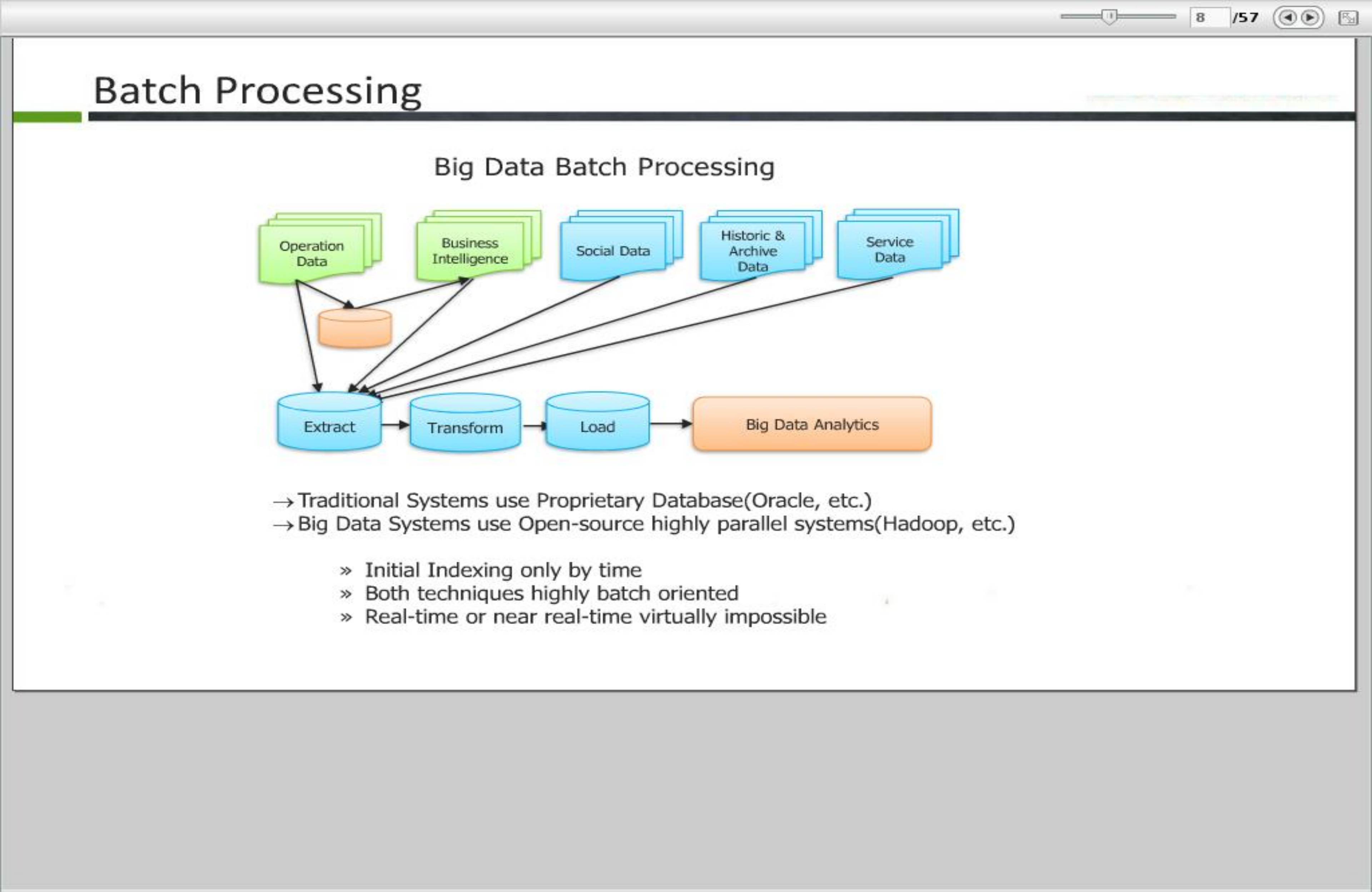
→ Batch processing is an efficient and preferred way for processing high volumes of data

→ Data processing programs are run over a group of transactions is collected over a business agreed time period

→ Data is collected, entered, processed and then the batch results are produced  for every batch window (Hadoop is focused on batch data processing).

→ Batch processing requires separate programs for input, process and output

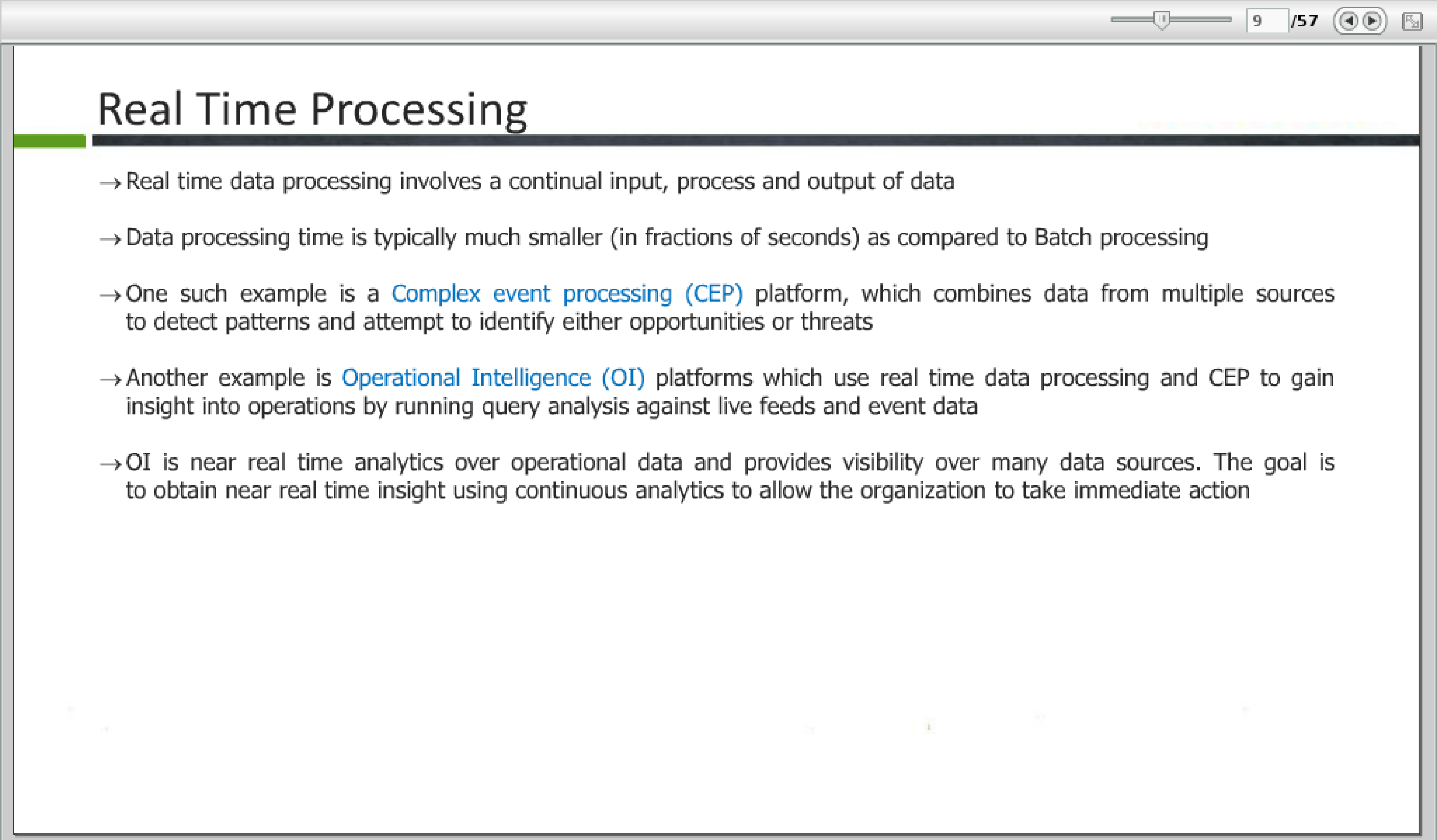
→ Examples:
- » Dynamic Pricing,
- » Financial Reporting and
- » Forecasting



# Batch Processing

→Traditional Systems use Proprietary Database(Oracle, etc.)
→Big Data Systems use Open-source highly parallel systems(Hadoop, etc.)

- » Initial Indexing only by time
- » Both techniques highly batch oriented
- » Real-time or near real-time virtually impossible

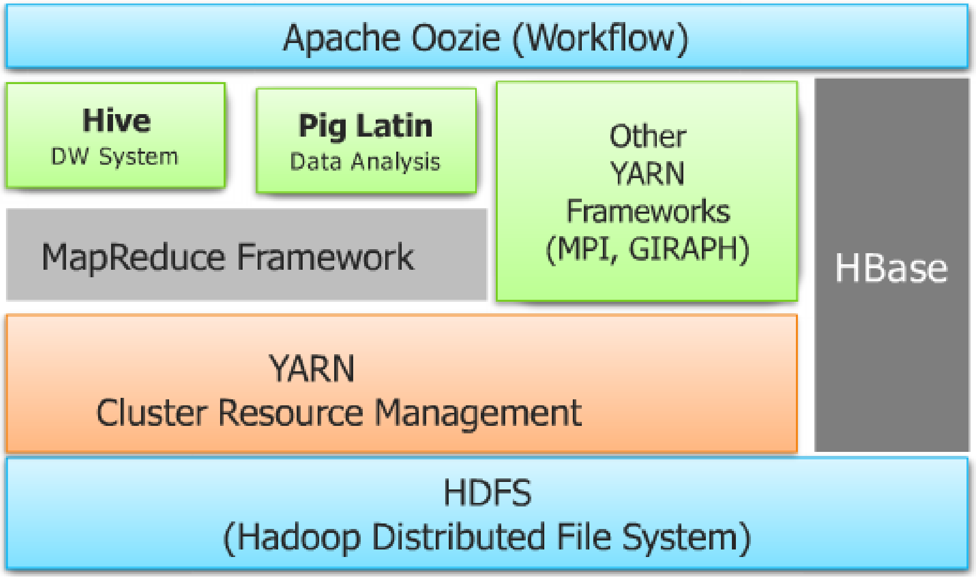


# Real Time Processing

- → Real time data processing involves a continual input, process and output of data
- → Data processing time is typically much smaller (in fractions of seconds) as compared to Batch processing
- → One such example is a Complex event processing (CEP) platform, which combines data from multiple sources to detect patterns and attempt to identify either opportunities or threats
- → Another example is Operational Intelligence (OI) platforms which use real time data processing and CEP to gain insight into operations by running query analysis against live feeds and event data
- → OI is near real time analytics over operational data and provides visibility over many data sources. The goal is to obtain near real time insight using continuous analytics to allow the organization to take immediate action

# Hadoop for Batch Processing

Hadoop 2.0



Apache Oozie (Workflow)

**Hive**
DW System

**Pig Latin**
Data Analysis

Other
YARN
Frameworks
(MPI, GIRAPH)

