VOIDBOX - DOCKER ON YARN

YUMING LIANG

hulu

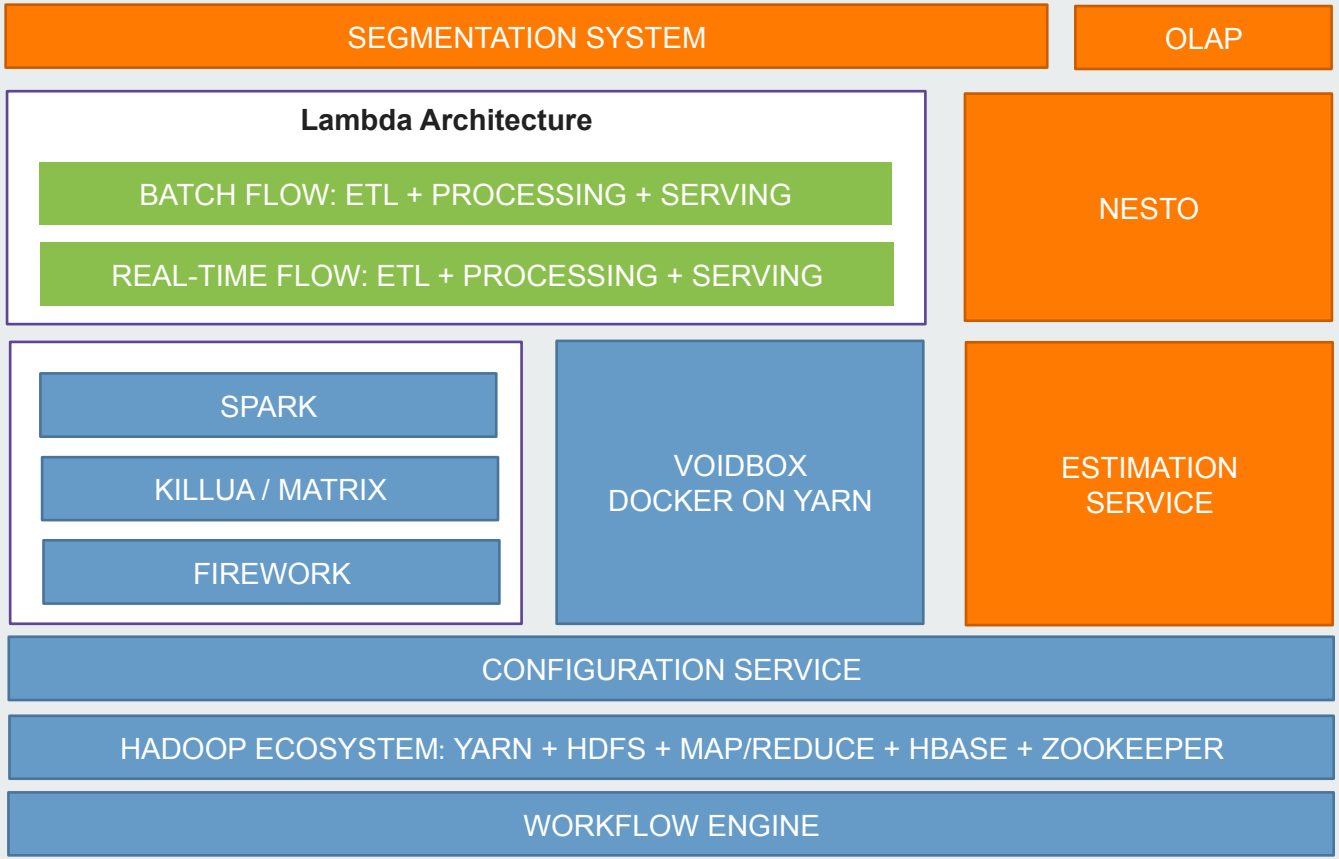
WHAT IS HULU

GAMING • CONNECTED TVS • MOBILE • COMPUTER


100+
HULU BEIJING

hulu

WHAT IS AUDIENCE PLATFORM



AGENDA

- Why Voidbox
- What is Voidbox
- Architecture
- How to use Voidbox
- Voidbox in Hulu
- What’s next
- Q&A

