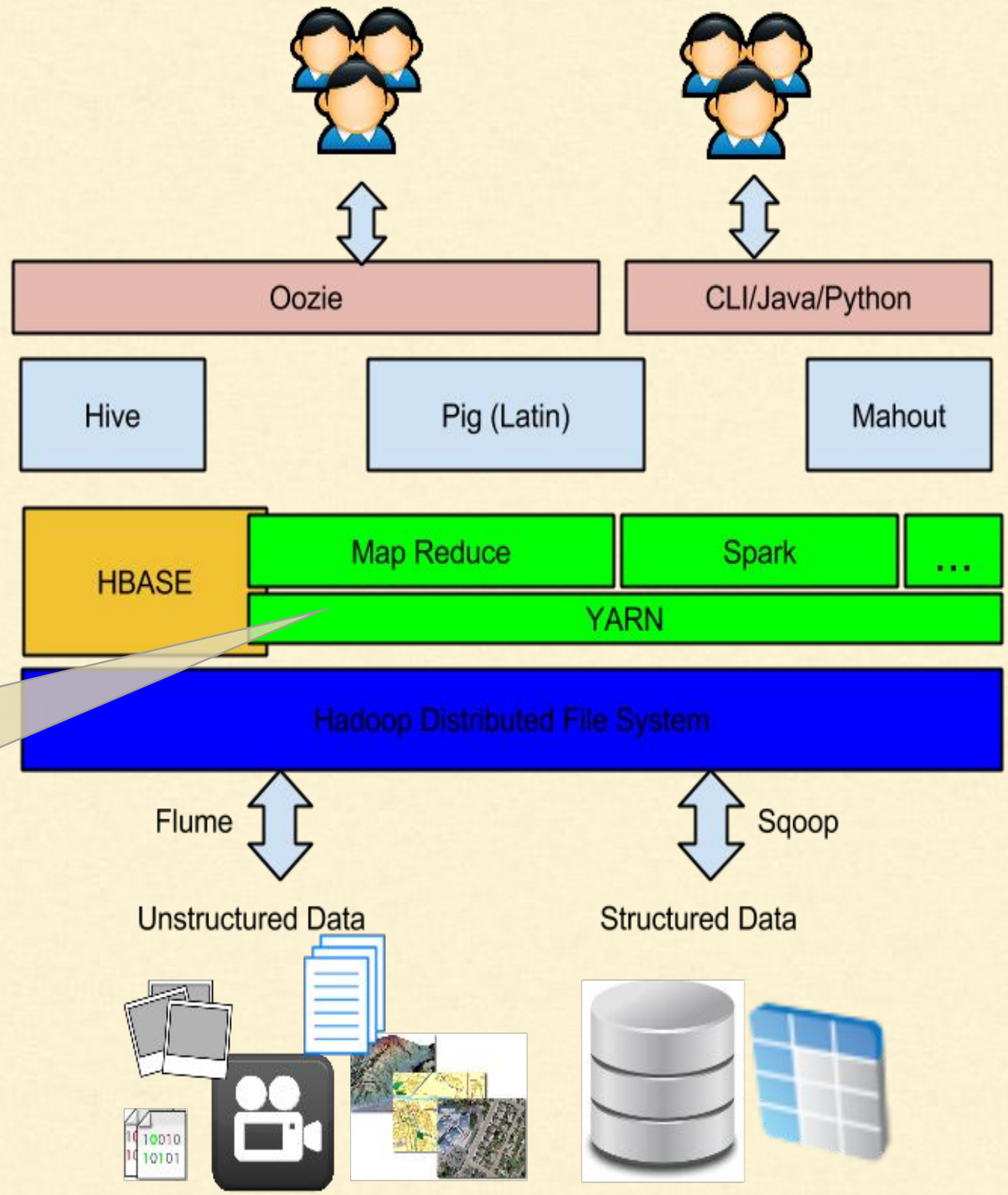
Welcome to YARN

Components



Yet Another Resource Negotiator

Big Data Problem - Processing

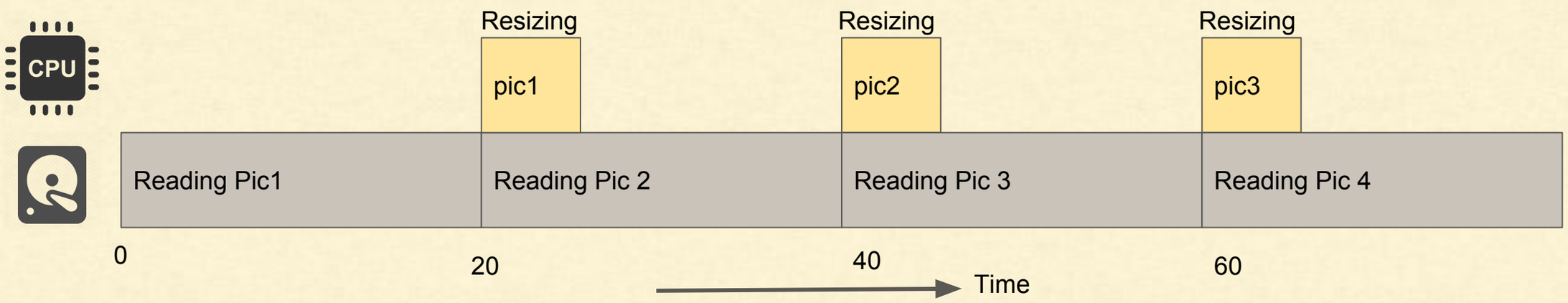
Q: How much time would it take for a (1Ghz, 1GB ram) computer to resize a million profile pics?

If it takes 20 millisecond to read the pic and 5 millisecond to resize?

Big Data Problem - Processing

Q: How much time would it take for a single computer to **resize a million profile pics**?
If it takes 20 milliseconds to read the pic and 5 millisecond to resize?

Note that on large scale it will take total 20ms for each image processing.



Big Data Problem - Processing

Q: How much time would it take for a single computer to resize a million profile pics?
If it takes 20 millisecond to read the pic and 5 millisecond to resize?

A: 10^6 * 20 ms = 20000 secs = 20000/3600 hrs **~5.6hours**

If we are using threads (parallel computing), then the main time consumed is in reading not in processing.

Big Data Problem - Processing

Q: How much time would it take for a single computer to resize a million profile pics?
If it takes 20 millisecond to read the pic and 5 millisecond to resize?

A: 10^6 * 20 ms = 20000 secs = 20000/3600 hrs **~5.6hours**

Q: Can this be improved if we buy a quad (4Ghz) core?