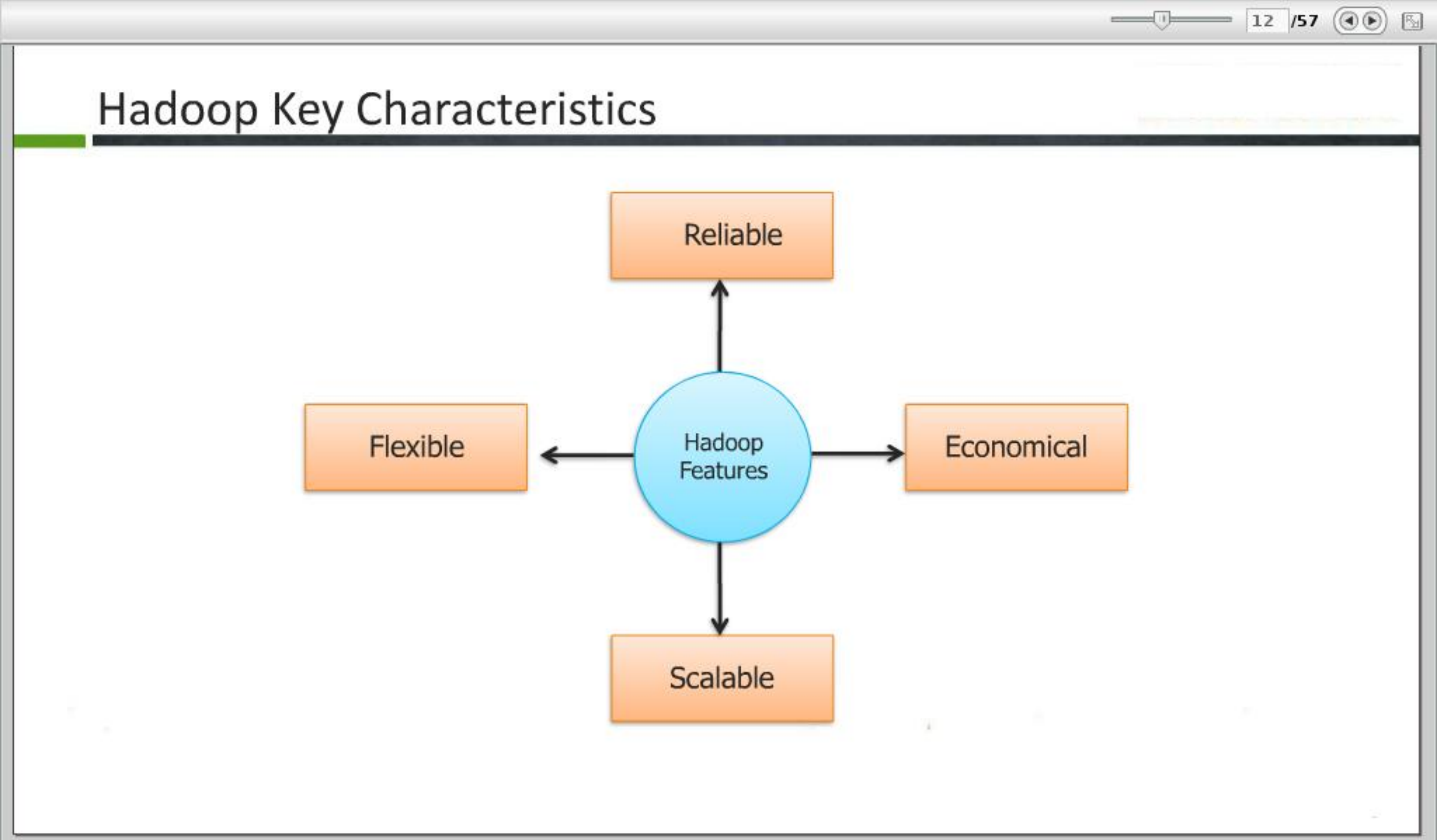
MapReduce Framework

HBase

YARN
Cluster Resource Management

HDFS
(Hadoop Distributed File System)

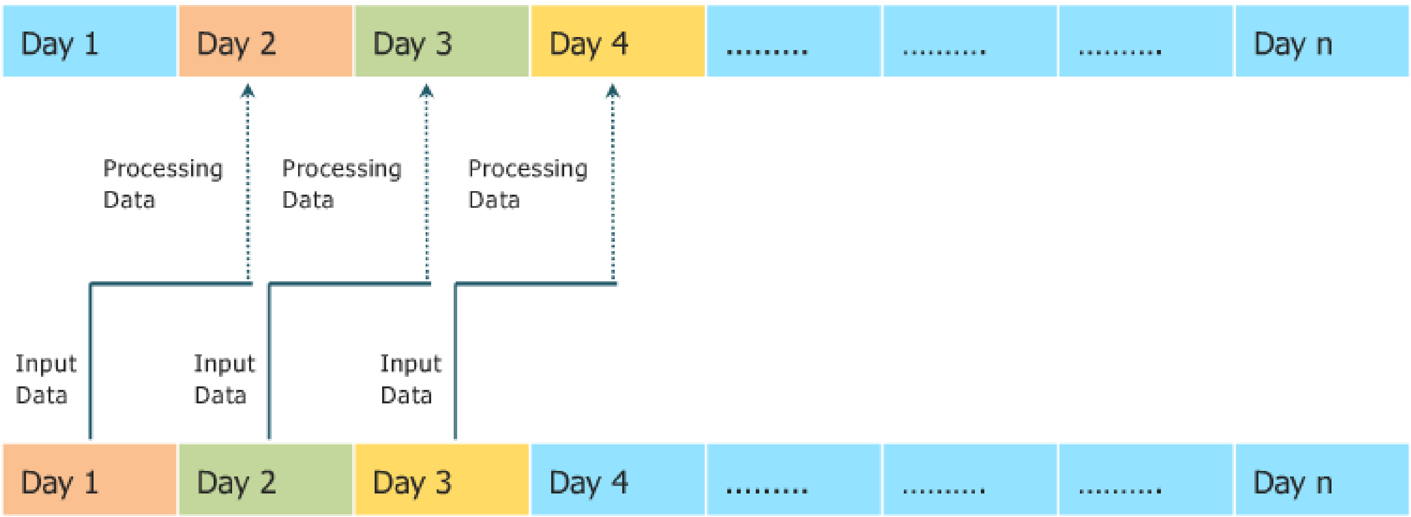
YARN adds a more general interface to run non-MapReduce jobs (such as Graph Processing) within the Hadoop framework



## Hadoop Key Characteristics

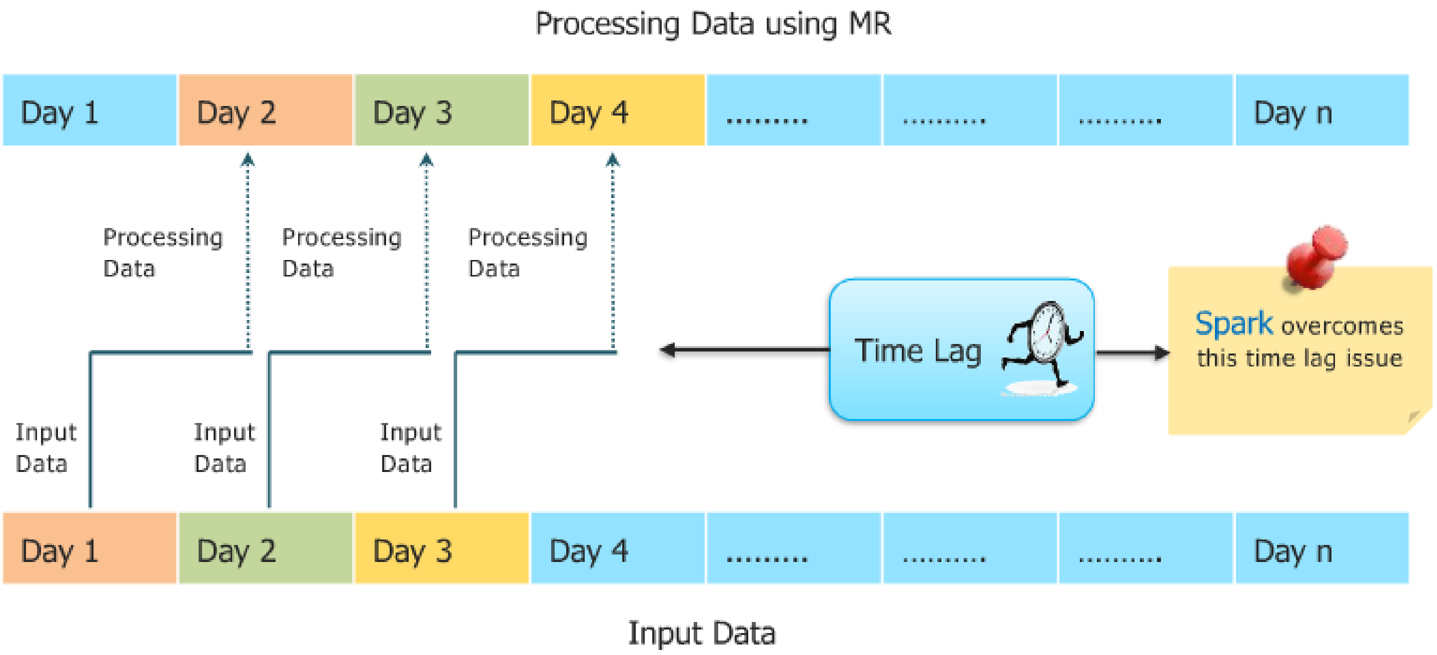
# Data Processing in Hadoop

Processing Data using MR



Input Data

# Data Processing in Hadoop



Processing Data using MR

Day 1 | Day 2 | Day 3 | Day 4 | ......... | ......... | ......... | Day n

Processing Data

Processing Data

Processing Data

Input Data

Input Data

Input Data

Time Lag

Spark overcomes this time lag issue

Day 1 | Day 2 | Day 3 | Day 4 | ......... | ......... | ......... | Day n

Input Data

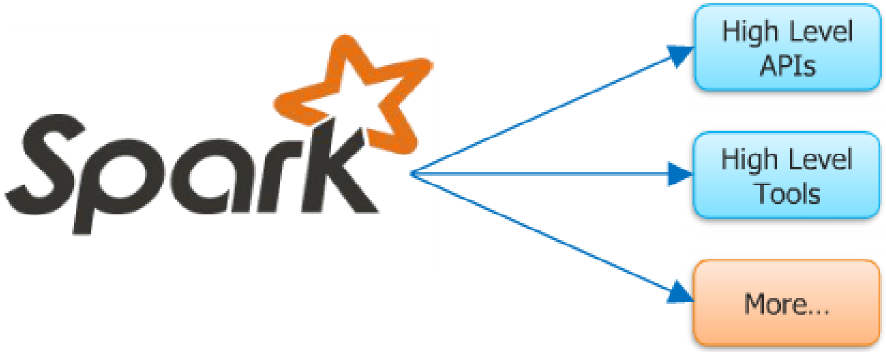
# What is Spark?

→Apache Spark is a general-purpose cluster in-memory computing system

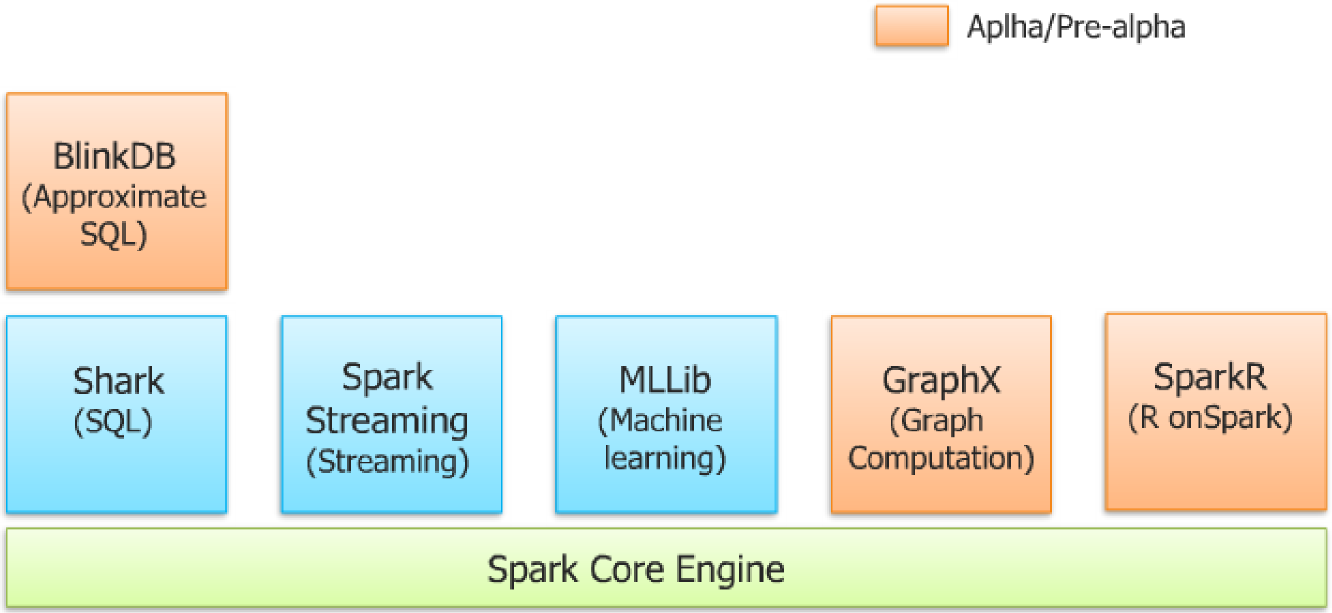
→It is used for fast data analytics

→It abstracts APIs in Java, Scala and Python, and provides an optimized engine that supports general execution graphs

→Provides various high level tools like Spark SQL for structured data processing, Mlib for Machine Learning and more



# Spark Ecosystem



🟧 Aplha/Pre-alpha

BlinkDB (Approximate SQL)

Shark (SQL)