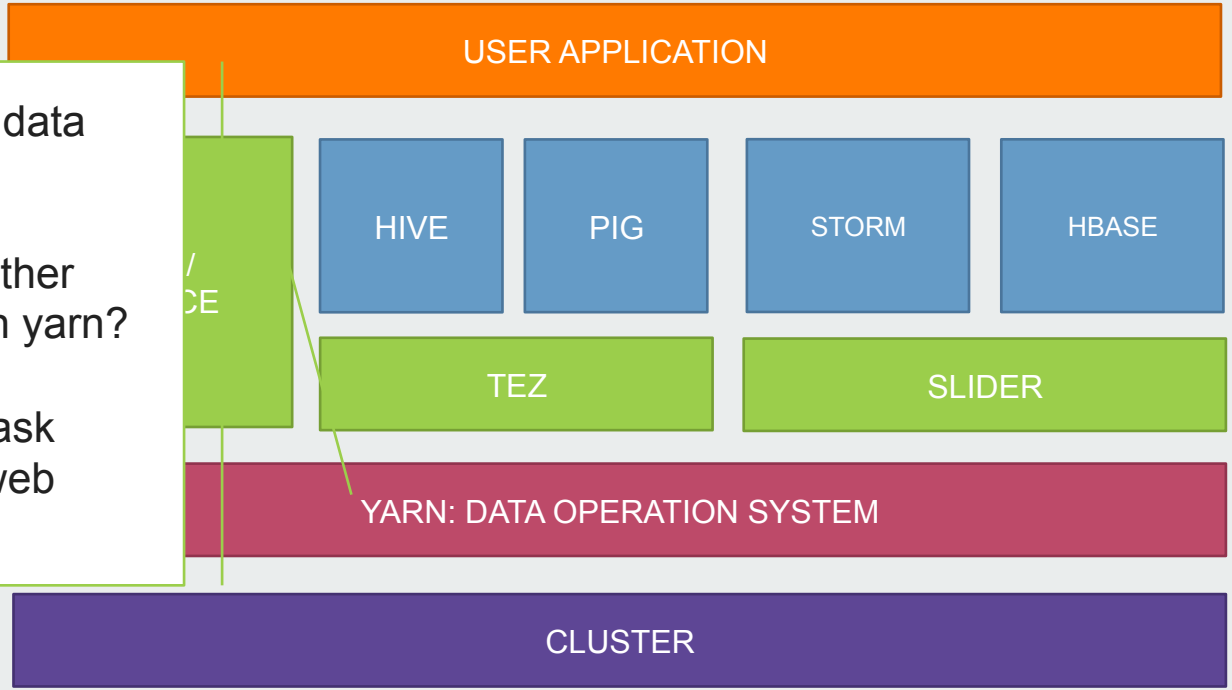
hulu

WHY VOIDBOX

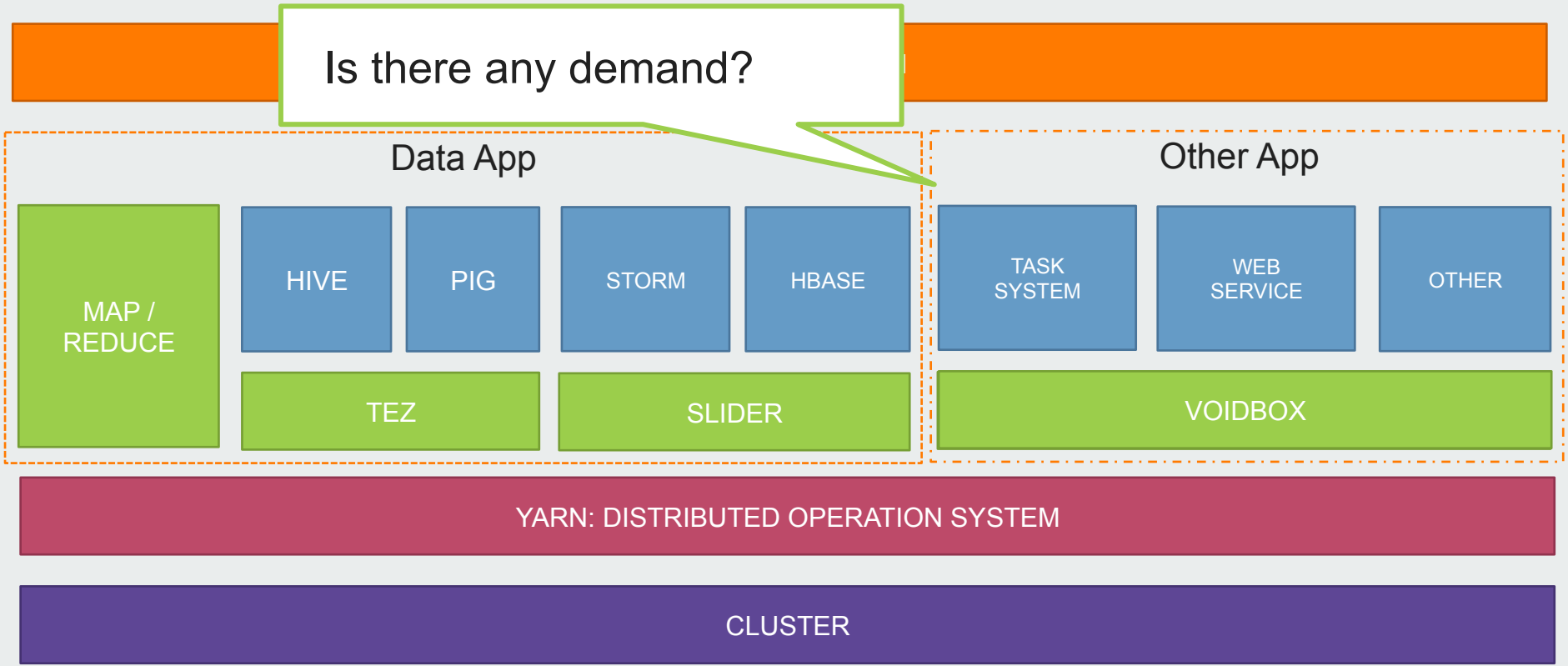
Why only big data application?

Can we run other application on yarn?

What about task processing, web service?

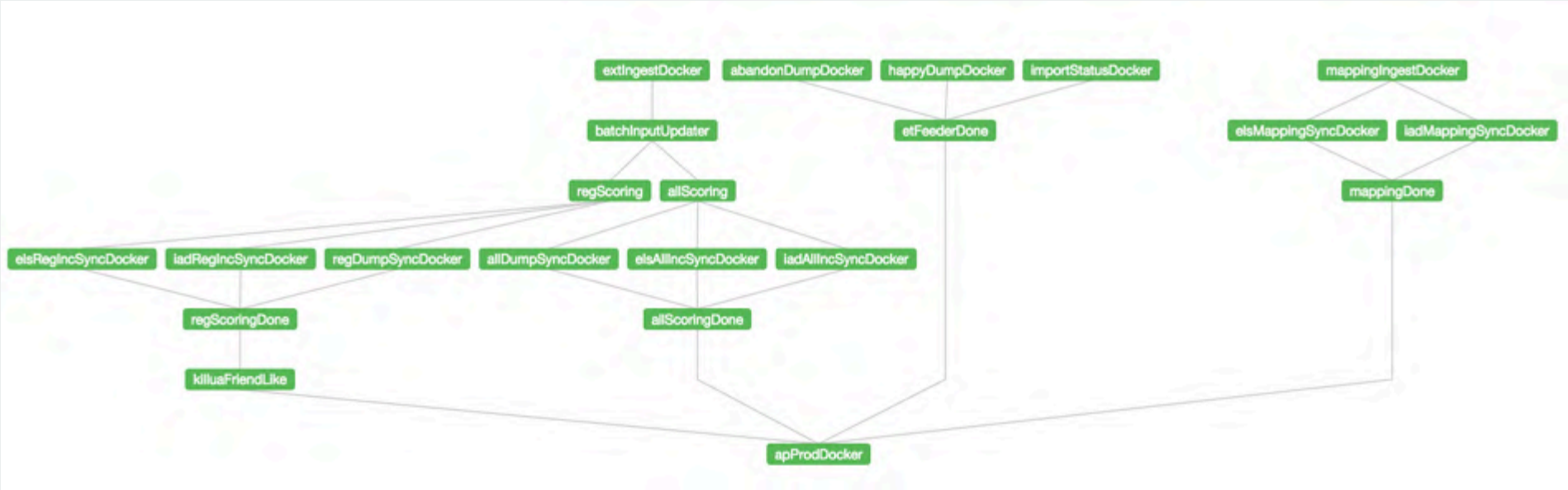


WHY VOIDBOX


Is there any demand?
Data App
Other App
MAP / REDUCE
HIVE
PIG
STORM
HBASE
TEZ
SLIDER
TASK SYSTEM
WEB SERVICE
OTHER
VOIDBOX
YARN: DISTRIBUTED OPERATION SYSTEM
CLUSTER