Big Data Problem - Processing

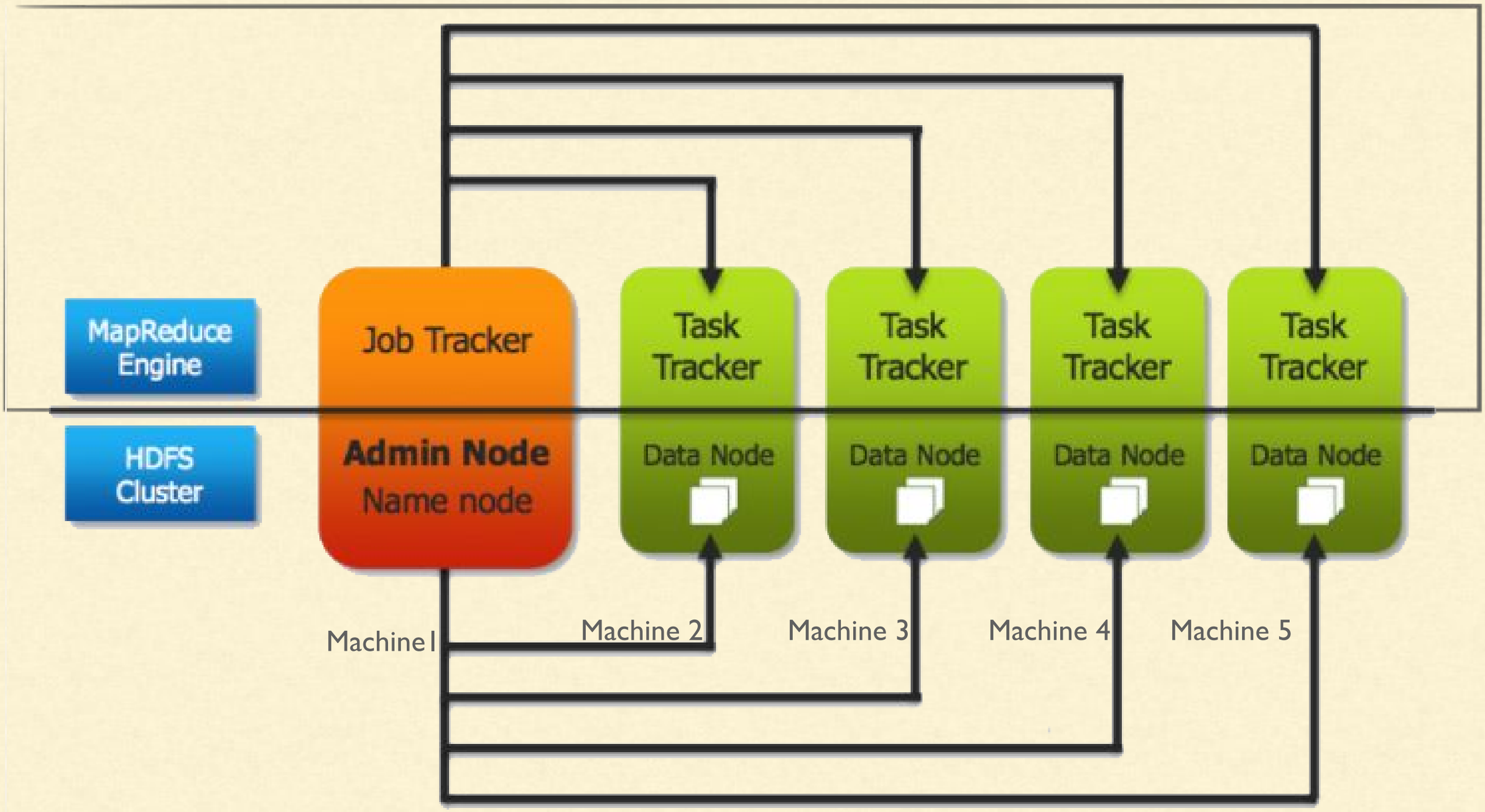
Q: How much time would it take for a single computer to resize a million profile pics?
If it takes 20 millisecond to read the pic and 5 millisecond to resize?

A: 10^6 * 20 ms = 20000 secs ~ 5.6hours

Q: Can this be improved if we buy a quad core?