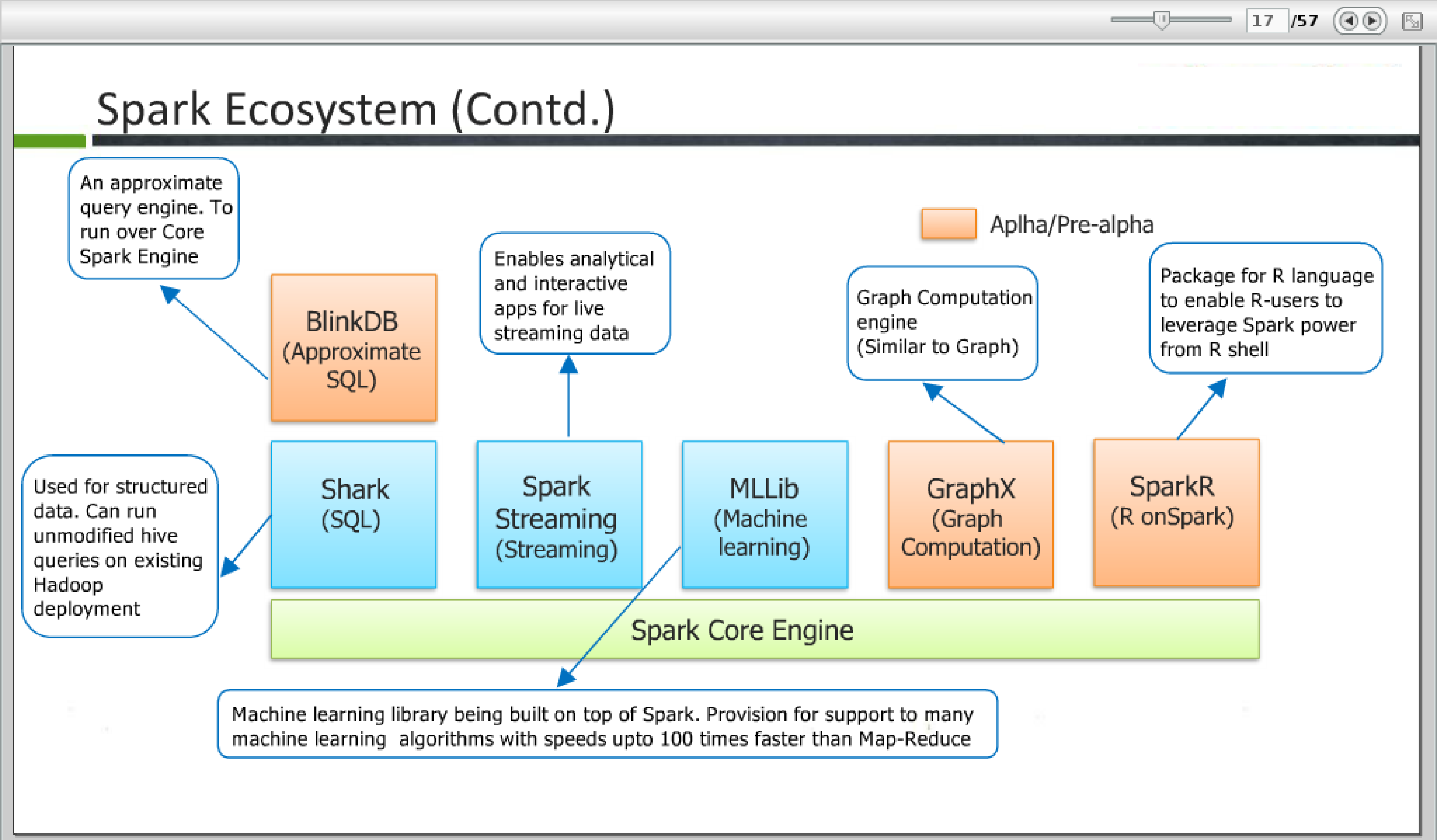
Spark Streaming (Streaming)

MLLib (Machine learning)

GraphX (Graph Computation)

SparkR (R onSpark)

Spark Core Engine



# Spark Ecosystem (Contd.)

🟧 Aplha/Pre-alpha

An approximate query engine. To run over Core Spark Engine

BlinkDB (Approximate SQL)

Enables analytical and interactive apps for live streaming data

Graph Computation engine (Similar to Graph)

Package for R language to enable R-users to leverage Spark power from R shell

Used for structured data. Can run unmodified hive queries on existing Hadoop deployment

Shark (SQL)

Spark Streaming (Streaming)

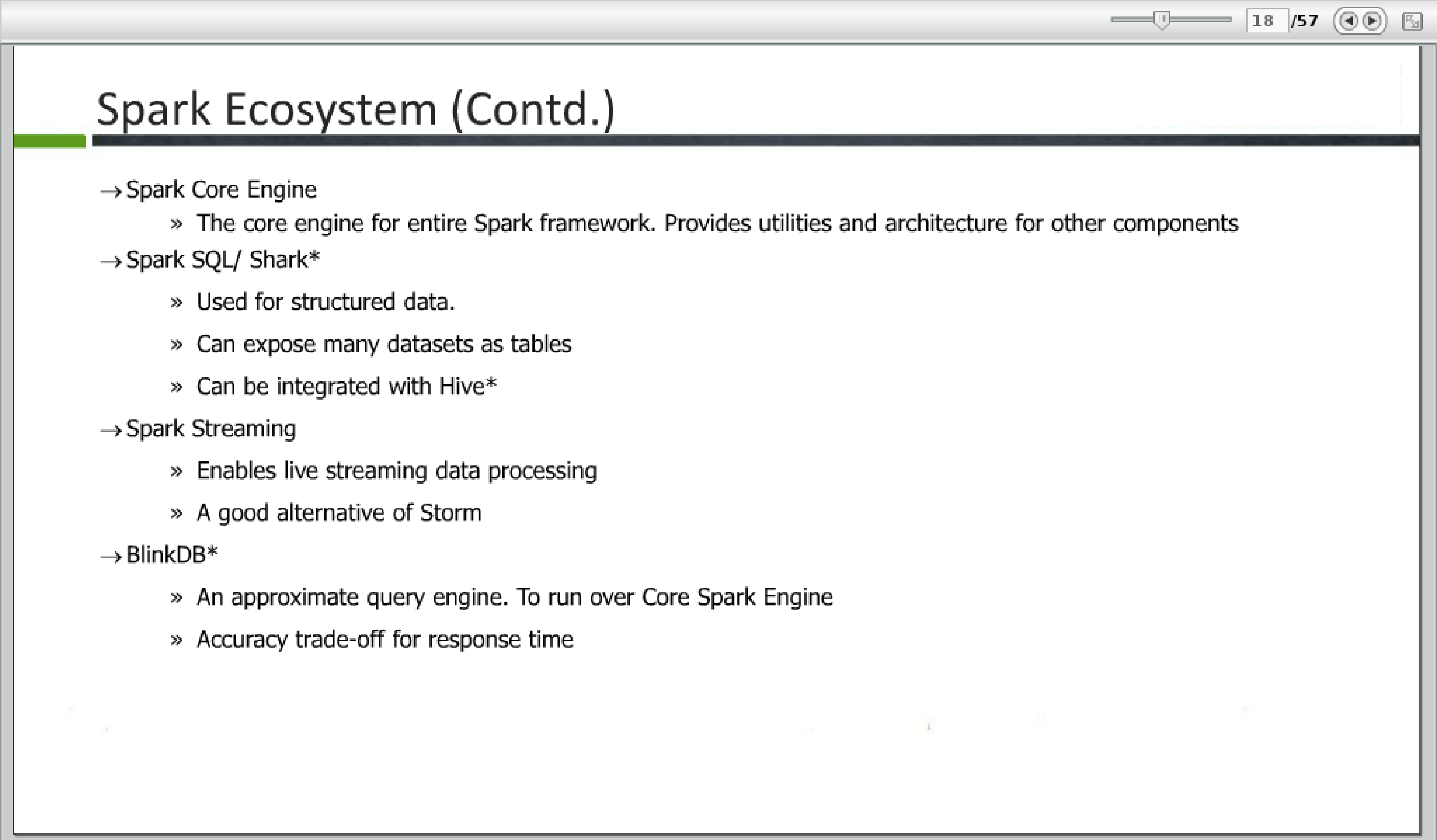
MLLib (Machine learning)

GraphX (Graph Computation)

SparkR (R onSpark)

Spark Core Engine

Machine learning library being built on top of Spark. Provision for support to many machine learning algorithms with speeds upto 100 times faster than Map-Reduce



# Spark Ecosystem (Contd.)

→Spark Core Engine

- » The core engine for entire Spark framework. Provides utilities and architecture for other components

→Spark SQL/ Shark*

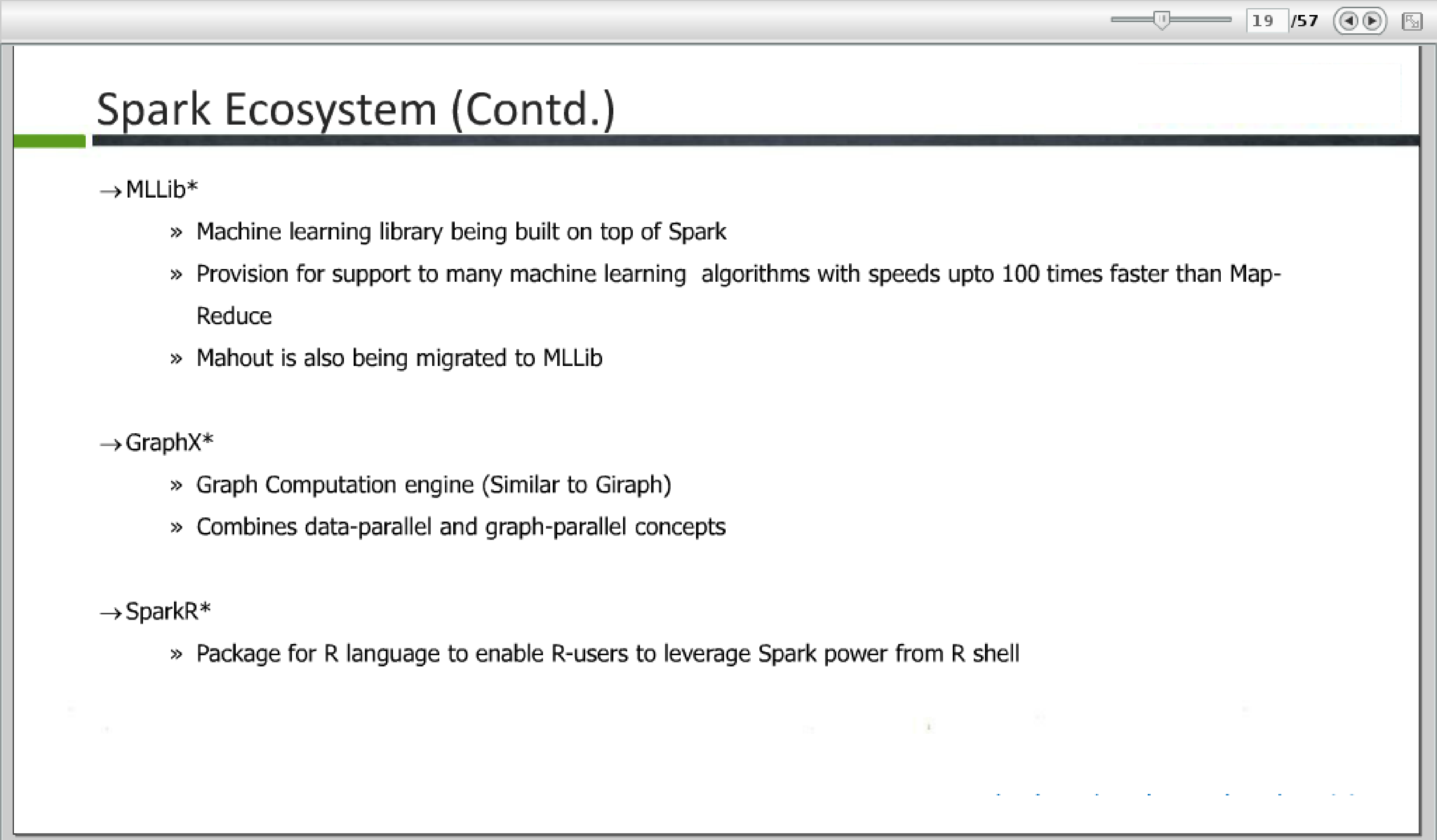
- » Used for structured data.
- » Can expose many datasets as tables
- » Can be integrated with Hive*