hulu

WHY VOIDBOX



20+ TASK IN DIFFERENT LANGUAGE → 20+ VIRTUAL MACHINE → LOW UTILIZATION / MACHINE FAILURE → VOIDBOX

WHAT IS VOIDBOX

VOIDBOX IS A PROGRAMMING FRAMEWORK

- LIGHTWEIGHT VM
- RUNTIME ISOLATION
- UNION FILE SYSTEM

+

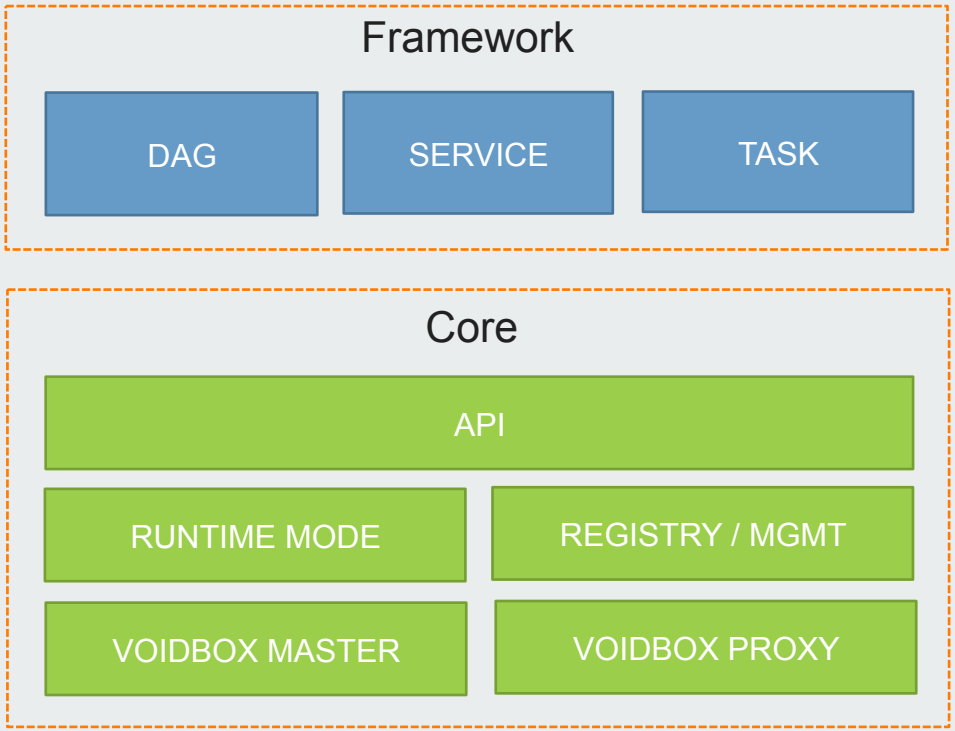
- RESOURCE MANAGEMENT
- SCHEDULING
- FAULT TOLERANCE

=

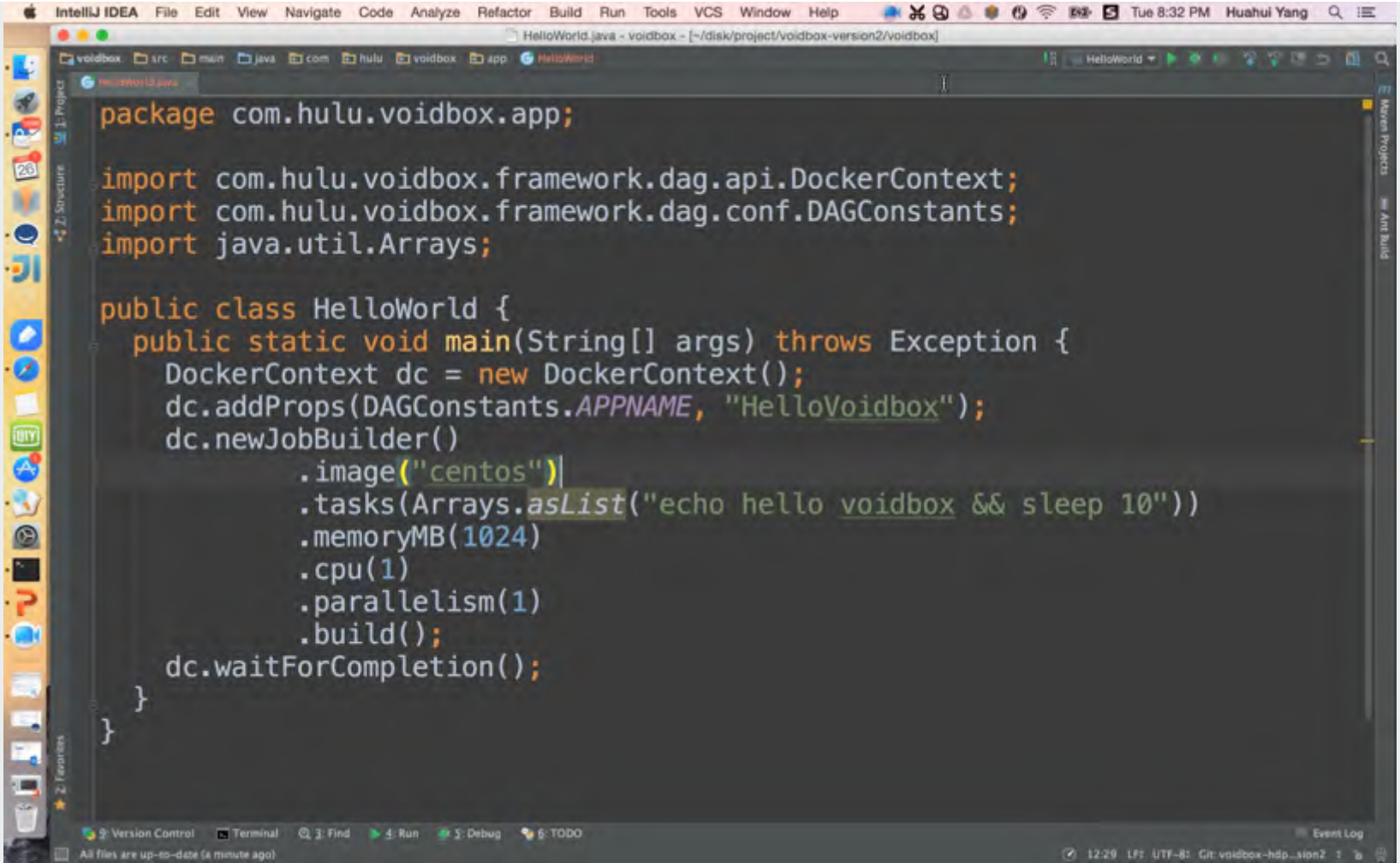
- DISTRIBUTE COMPUTATION
- ARBITRARY APPLICATION
- FAULT TOLERANCE

hulu

WHAT IS VOIDBOX



hulu

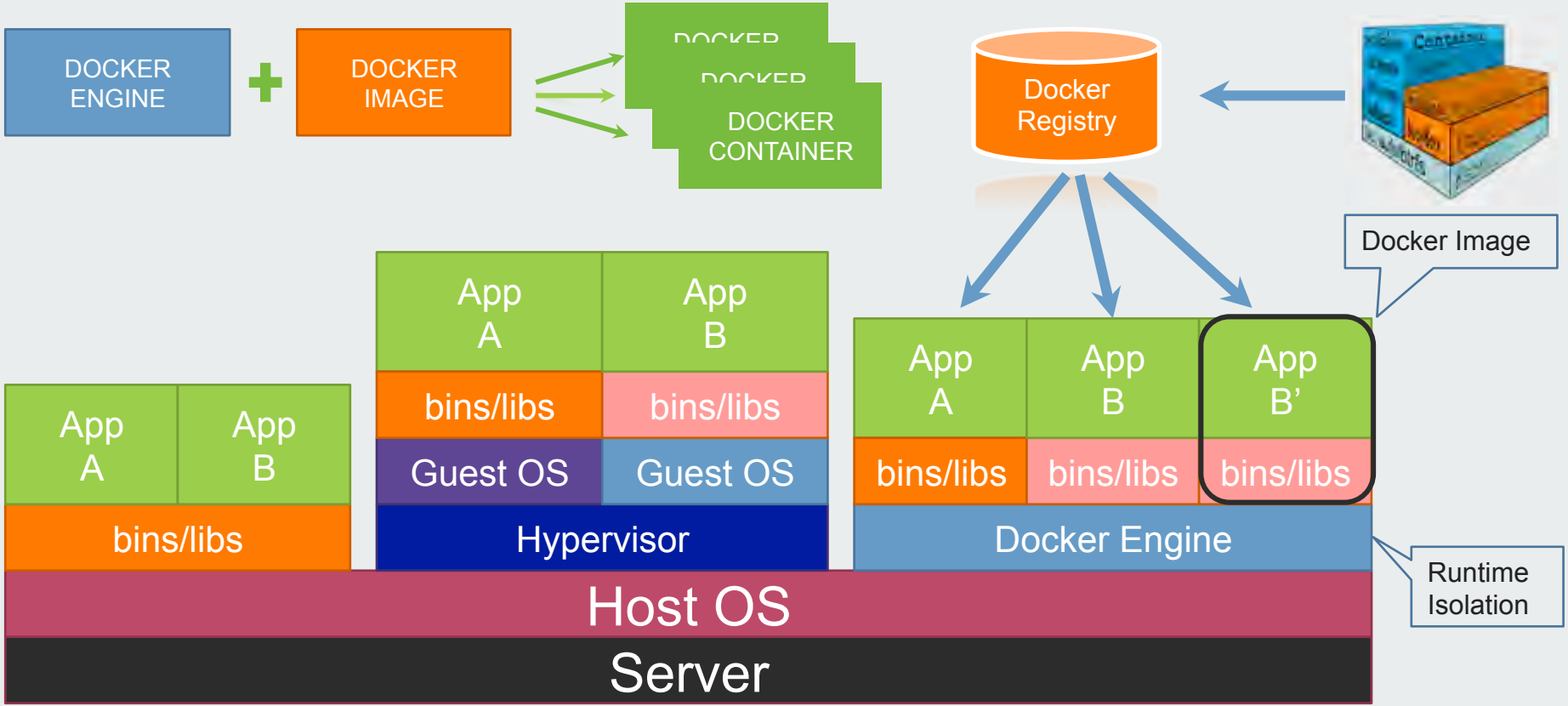

IntelliJ IDEA File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Tue 8:32 PM Huahui Yang
HelloWorld.java - voidbox - [~/disk/project/voidbox-version2/voidbox]
package com.hulu.voidbox.app;
import com.hulu.voidbox.framework.dag.api.DockerContext;
import com.hulu.voidbox.framework.dag.conf.DAGConstants;
import java.util.Arrays;
public class HelloWorld {
public static void main(String[] args) throws Exception {
DockerContext dc = new DockerContext();
dc.addProps(DAGConstants.APPNAME, "HelloVoidbox");
dc.newJobBuilder()
.image("centos")
.tasks(Arrays.asList("echo hello voidbox && sleep 10"))
.memoryMB(1024)
.cpu(1)
.parallelism(1)
.build();
dc.waitForCompletion();
}
}
9: Version Control
Terminal
3: Find
4: Run
5: Debug
6: TODO
Event Log
All files are up-to-date (a minute ago)