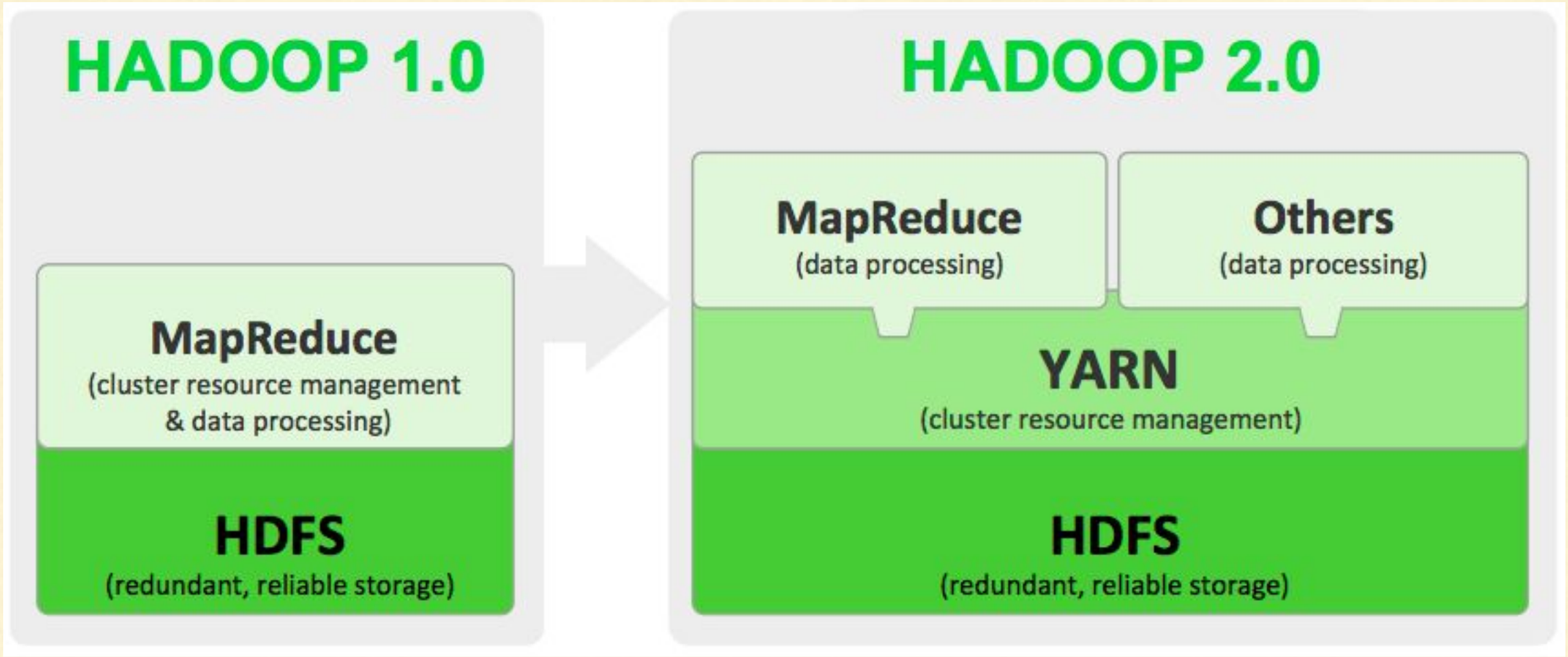
A: No

Core Components - Computation Engine
Map Reduce - 1.0



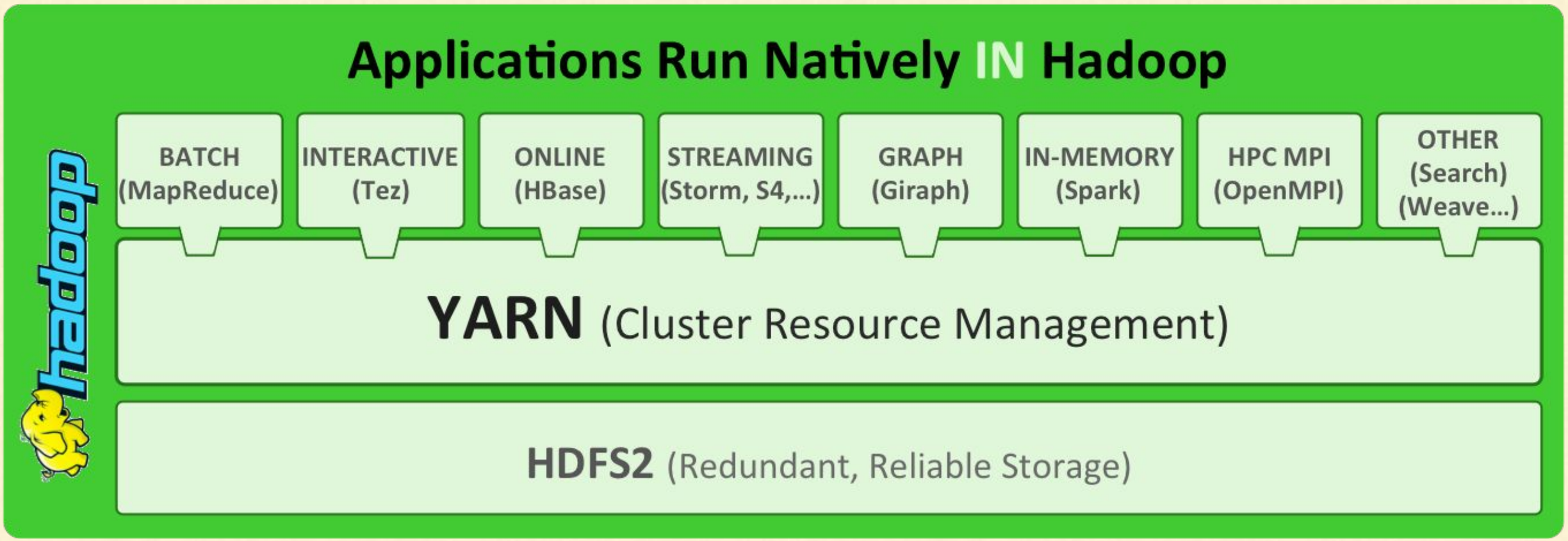
YARN

Yet Another Resource Negotiator



YARN

Yet Another Resource Negotiator