→Spark Streaming

- » Enables live streaming data processing
- » A good alternative of Storm

→BlinkDB*

- » An approximate query engine. To run over Core Spark Engine
- » Accuracy trade-off for response time



## Spark Ecosystem (Contd.)

→MLLib*

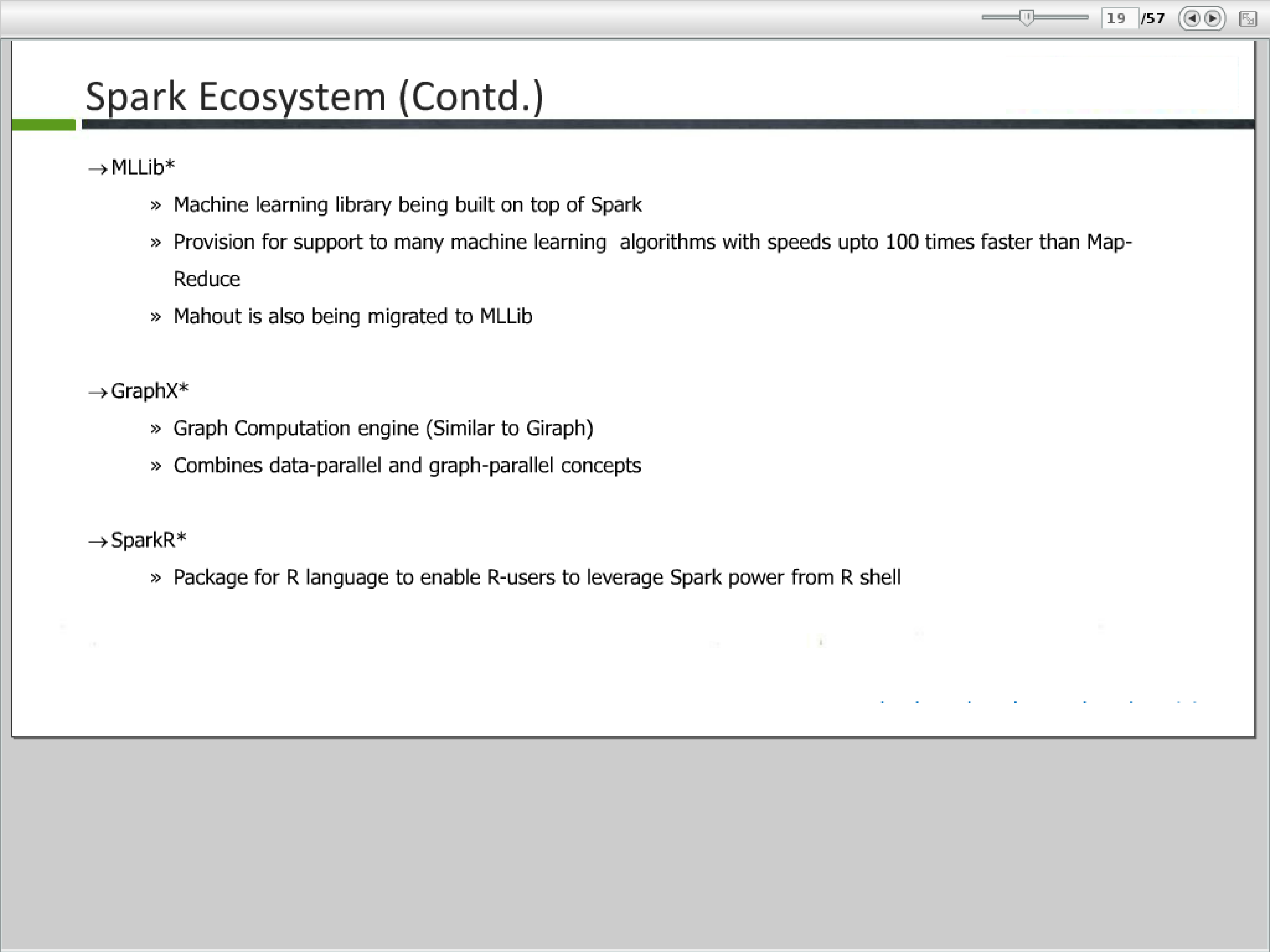
- » Machine learning library being built on top of Spark
- » Provision for support to many machine learning  algorithms with speeds upto 100 times faster than Map-Reduce
- » Mahout is also being migrated to MLLib

→GraphX*

- » Graph Computation engine (Similar to Giraph)
- » Combines data-parallel and graph-parallel concepts

→SparkR*

- » Package for R language to enable R-users to leverage Spark power from R shell



# Spark Ecosystem (Contd.)

→ MLLib*

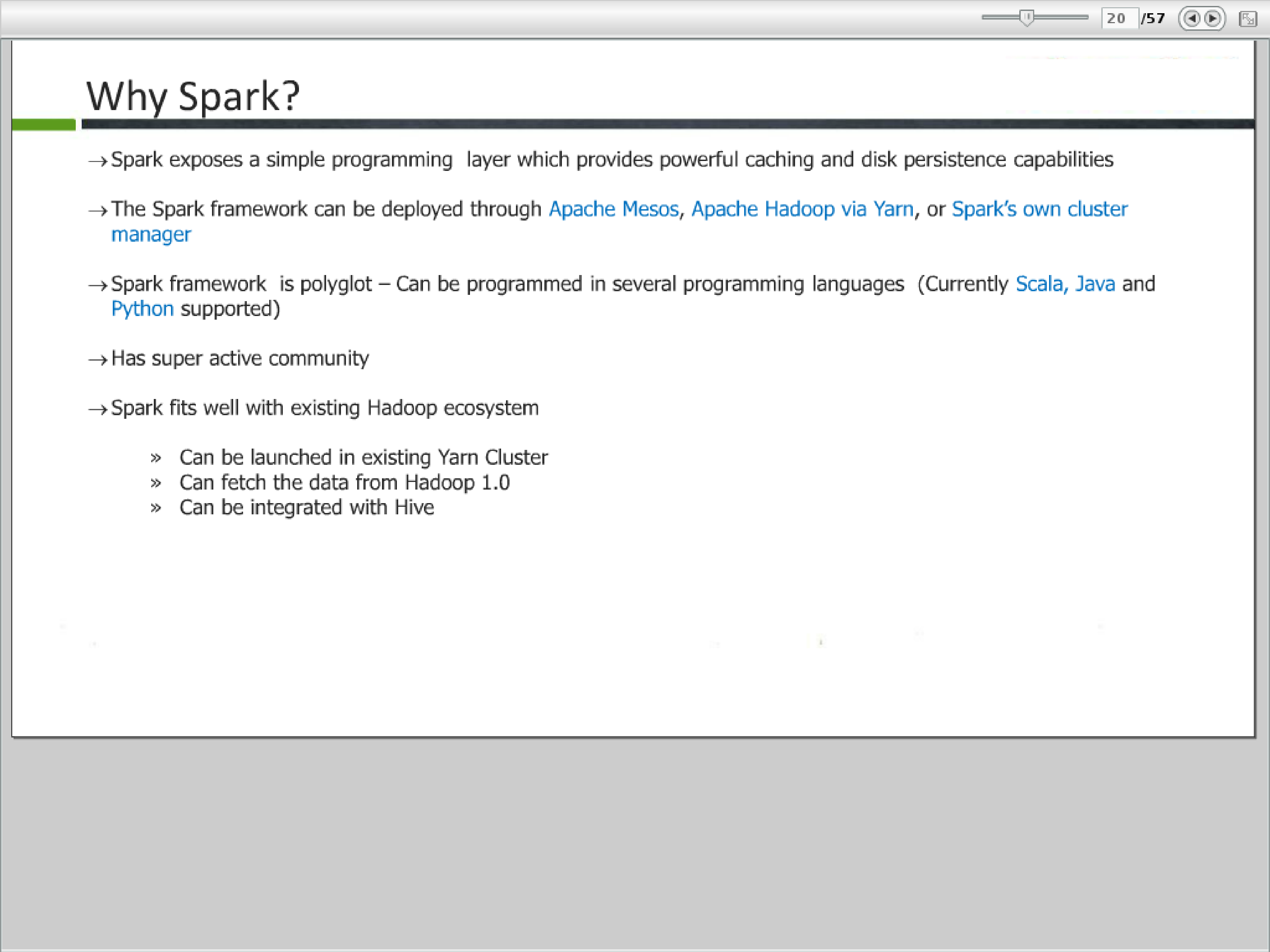
- » Machine learning library being built on top of Spark
- » Provision for support to many machine learning algorithms with speeds upto 100 times faster than Map-Reduce
- » Mahout is also being migrated to MLLib

→ GraphX*

- » Graph Computation engine (Similar to Giraph)
- » Combines data-parallel and graph-parallel concepts

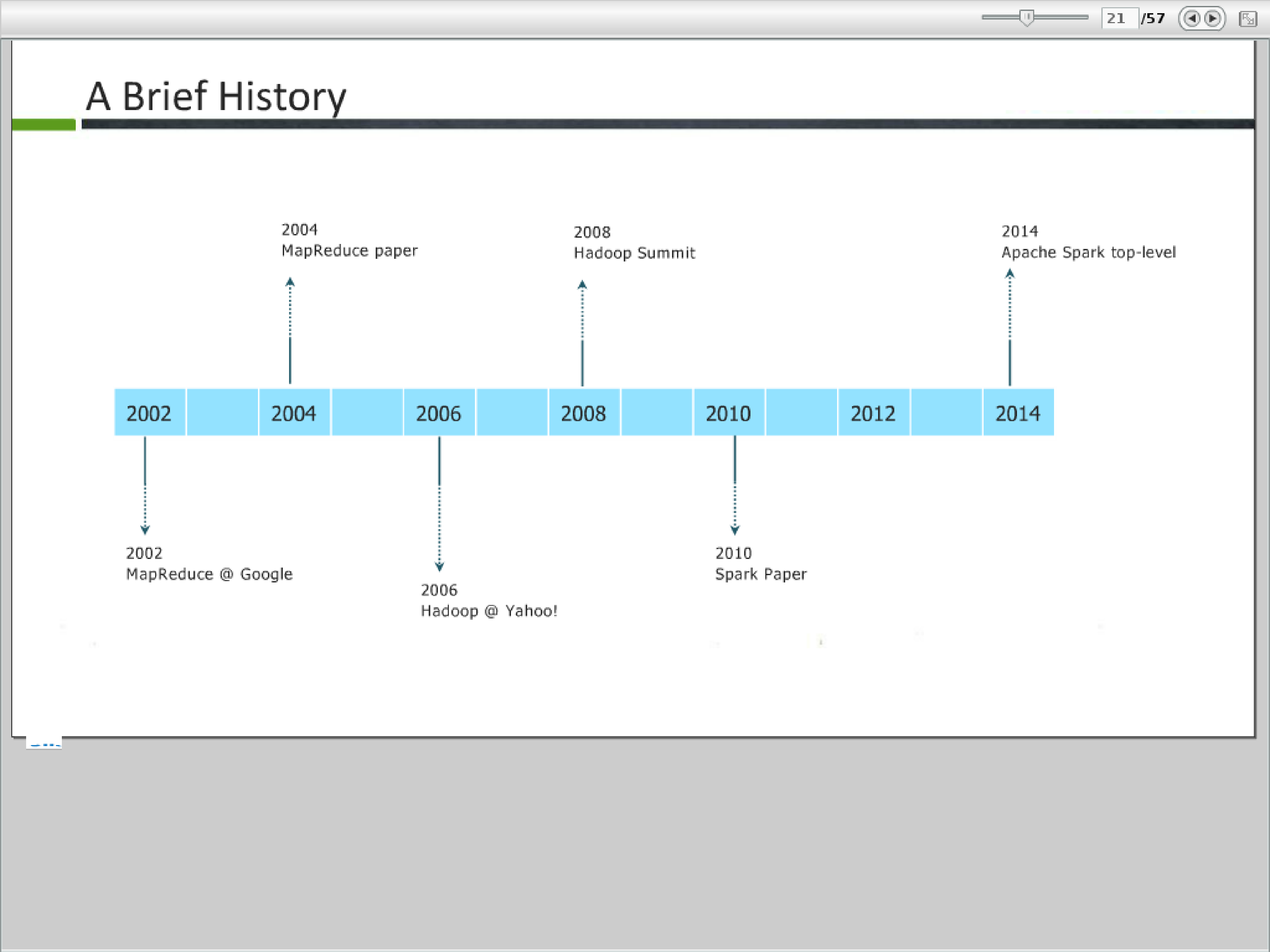
→ SparkR*

- » Package for R language to enable R-users to leverage Spark power from R shell