hulu

```
public static void main(String[] args) throws Exception {
  ServiceContext sc = new ServiceContext();
  sc.addProps(ServiceConstants.APPNAME, "Service Example");

  ServiceContainer memcachedContainer = ServiceContainer.newBuilder()
          .name("memcached").image("memcached").memoryMB(256).cpuShares(1)
          .build();

  ServiceContainer pythonContainer = ServiceContainer.newBuilder()
          .name("webapp").image("python-health-check").memoryMB(256).cpuShares(1)
          .publishTCPPort(5000)
          .build();

  ServiceGroup serviceGroup = ServiceGroup.newBuilder()
          .next(memcachedContainer).next(pythonContainer)
          .build();

  ServiceHealthCheck healthCheck = ServiceHealthCheck.newBuilder()
          .protocol(ServiceConstants.Protocol.HTTP)
          .containerPort(5000).successCode(200)
          .intervalSeconds(20).timeoutSeconds(10)
          .maxConsecutiveFailures(3)
          .path("/healthcheck")
          .build();

  ILoadBalance huluLoadBalance = new HuluLoadBalance("voidbox-long-service");

  sc.serviceBuilder()
          .setGroup(serviceGroup)
          .setHealthCheck(healthCheck)
          .setInstance(new ConstantsInstancesPolicy(3))
          .setLoadbalance(5000, huluLoadBalance, true)
          .build();

  sc.start();
}
```