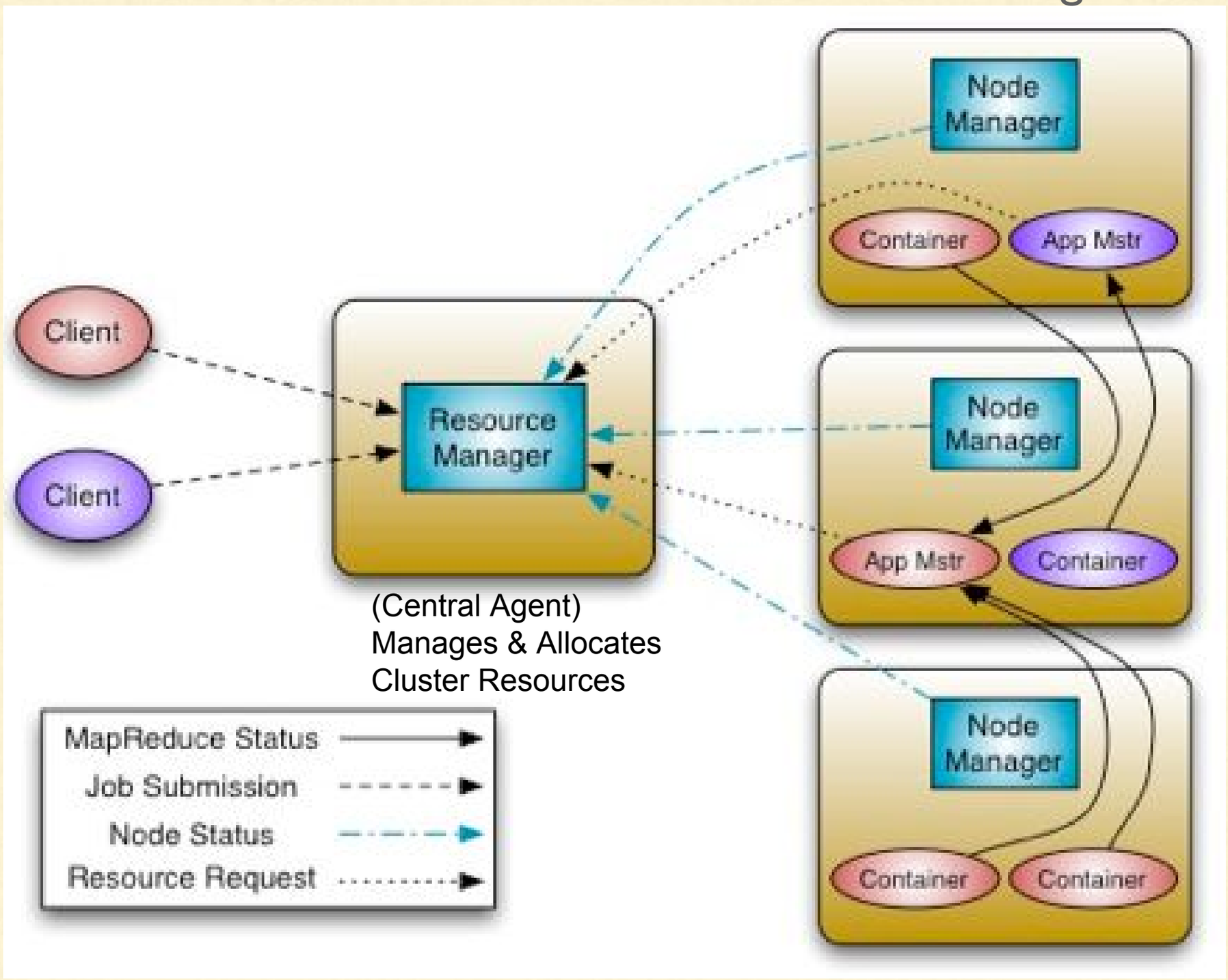
YARN

- Support for workloads other than MapReduce
- Scalability
- Compatibility with MapReduce
- Improved cluster utilization.
- Agility
- MapReduce was batch oriented

MapReduce 1	**YARN**
Jobtracker	Resource manager, application master, timeline server
Tasktracker	Node manager
Slot	Container

YARN - Hadoop 2.x

Yet Another Resource Negotiator



Node Manager
(Per Node Agent)
Manages & Enforces
Node Resource Allocations

Application Masters
(Per Application)
Manages:

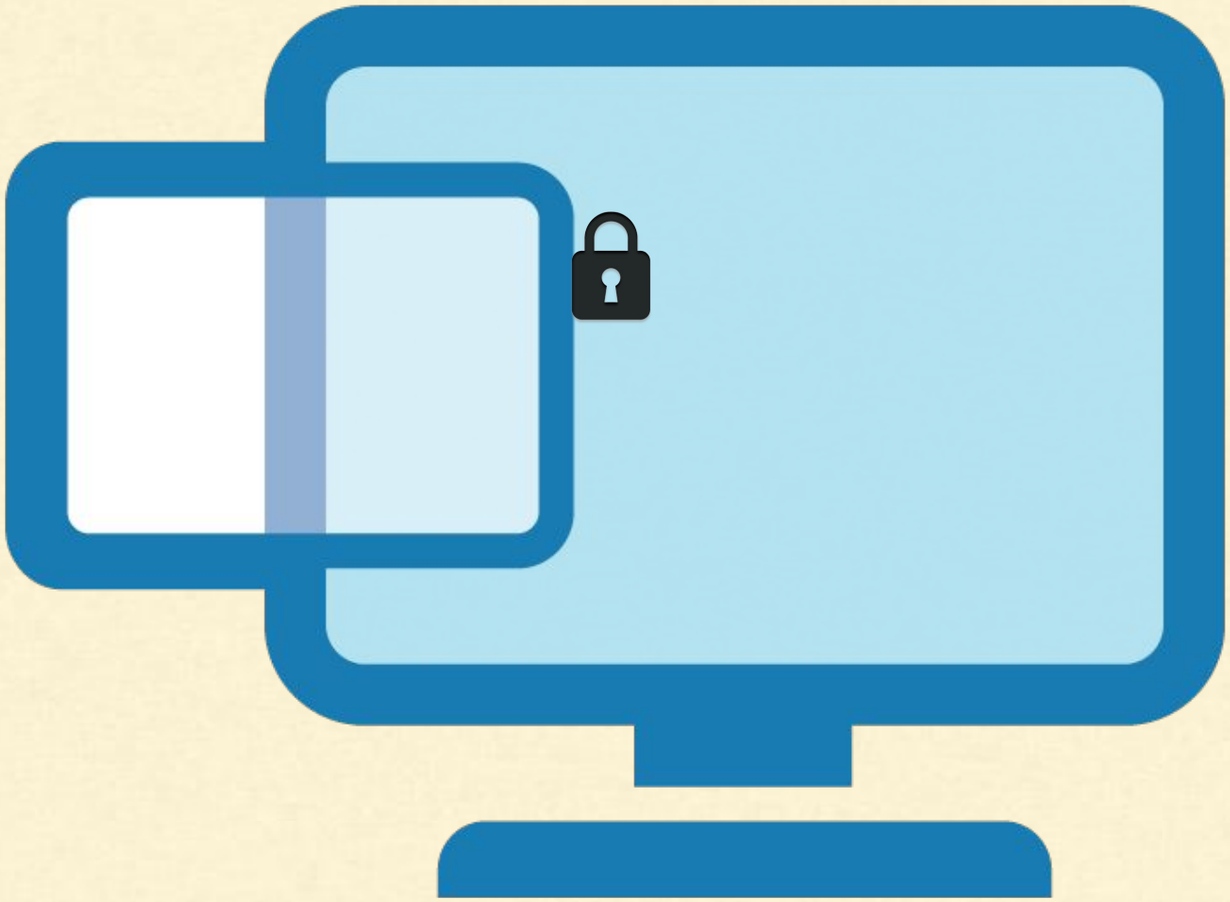
- Application Lifecycle
- Task Scheduling

Examples:

- MapReduce AM
- Spark AM

YARN - Container

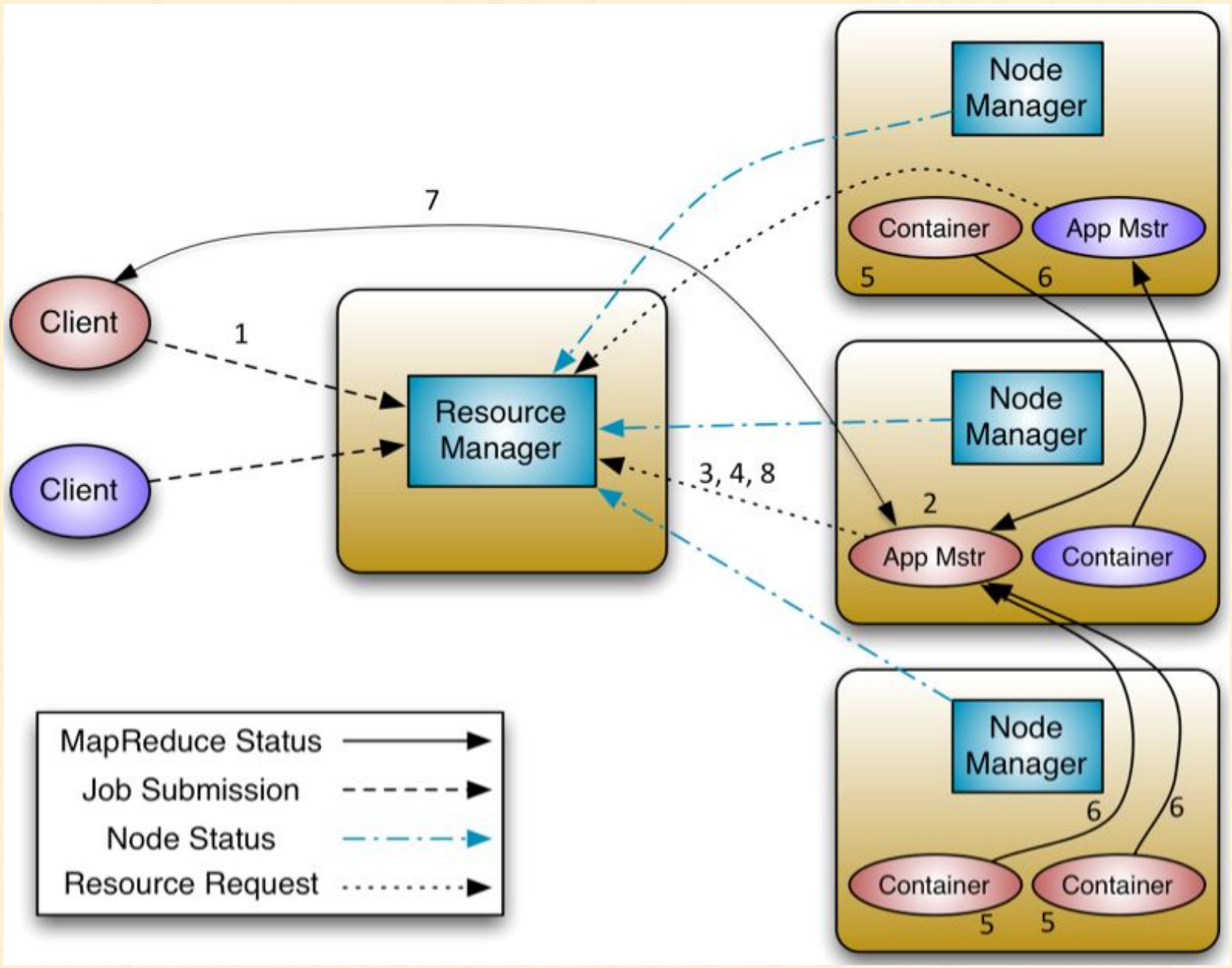
What is the role of Container?



- Process
- Runs client workload
- Enforces security on client processes
  - Memory Consumption
  - Resource Access

YARN - Hadoop 2.x

Yet Another Resource Negotiator



1. Client submits the App.
2. RM launches the AM in container
3. AM registers with RM. Client can connect to it.
4. AM asks for containers
5. RM allocates the containers
6. NM launches the app in these containers
7. NM talks back to RM about the resource usage
8. Clients talks to AM for status
9. On complete, AM deregisters with RM & Shutdown

President

Microsoft

Client

Client

Shell

Resource Manager

Node Manager

Governor

Shell head Office

Container

App Mstr

Office

CALIFORNIA REPUBLIC

Node Manager

App Mstr

Container

Head Office Microsoft

Node Manager

Container

Container

MapReduce Status

Job Submission

Node Status

Resource Request

YARN - Anatomy

A Resource Request

1. Includes
    a. Number of Containers
    b. Memory
    c. CPU
    d. Locality Constraints
2. Either upfront or as you go

Application Lifespan

- Can vary too dramatically
- Lifespan is categorised into three categories:
    - The simplest case is one application **per user job** (MapReduce)
    - One application per workflow or **user session** (Spark)
    - A **long-running** application (Apache Slider, impala) - Always On

YARN - Anatomy

Building Yarn Applications

- First, try to use existing frameworks: MapReduce, Spark etc.
- Second, try to utilise existing tools to Build Jobs:
    - Apache Slider(Run existing distributed Application such as HBase)
    - Apache Twill (Java Runnable)
- Else:
    - Building from scratch is complex
    - Use Yarn project's bundled example: distributed shell application

Thank You!

YARN - What happens Inside?