# Why Spark?

- Spark exposes a simple programming layer which provides powerful caching and disk persistence capabilities
- The Spark framework can be deployed through Apache Mesos, Apache Hadoop via Yarn, or Spark's own cluster manager
- Spark framework is polyglot – Can be programmed in several programming languages (Currently Scala, Java and Python supported)
- Has super active community
- Spark fits well with existing Hadoop ecosystem
  - Can be launched in existing Yarn Cluster
  - Can fetch the data from Hadoop 1.0
  - Can be integrated with Hive



# A Brief History

2002 | 2004 | 2006 | 2008 | 2010 | 2012 | 2014

2002
MapReduce @ Google

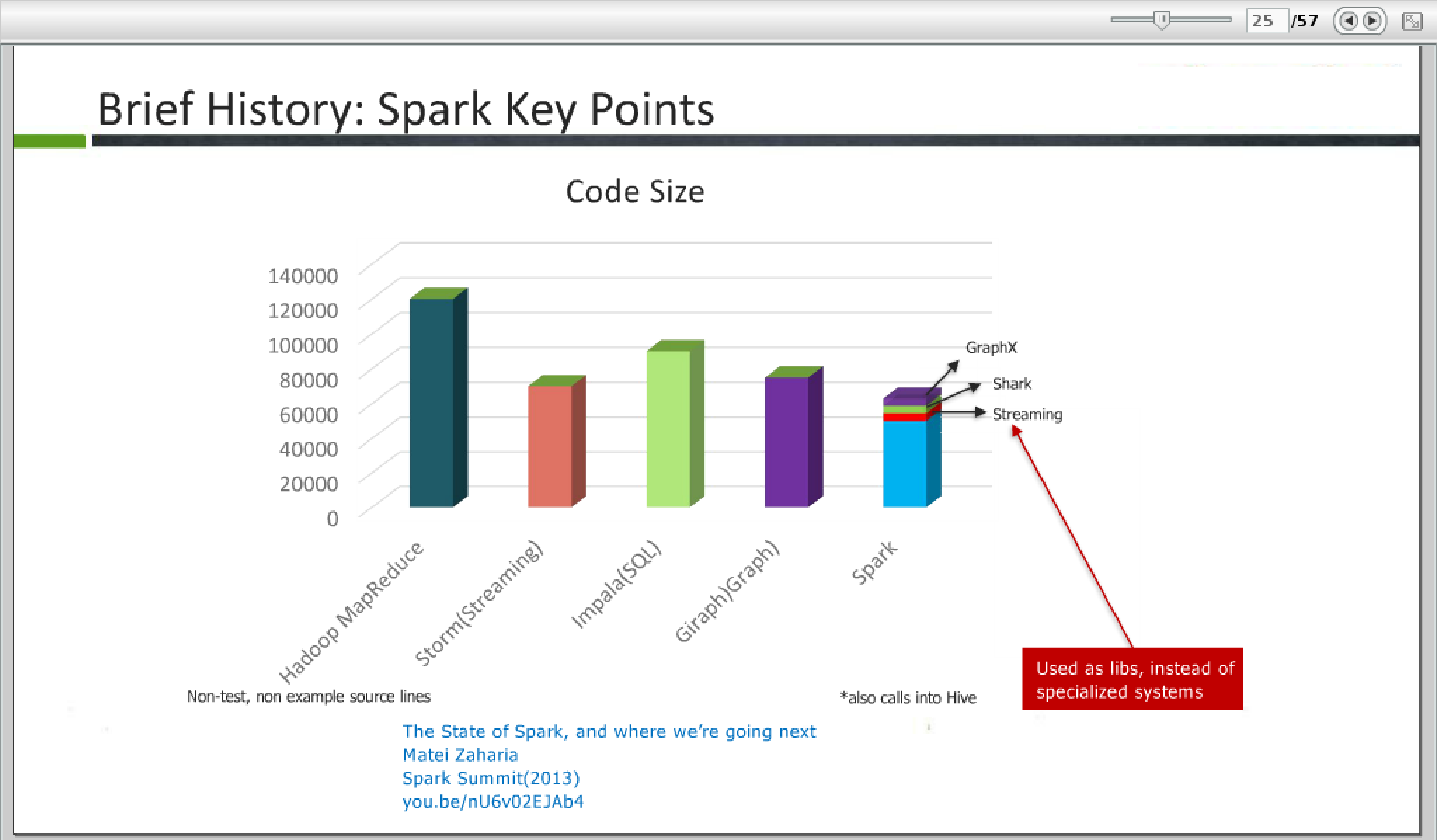
2004
MapReduce paper

2006
Hadoop @ Yahoo!

2008
Hadoop Summit

2010
Spark Paper

2014
Apache Spark top-level



# Brief History: Spark Key Points

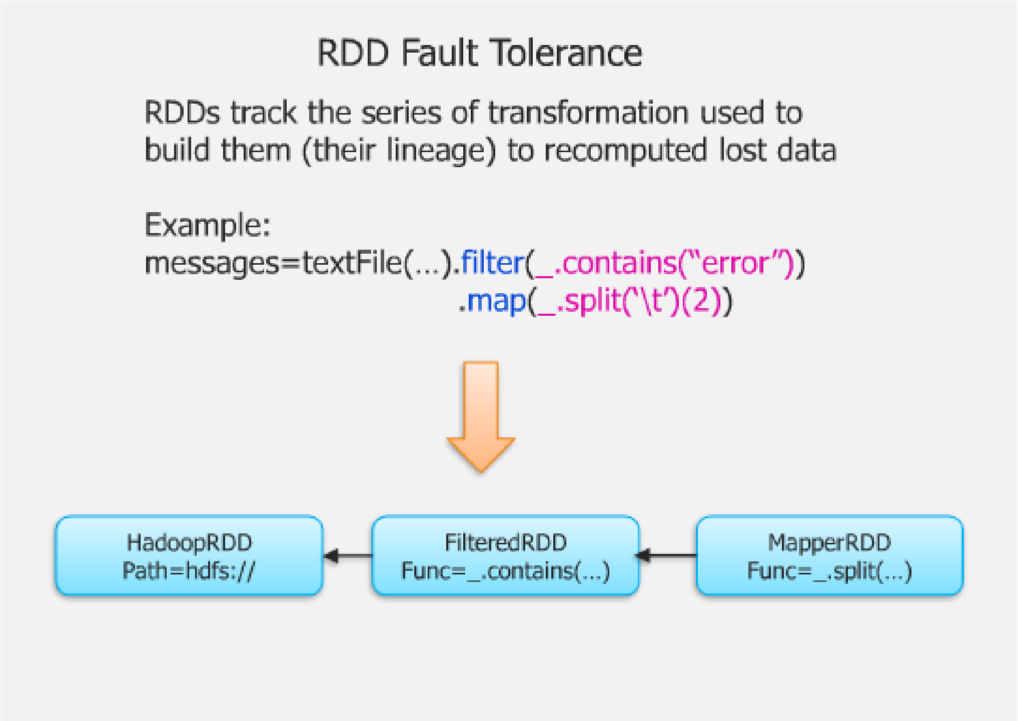
Non-test, non example source lines

*also calls into Hive

Used as libs, instead of specialized systems

The State of Spark, and where we're going next
Matei Zaharia
Spark Summit(2013)
you.be/nU6v02EJAb4

# Brief History: Spark Key Points



## RDD Fault Tolerance

RDDs track the series of transformation used to build them (their lineage) to recomputed lost data

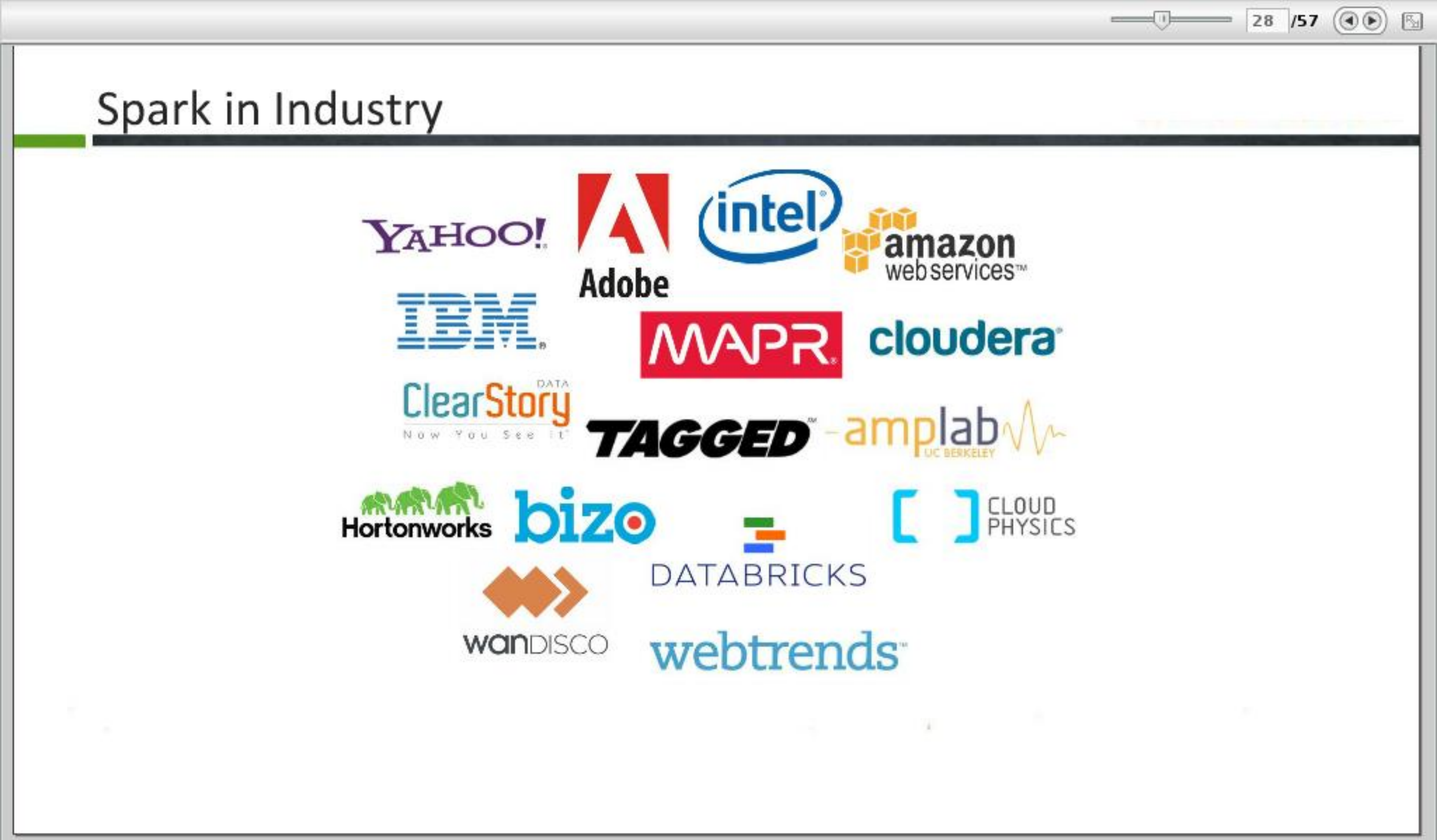
Example:

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

HadoopRDD
Path=hdfs://

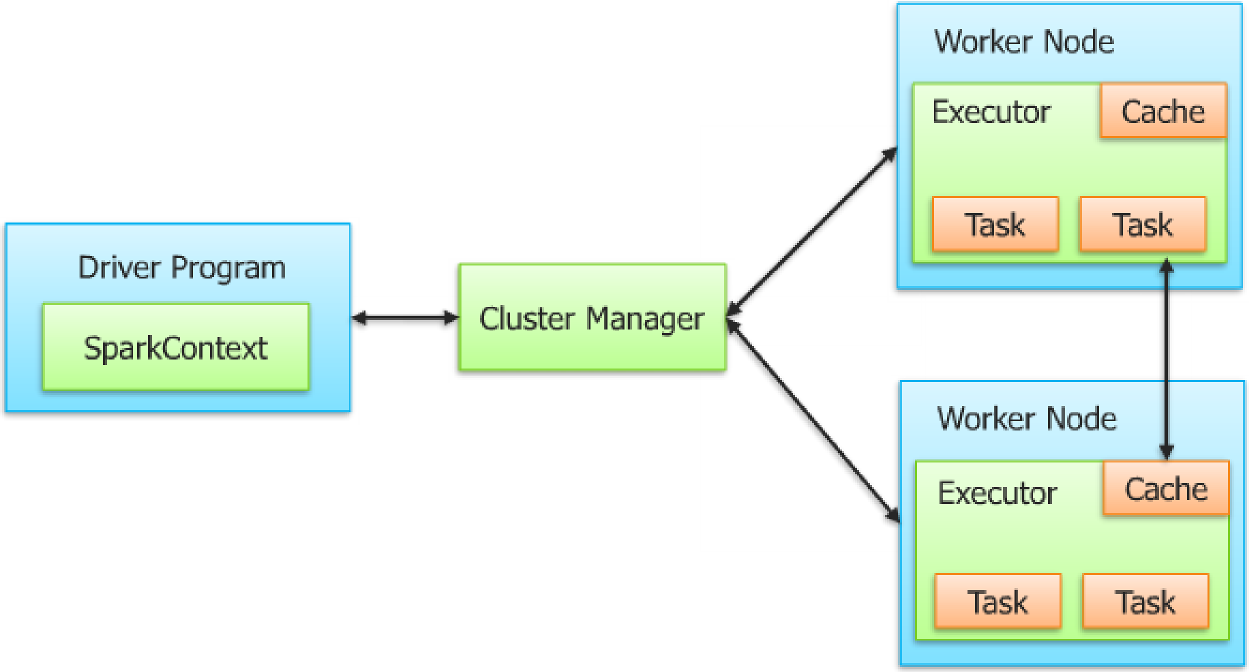
FilteredRDD
Func=_.contains(...)