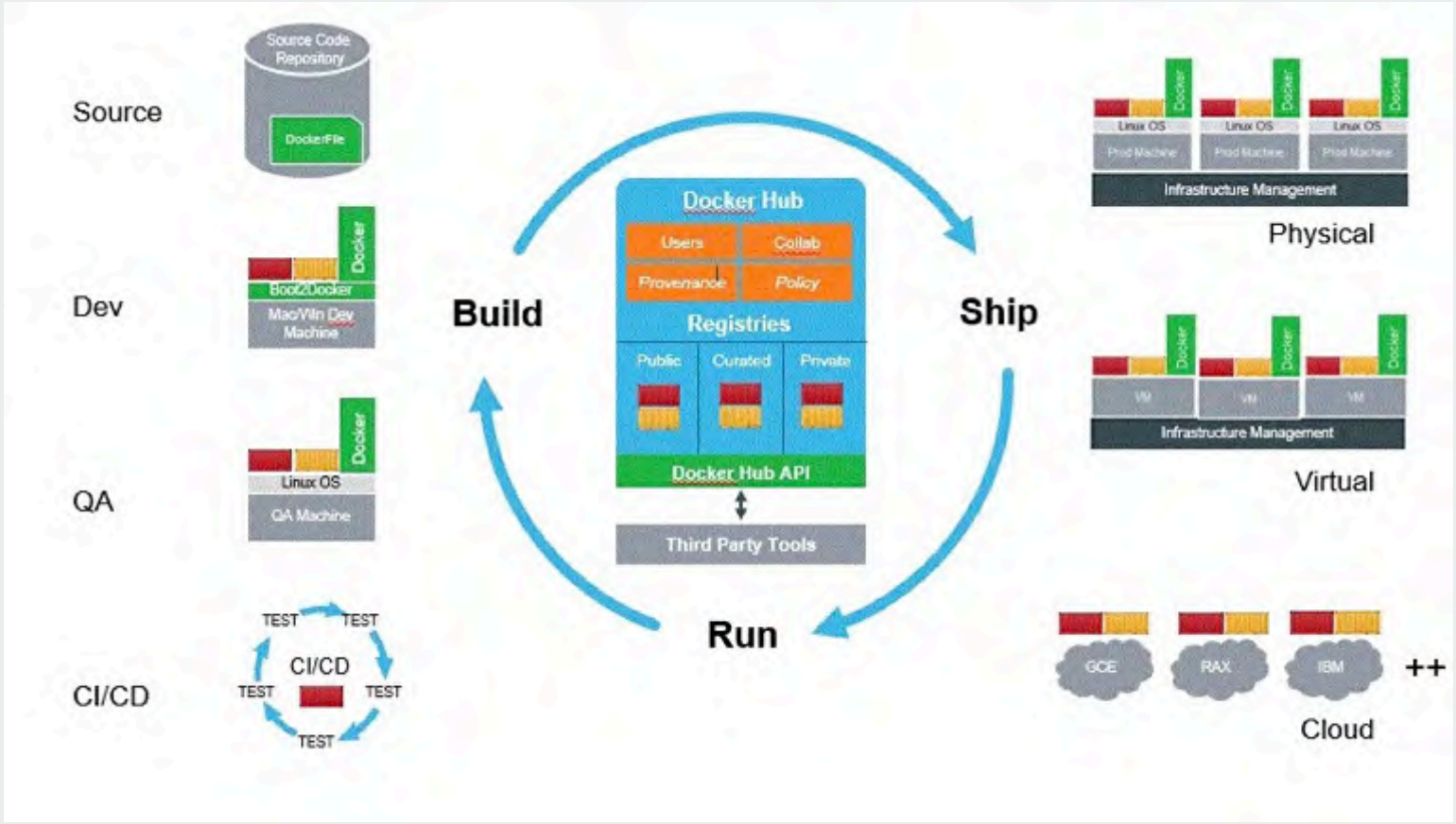
hulu

ARCHITECTURE – DOCKER

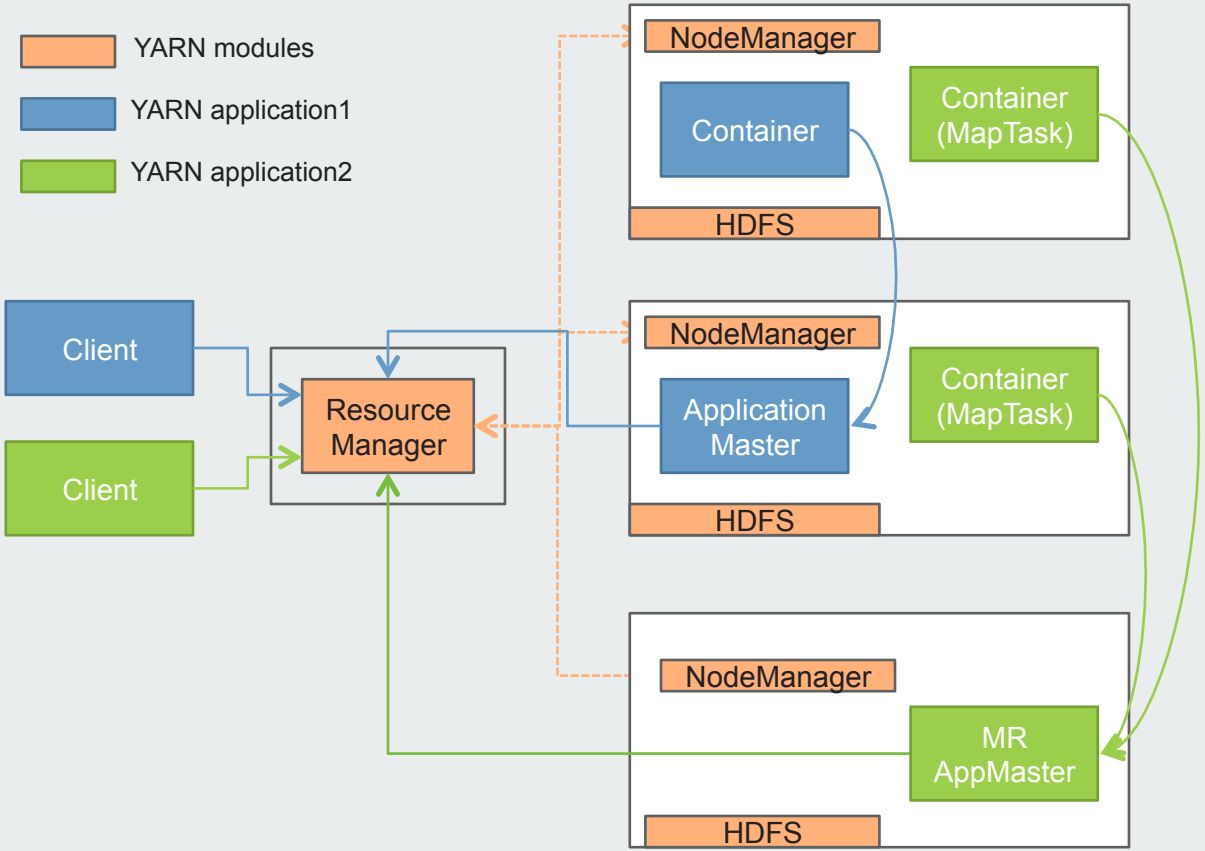


hulu

ARCHITECTURE – DOCKER

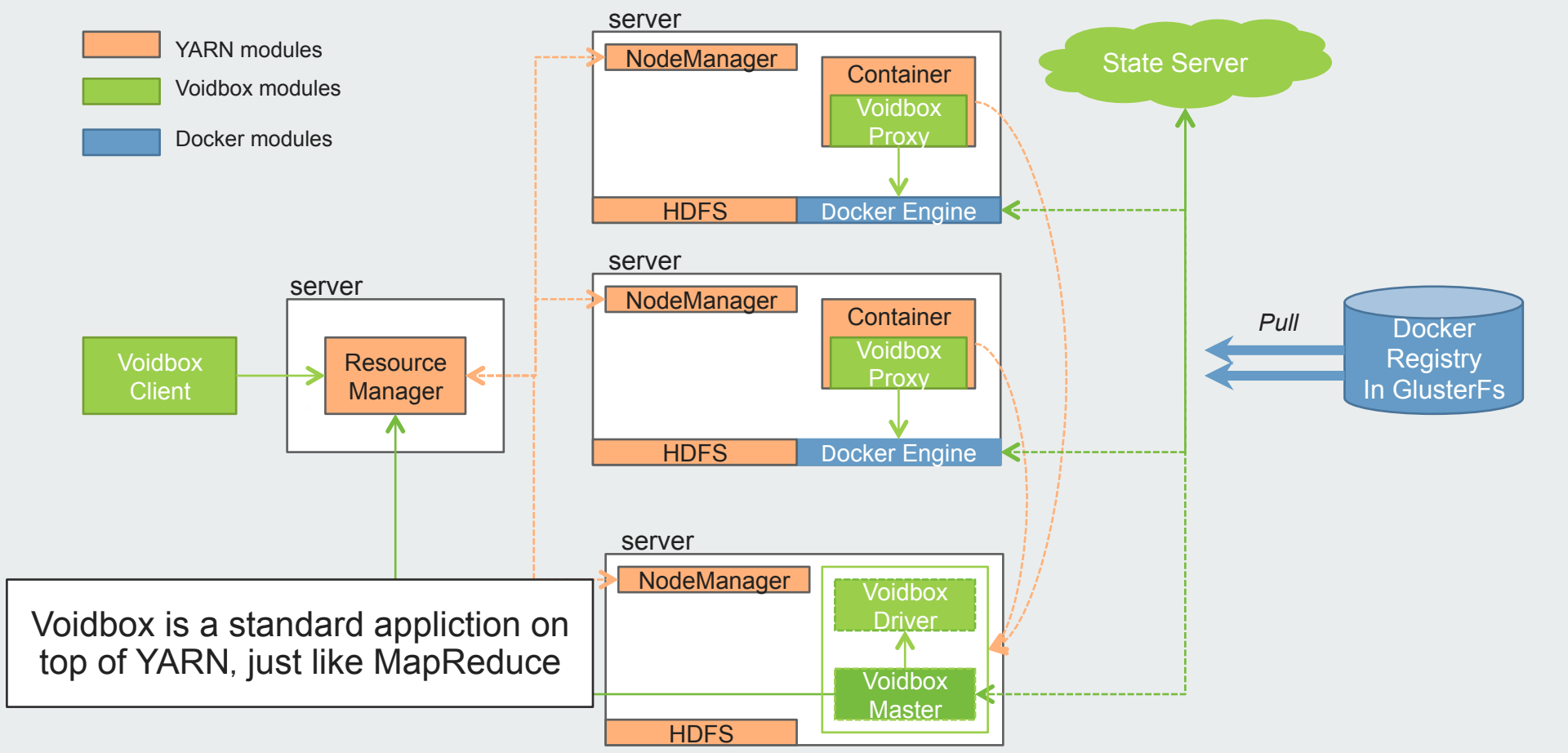


ARCHITECTURE – VOIDBOX



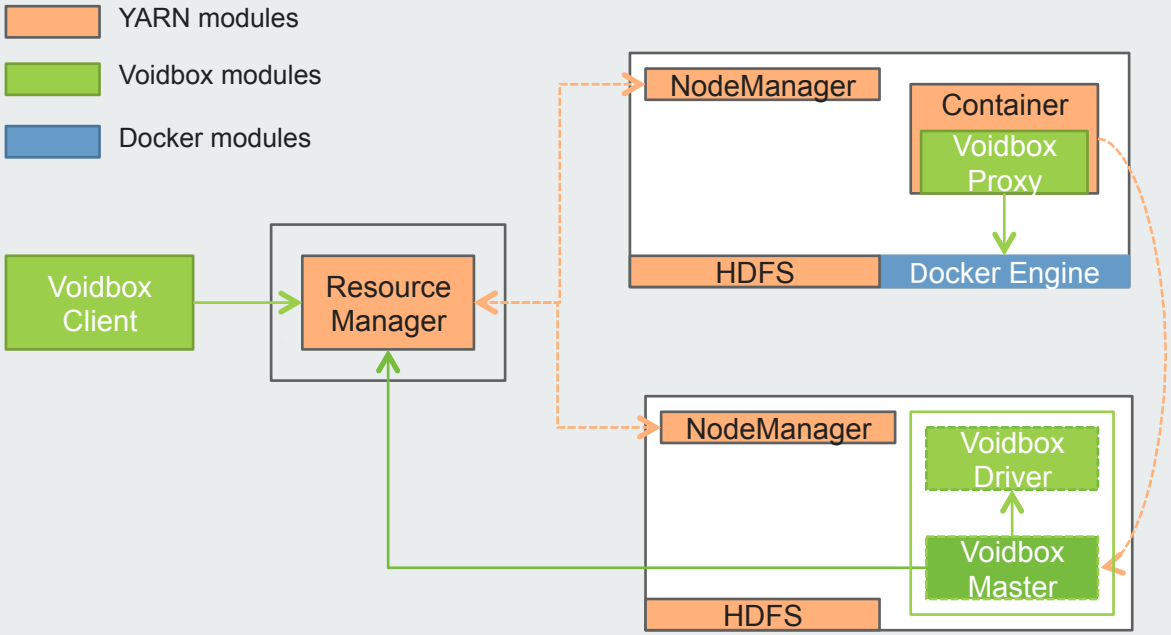
hulu

ARCHITECTURE – VOIDBOX



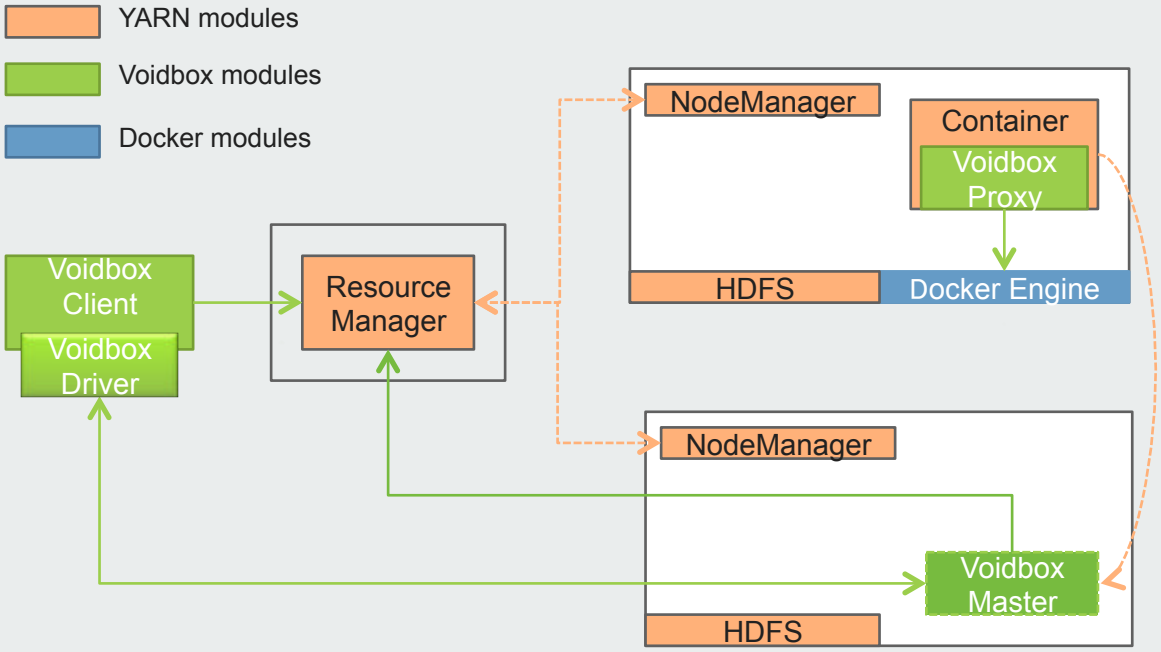
hulu

ARCHITECTURE – VOIDBOX YARN CLUSTER MODE



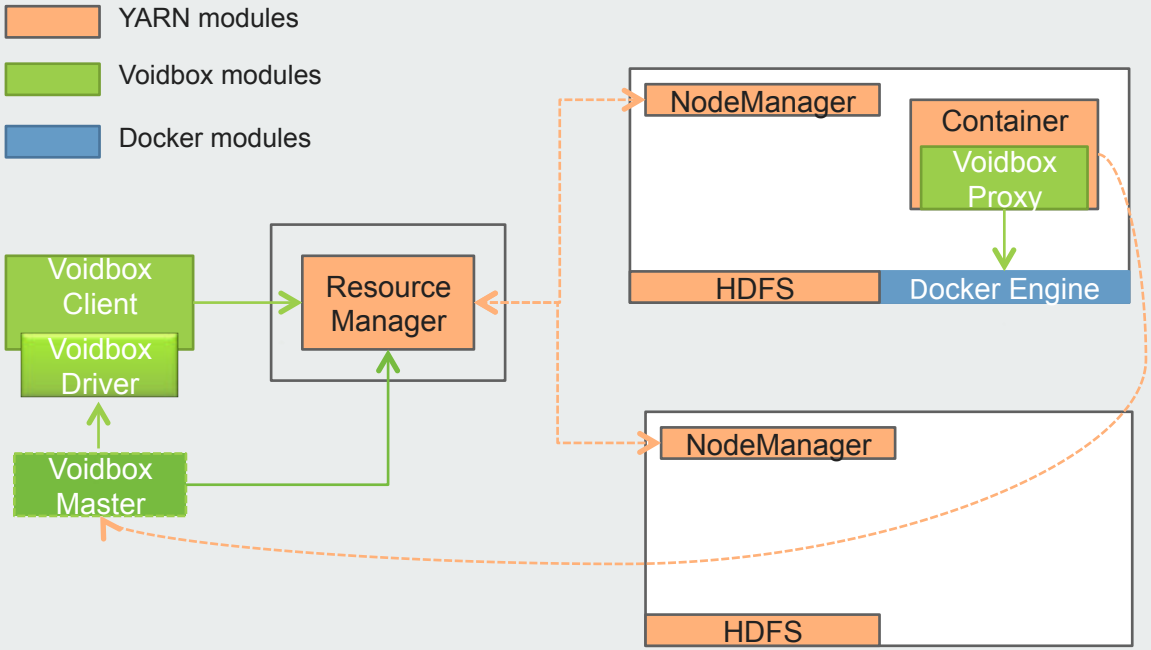
hulu

ARCHITECTURE – VOIDBOX YARN CLIENT MODE



hulu

ARCHITECTURE – VOIDBOX YARN LOCAL MODE



ARCHITECTURE – VOIDBOX FAULT TOLERANCE

- Work preserving in case ResourceManager restarts
  - https://issues.apache.org/jira/browse/YARN-556 target version:2.6.0
  - Improvement of ResourceManager HA
- Container preserving in case NodeManager restarts
  - https://issues.apache.org/jira/browse/YARN-1336
  - Avoid multiple ApplicationMasters
  - Avoid out-of-control containers in NM crashed machine
- ApplicationMaster retry count reset window
  - https://issues.apache.org/jira/browse/YARN-611
- YARN container −> Docker cli −> Docker container
  - Add shutdown hook to recycle docker container in case container unexpected exit
  - Deal with voidboxproxy exits without running shutdown hook(kill -9 proxy.pid)

hulu

ARCHITECTURE – VOIDBOX FAULT TOLERANCE

yarn-site.xml

```
<property>
   <name>yarn.resourcemanager.work-preserving-recovery.enabled</name>
   <value>true</value>
</property>
<property>
   <name>yarn.resourcemanager.am.max-attempts</name>
   <value>100</value>
</property>
<property>
   <name>yarn.nodemanager.recovery.enabled</name>
   <value>true</value>
</property>
```

