Running Apache Spark & Apache Zeppelin in Production

Vinay Shukla
Director, Product Management
August 31, 2016
Twitter: @neomythos

HORTONWORKS®
POWERING THE FUTURE OF DATA™

Who am i?

Vinay Shukla

- Product Management
- Spark for 2.5 + years, Hadoop for 3+ years
- Recovering Programmer
- Blog at www.vinayshukla.com
- Twitter: @neomythos
- Addicted to Yoga, Hiking, & Coffee
- Smallest contributor to Apache Zeppelin

HORTONWORKS

Security: Rings of Defense

Perimeter Level Security

- Network Security (i.e. Firewalls)

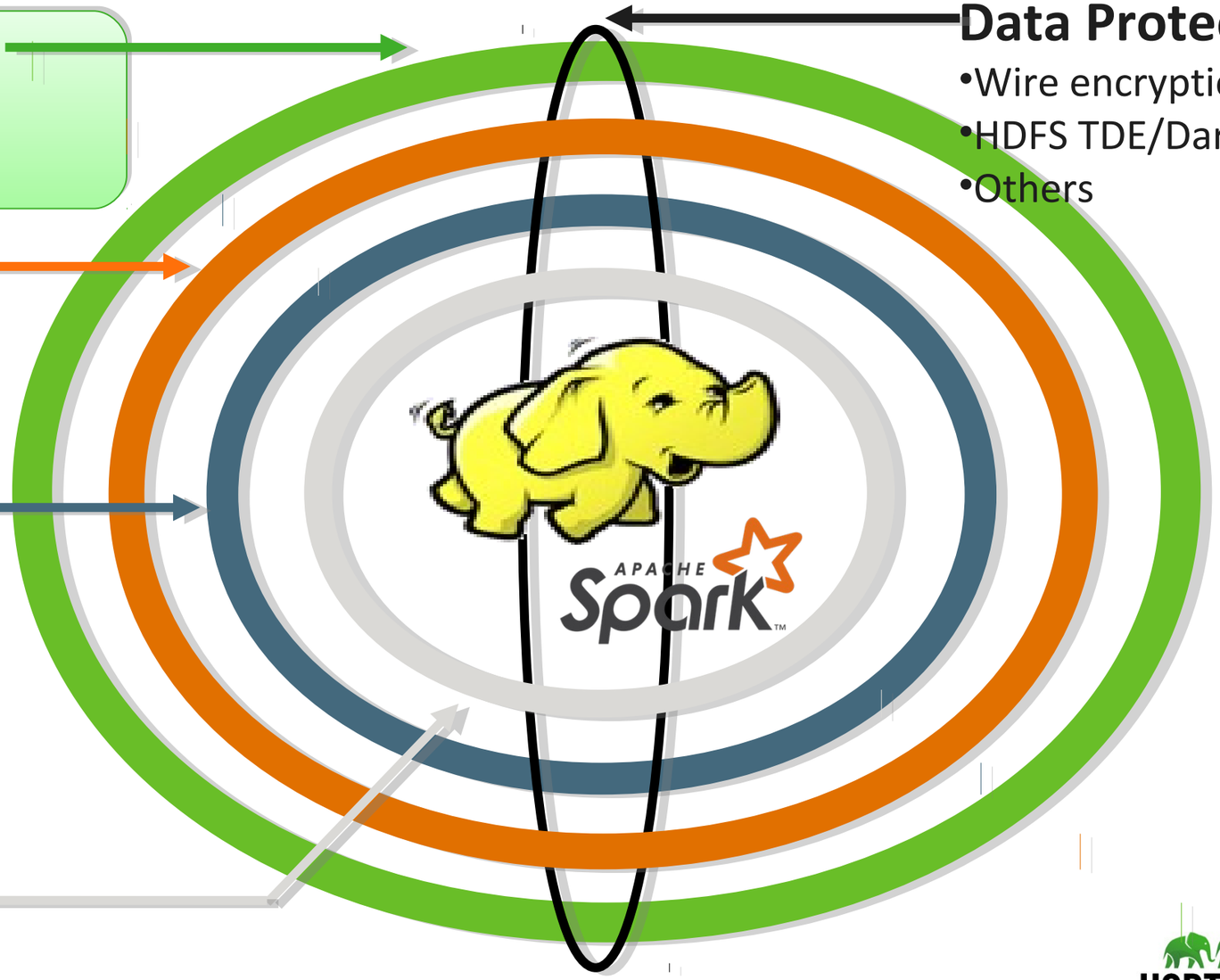
Authentication

- Kerberos
- Knox (Other Gateways)