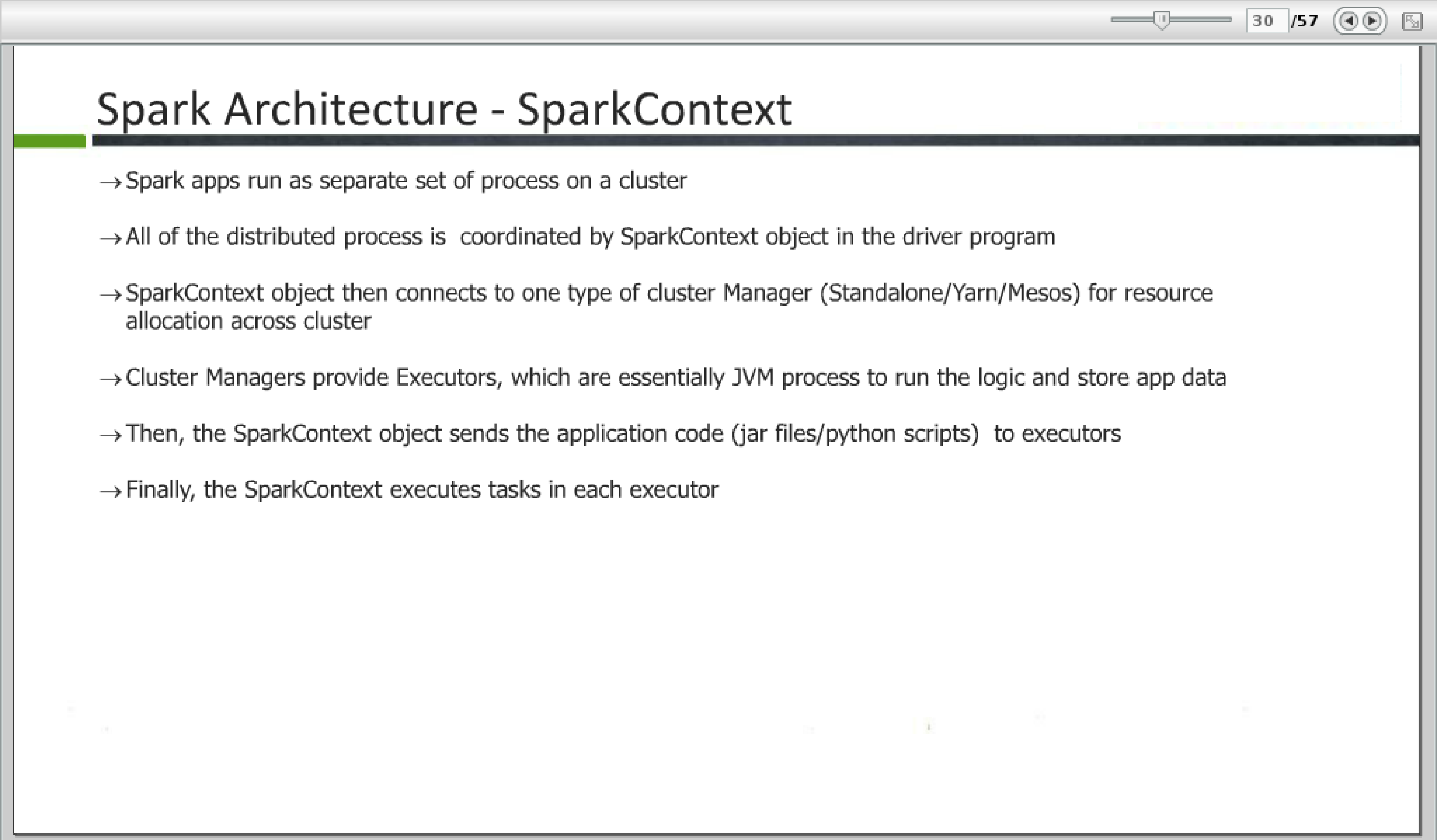
MapperRDD
Func=_.split(...)



## Spark in Industry

# Spark Architecture





# Spark Architecture - SparkContext

→Spark apps run as separate set of process on a cluster

→All of the distributed process is coordinated by SparkContext object in the driver program

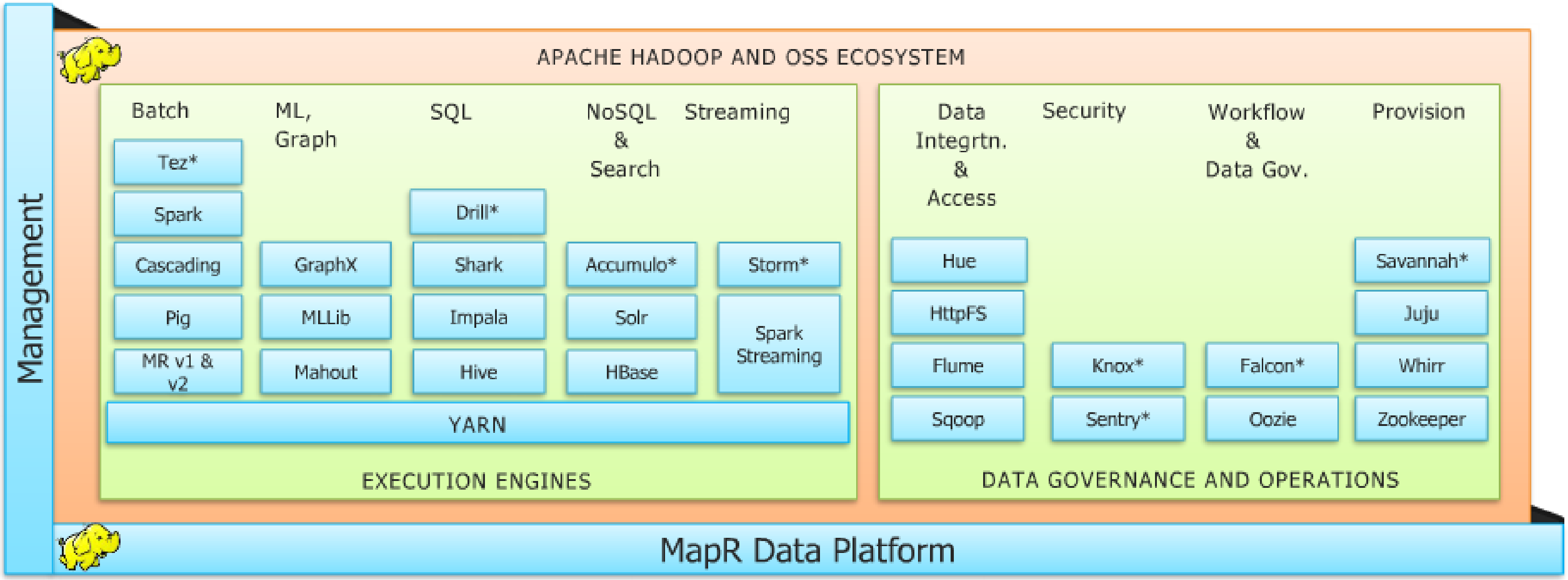
→SparkContext object then connects to one type of cluster Manager (Standalone/Yarn/Mesos) for resource allocation across cluster

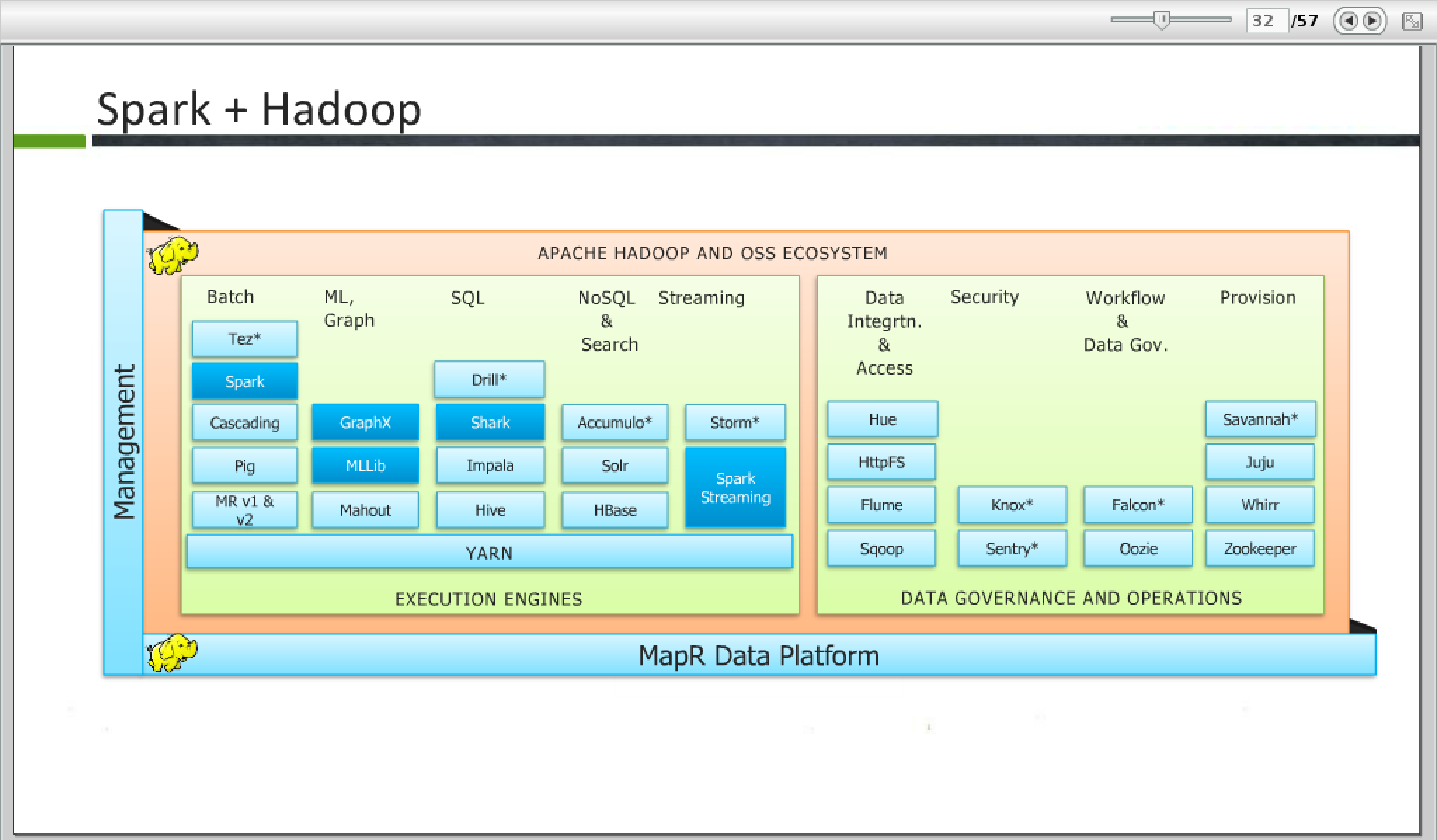
→Cluster Managers provide Executors, which are essentially JVM process to run the logic and store app data