ApplicationSubmissionContext

```
public abstract void setMaxAppAttempts(int maxAppAttempts);
public abstract void setAttemptFailuresValidityInterval(long attemptFailuresValidityInterval);
public abstract void setKeepContainersAcrossApplicationAttempts(boolean keepContainers);
```

ARCHITECTURE – VOIDBOX RESOURCE MANAGEMENT

- Management Center
  - Add/remove Docker engine in cluster
  - Garbage collection of out-of-control Container(NodeManage crashes)
  - History Log Server
  - Service discovery

ARCHITECTURE – LOGS

- VoidboxMaster's log
  - Specify log4j.properties to rotate master's log when submit an application
- Docker container's log
  - Stdout, stderr log
    - Copyfile, truncate(0) to rotate docker container's log
    - Rotate log file in host(default:/var/lib/docker/containers/../containerid−json.log)
  - A log dir in Docker image
    - Map specify log volume, by user-defined log rotate in Docker image

hulu

ARCHITECTURE – KEY COMPONENTS

CORE

FRAMEWORK

APP

hulu

ARCHITECTURE – KEY COMPONENTS

CORE

FRAMEWORK

APP

- Functionality
  - Allocate/release container
  - Launch/stop task in container
  - Multi run mode: yarn-cluster, yarn−client, yarn−local
- Component
  - VoidboxMaster
    - Container resource allocation/release/preserving
    - Container context management
  - VoidboxProxy
    - Docker container lifecycle management
  - Management Center
    - Docker engine management
    - Garbage collection

ARCHITECTURE – KEY COMPONENTS

CORE

FRAMEWORK

APP

- DAG framework
  - DAGDriver
    - Task scheduler
    - Task retry policy
  - DAGContext
    - Describe DAG docker job
- Service framework
  - ServiceDriver
    - Replication controller
    - Health check, warm up
  - ServiceContext
    - Describe service docker job

hulu

ARCHITECTURE – KEY COMPONENTS

CORE

FRAMEWORK

APP

- DAG APP
  - DAG programming based on Framework
- Service APP
  - Service programming based on Framework
  - Register to load balance

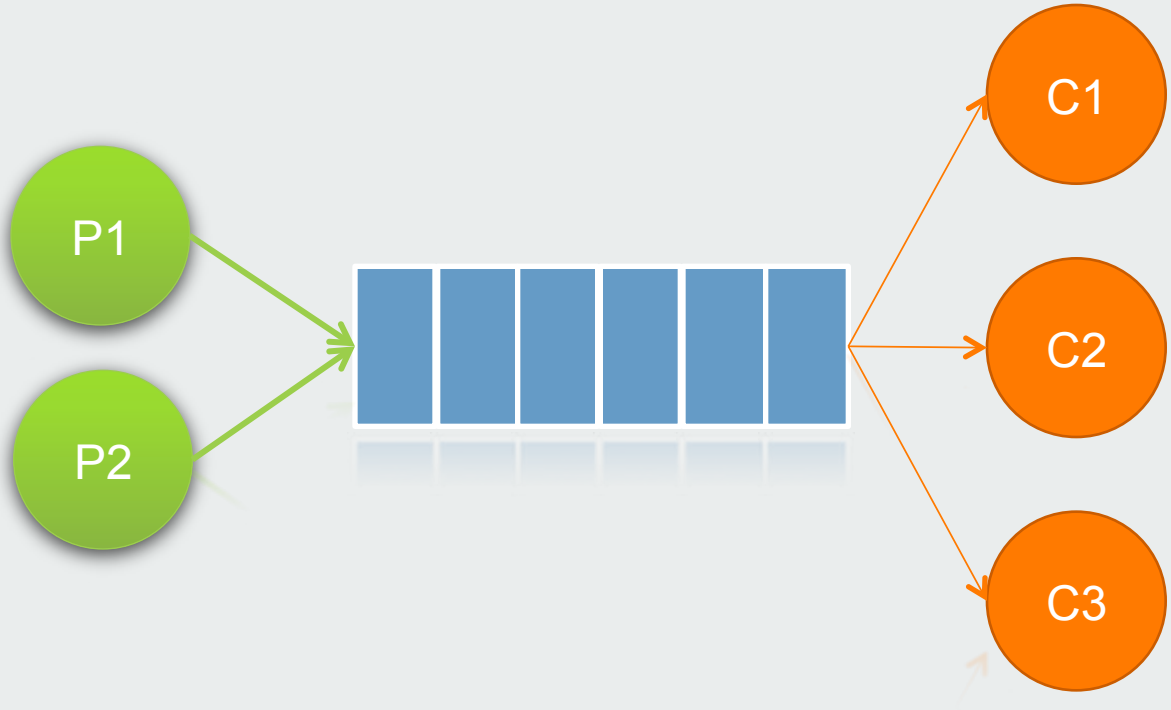
HOW TO USE VOIDBOX

Advanced API:

- AMClient
  - requestResource(cpu, memory)
  - releaseResource(container)

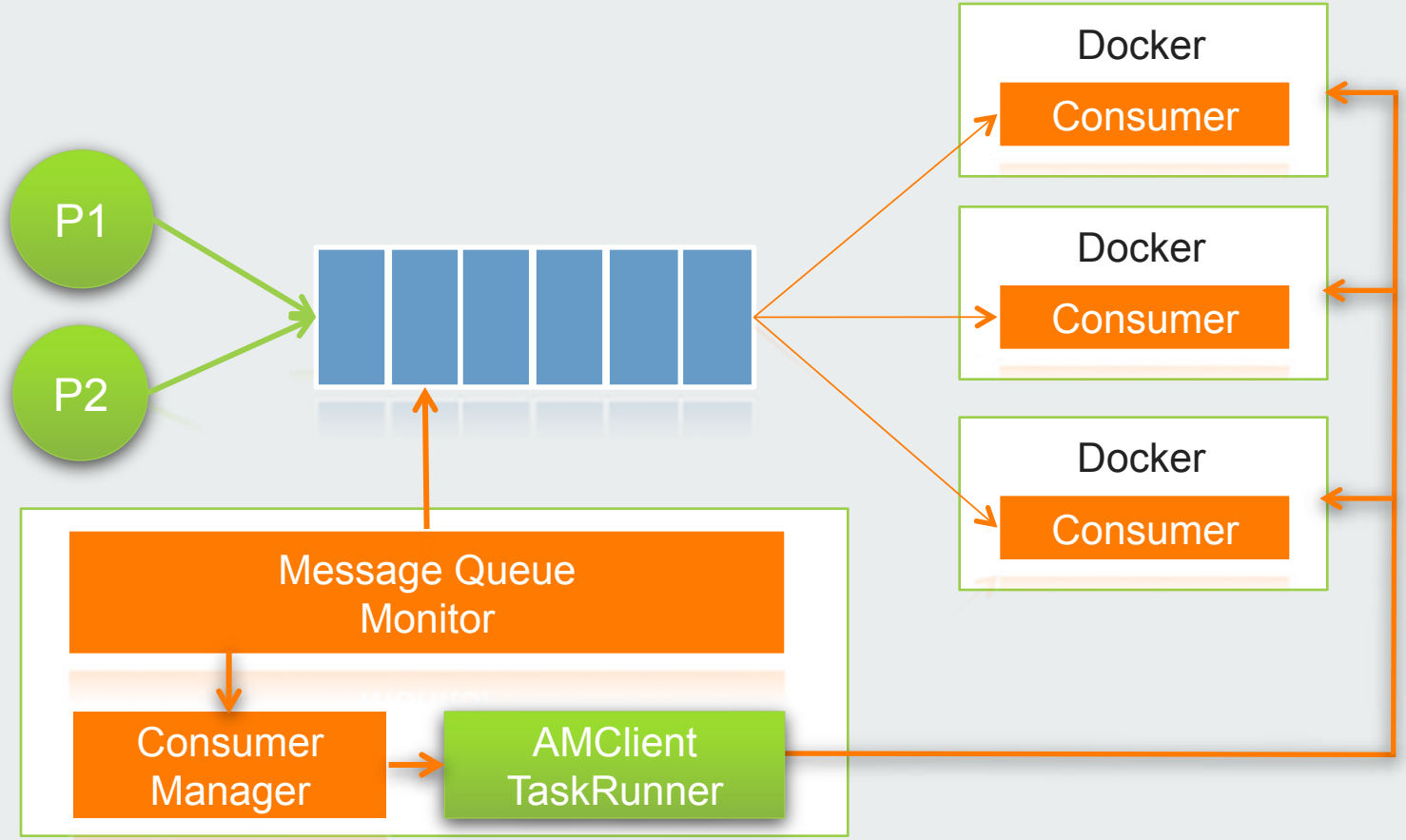
- DockerTaskRunner
  - run(task)

HOW TO USE VOIDBOX



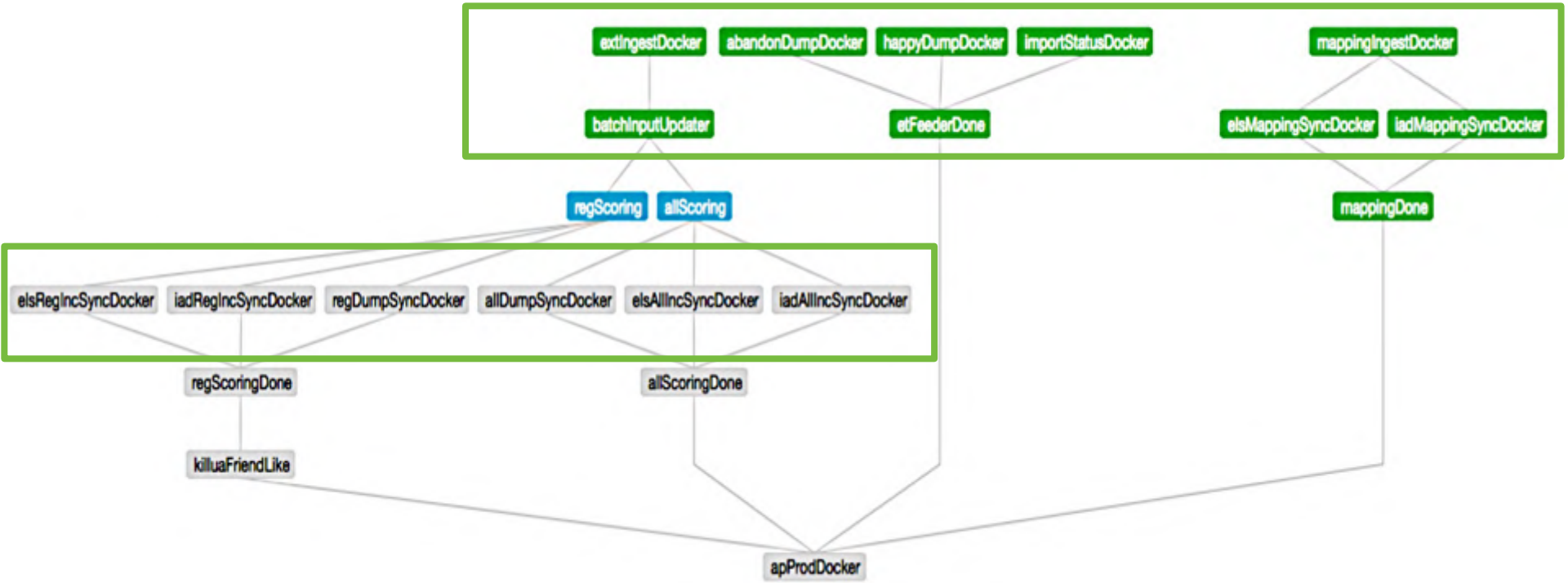
How to increase / decrease number of consumers automatically?

HOW TO USE VOIDBOX



hulu

VOIDBOX IN HULU



hulu

VOIDBOX IN HULU

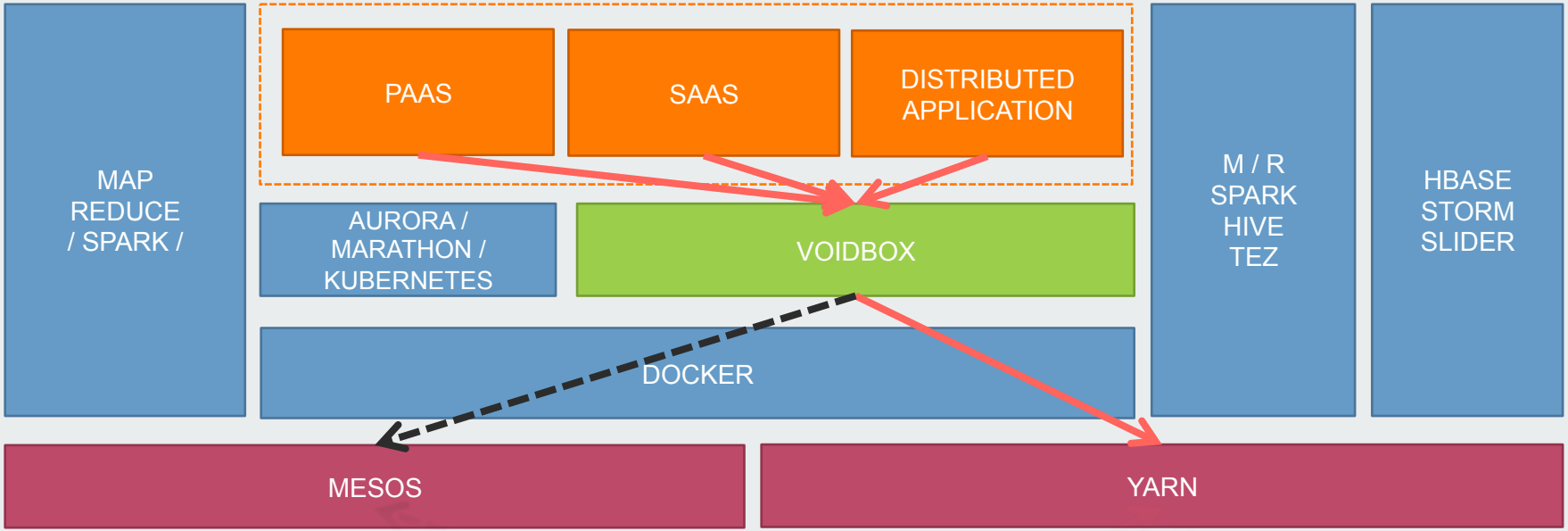
RESOURCE BOUND + HIGHLY PARALLELIZABLE + COMPLEX ENVIRONMENT DEPENDENCIES → VOIDBOX APPLICATION

hulu

VOIDBOX BENEFITS

- Ease creating distributed applications
  - Build-in service discovery, fault tolerance, scheduling, communication
  - Support complex environment dependencies
- Improve cluster efficiency
  - Share cluster with other data application
- Simplify deployment
  - Package only
  - No deployment
- Flexibility
  - Programming interface

WHAT'S NEXT


MAP
REDUCE
/ SPARK /
PAAS
SAAS
DISTRIBUTED
APPLICATION
AURORA /
MARATHON /
KUBERNETES
VOIDBOX
DOCKER
M / R
SPARK
HIVE
TEZ
HBASE
STORM
SLIDER
MESOS
YARN


Huahui Yang
Xicheng Dong


Q & A
QUESTIONS


hulu
THANK YOU