→Then, the SparkContext object sends the application code (jar files/python scripts) to executors

→Finally, the SparkContext executes tasks in each executor

# Spark + Hadoop





# Spark + Hadoop

Management

APACHE HADOOP AND OSS ECOSYSTEM

Batch: Tez*, Spark, Cascading, Pig, MR v1 & v2

ML, Graph: GraphX, MLLib, Mahout

SQL: Drill*, Shark, Impala, Hive

NoSQL & Search: Accumulo*, Solr, HBase

Streaming: Storm*, Spark Streaming

YARN

EXECUTION ENGINES

Data Integrtn. & Access: Hue, HttpFS, Flume, Sqoop

Security: Knox*, Sentry*

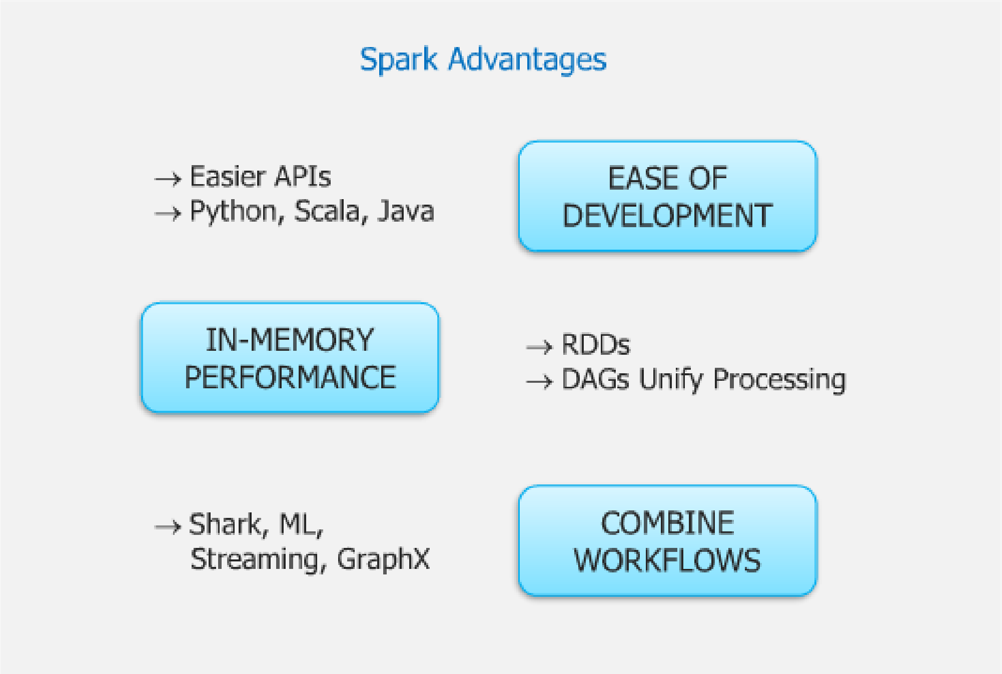
Workflow & Data Gov.: Falcon*, Oozie

Provision: Savannah*, Juju, Whirr, Zookeeper

DATA GOVERNANCE AND OPERATIONS

MapR Data Platform

# Spark Advantages



Spark Advantages

→ Easier APIs
→ Python, Scala, Java

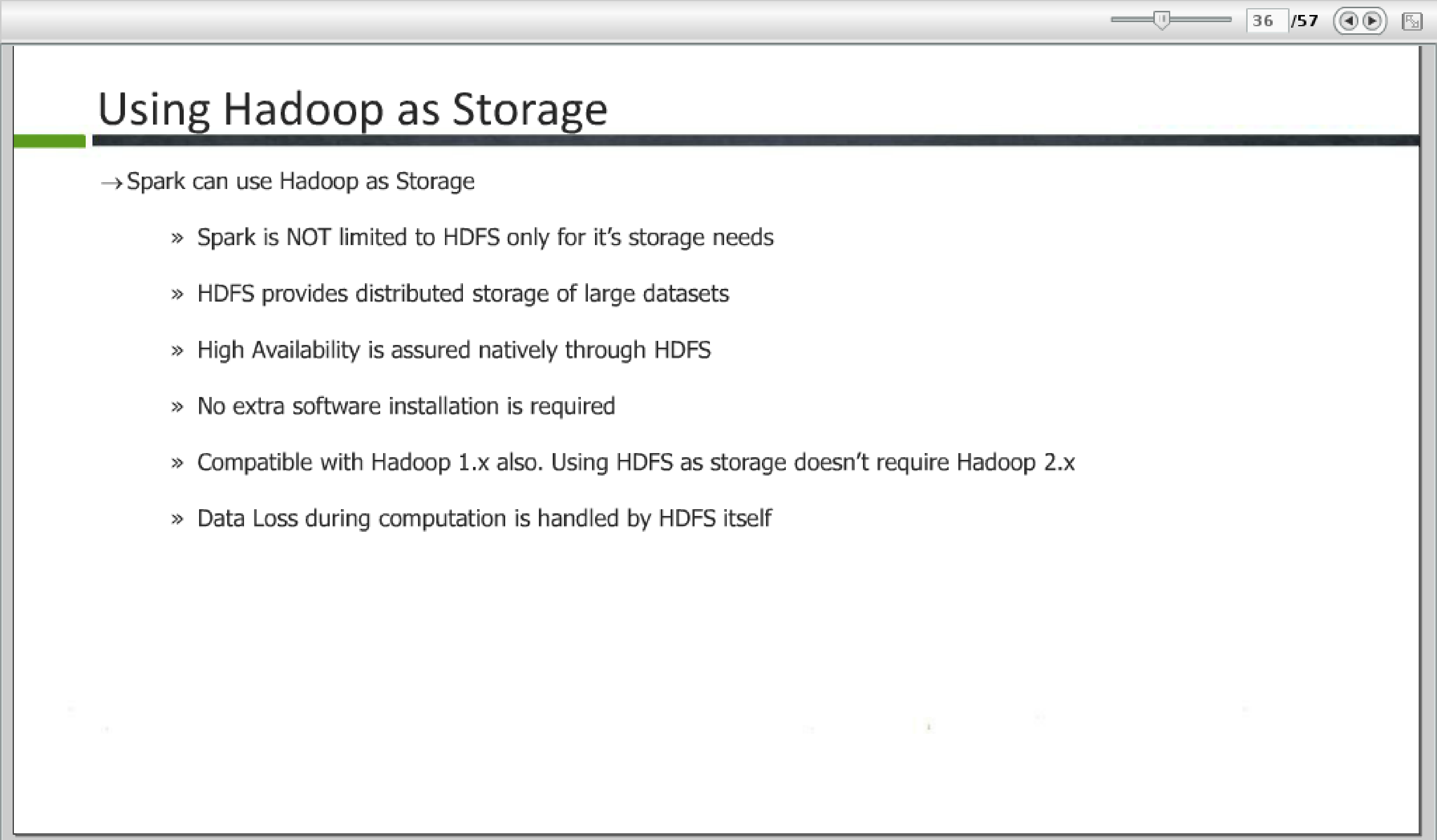
EASE OF DEVELOPMENT

IN-MEMORY PERFORMANCE

→ RDDs
→ DAGs Unify Processing

→ Shark, ML, Streaming, GraphX

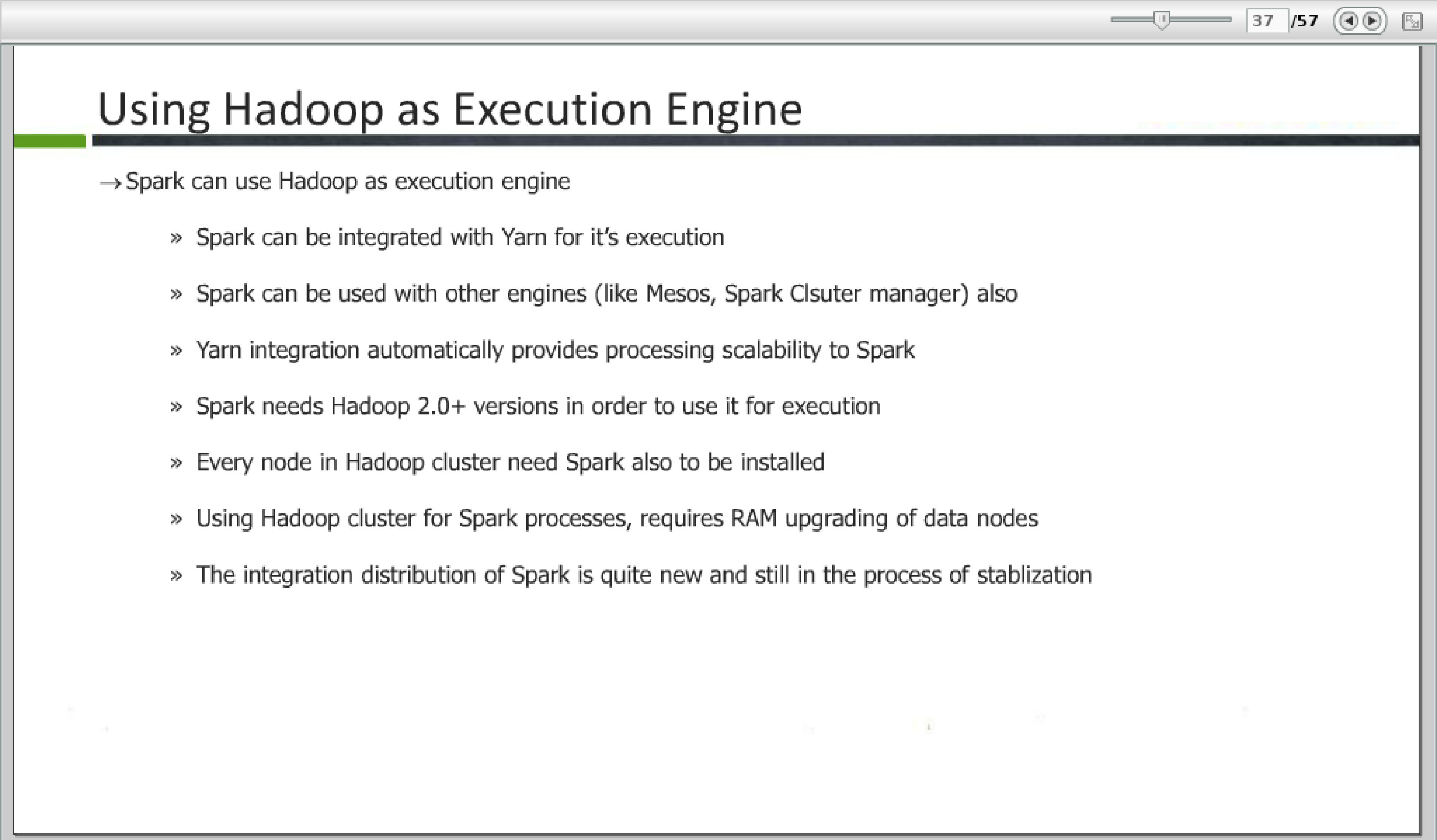
COMBINE WORKFLOWS



# Using Hadoop as Storage

→ Spark can use Hadoop as Storage

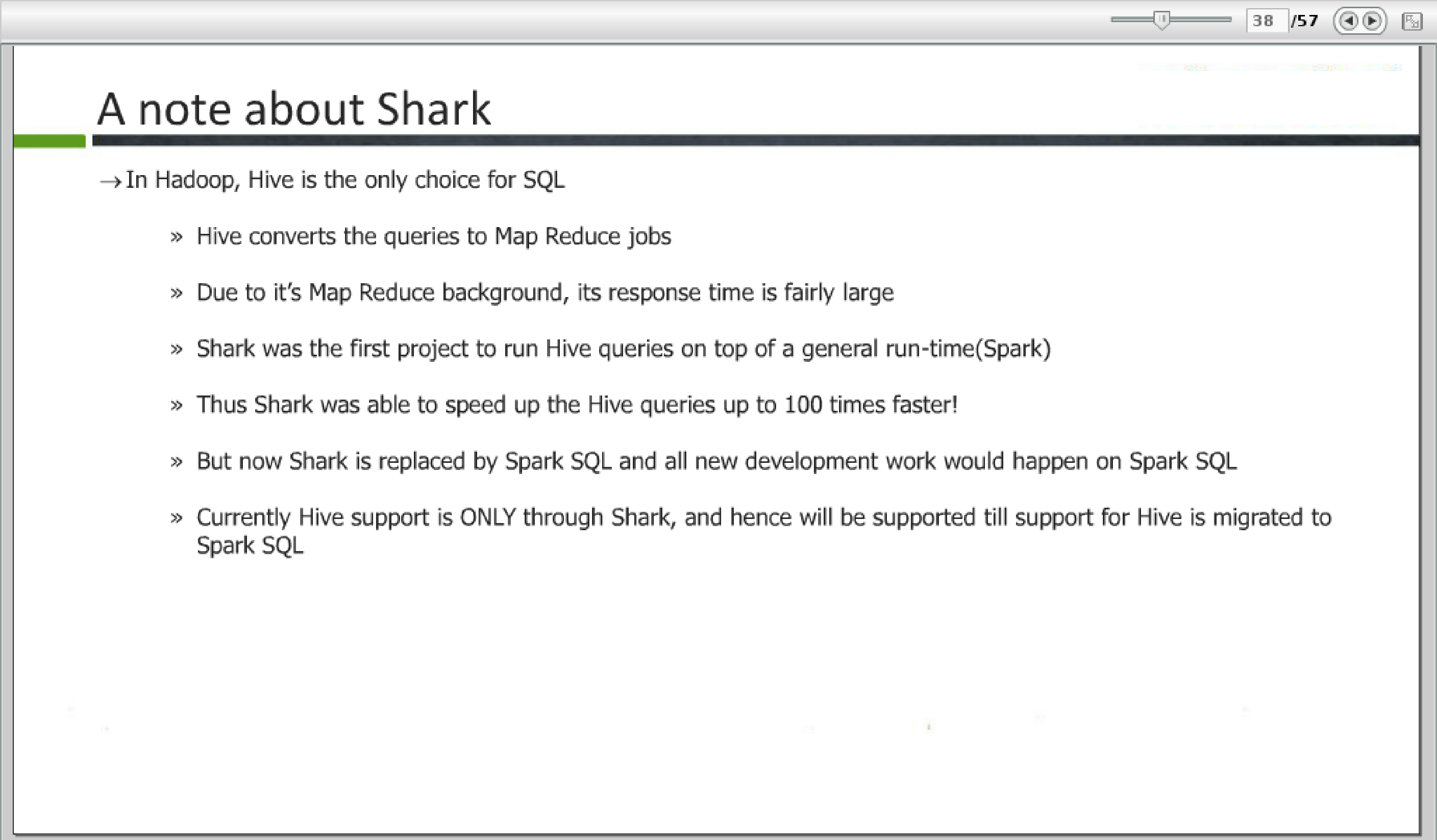
- » Spark is NOT limited to HDFS only for it's storage needs
- » HDFS provides distributed storage of large datasets
- » High Availability is assured natively through HDFS
- » No extra software installation is required
- » Compatible with Hadoop 1.x also. Using HDFS as storage doesn't require Hadoop 2.x
- » Data Loss during computation is handled by HDFS itself



# Using Hadoop as Execution Engine

→Spark can use Hadoop as execution engine

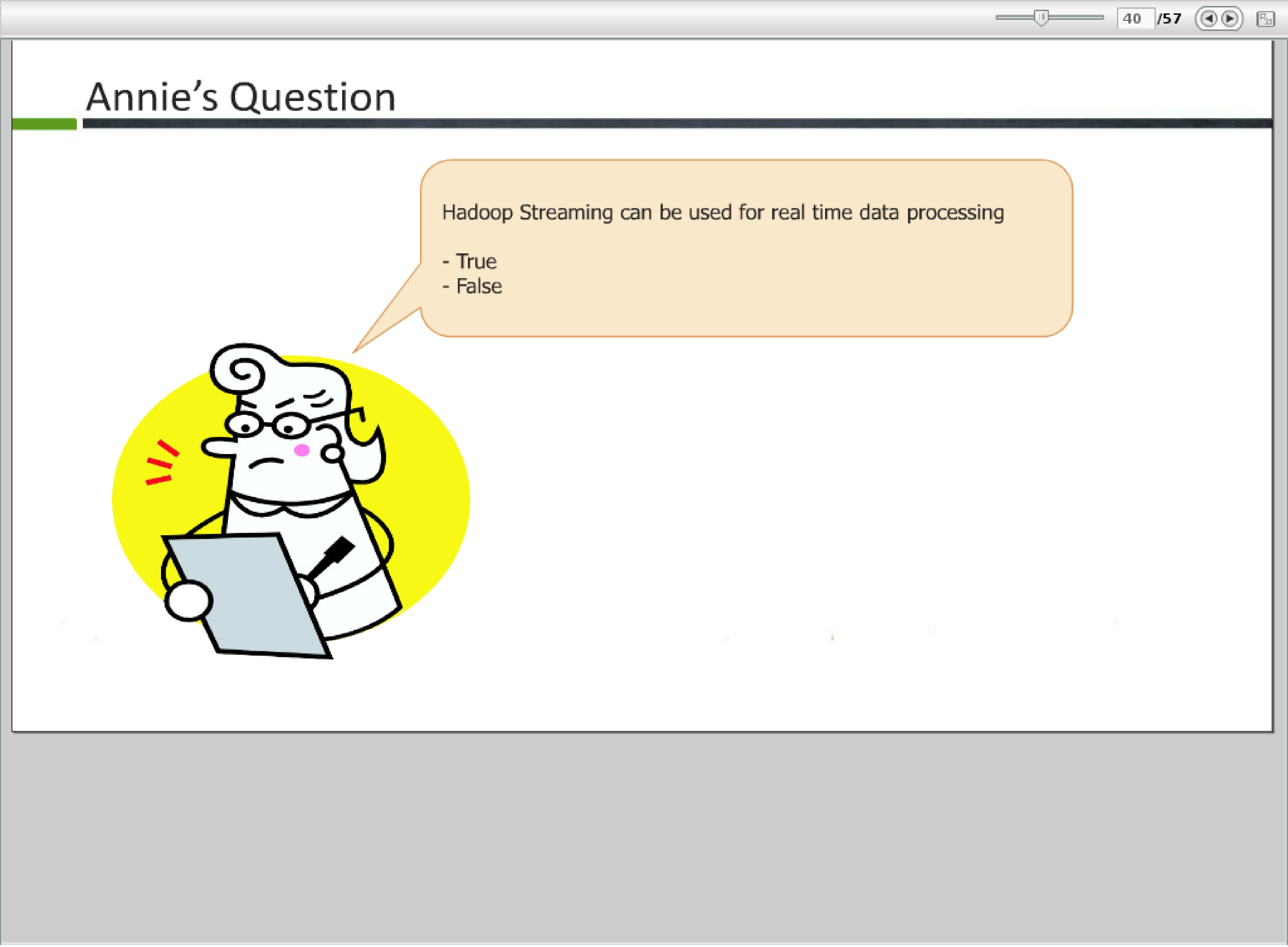
- » Spark can be integrated with Yarn for it's execution
- » Spark can be used with other engines (like Mesos, Spark Clsuter manager) also
- » Yarn integration automatically provides processing scalability to Spark
- » Spark needs Hadoop 2.0+ versions in order to use it for execution
- » Every node in Hadoop cluster need Spark also to be installed
- » Using Hadoop cluster for Spark processes, requires RAM upgrading of data nodes
- » The integration distribution of Spark is quite new and still in the process of stablization



# A note about Shark

→In Hadoop, Hive is the only choice for SQL

- » Hive converts the queries to Map Reduce jobs
- » Due to it's Map Reduce background, its response time is fairly large
- » Shark was the first project to run Hive queries on top of a general run-time(Spark)
- » Thus Shark was able to speed up the Hive queries up to 100 times faster!
- » But now Shark is replaced by Spark SQL and all new development work would happen on Spark SQL
- » Currently Hive support is ONLY through Shark, and hence will be supported till support for Hive is migrated to Spark SQL



# Annie’s Question

Hadoop Streaming can be used for real time data processing

- True
- False

# Annie's Answer

- False

# Annie’s Question

Hadoop is an ELT system:

- True
- False

# Annie’s Answer

- True

Press Esc to exit full screen mode.

# Annie's Question

The machine learning library of Spark is called:

- Mahout
- Mlib
- MLLib
- BlinkLib

# Annie’s Answer

-MLLib

# Annie’s Question

Shark is SQL engine of Spark for structured data:

- True
- False

# Annie’s Answer

- True

# Annie's Question

Data can be cached in Storm:

- True
- False

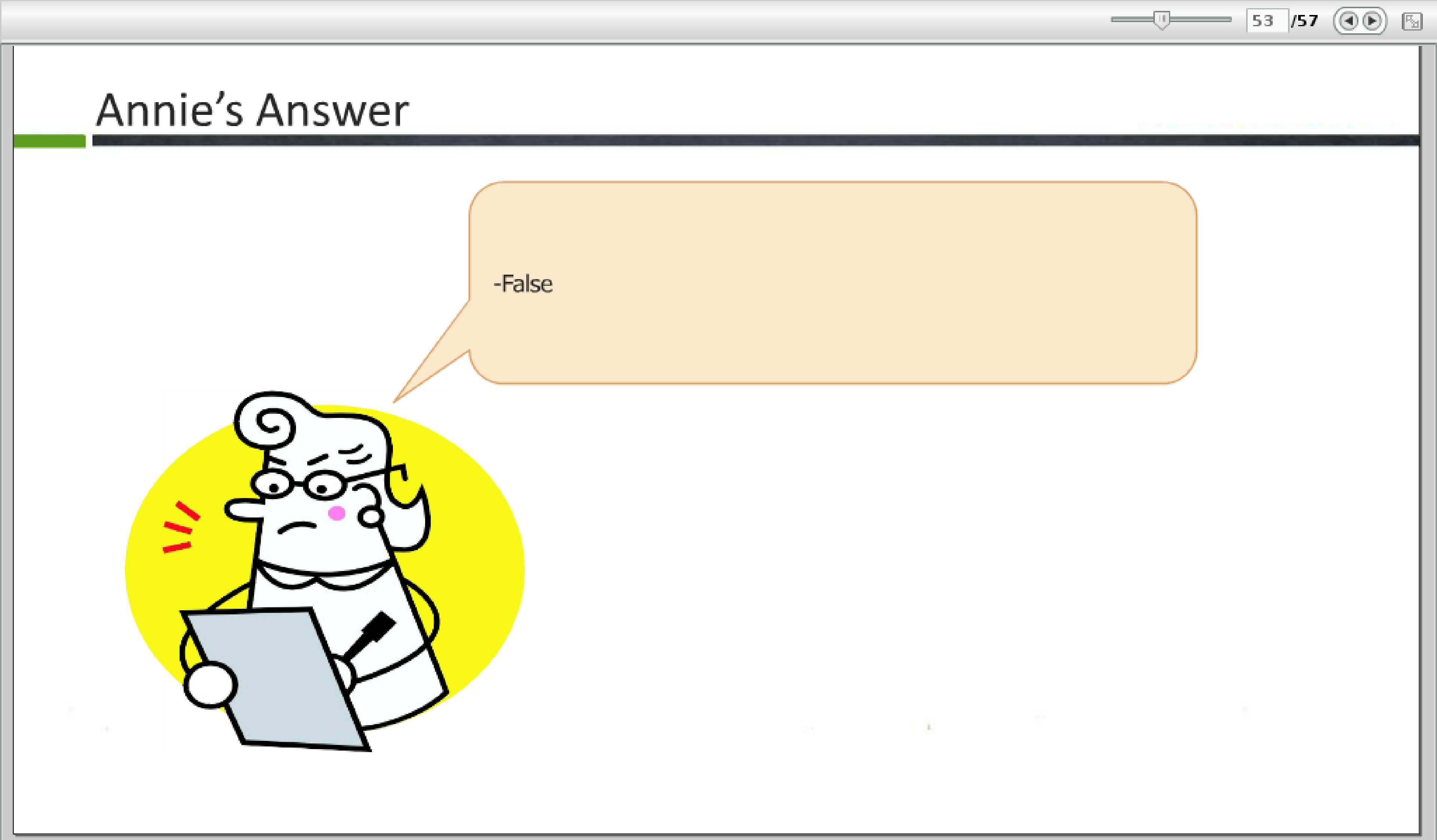
# Annie's Answer

- True

# Annie's Question

Spark doesn't use any Cluster manager for Stand-alone cluster mode-

- True
- False



# Annie's Answer

-False

57 /57

Thank you!