Authorization

- Apache Ranger/Sentry
- HDFS Permissions
- HDFS ACLs
- YARN ACL

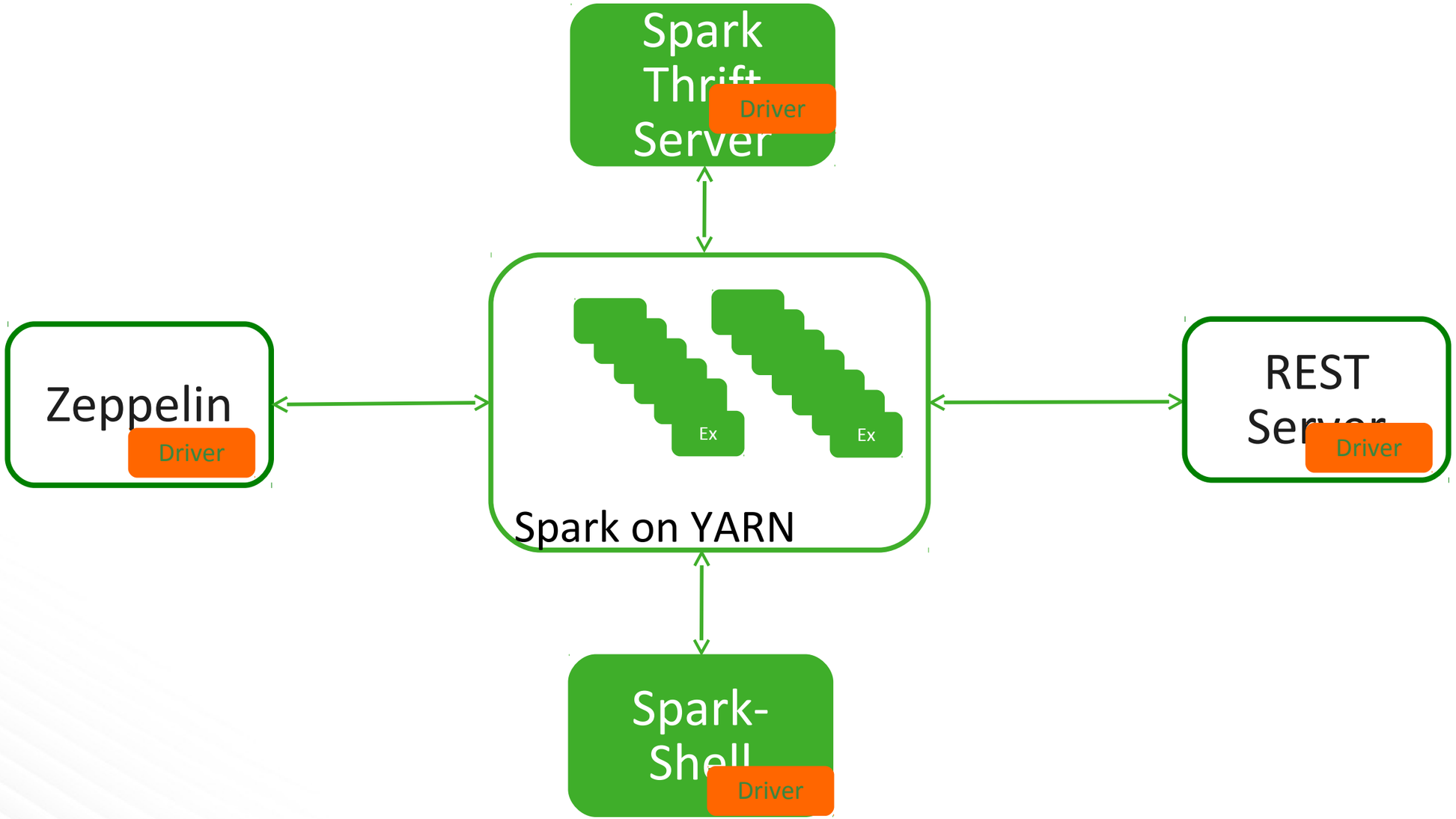
OS Security



Data Protection

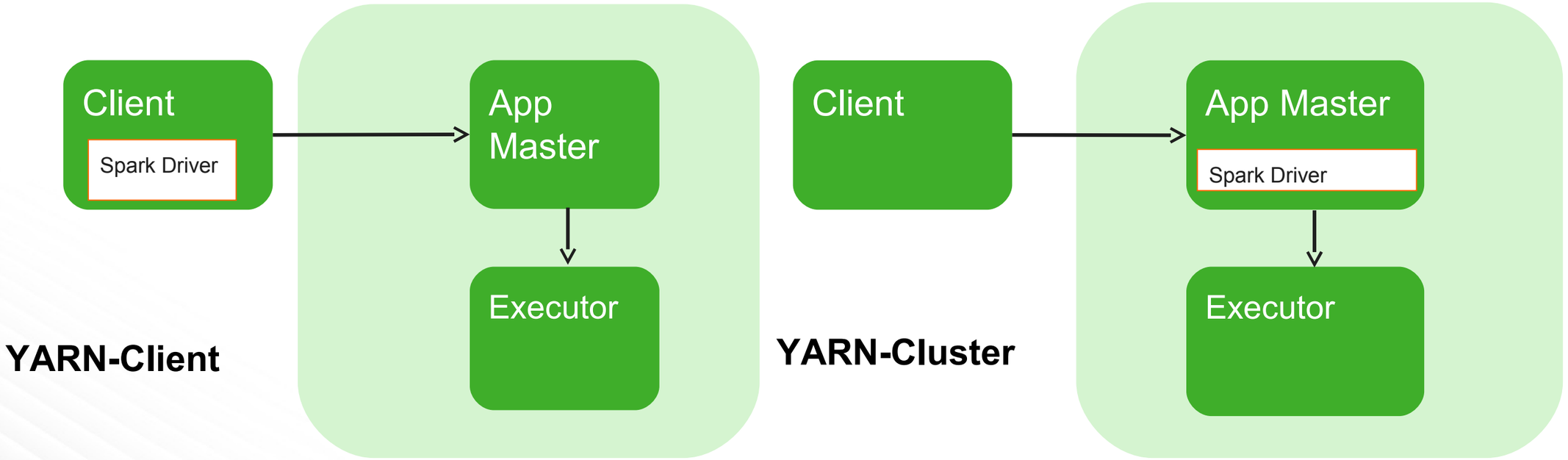
- Wire encryption
- HDFS TDE/Dare
- Others

Interacting with Spark

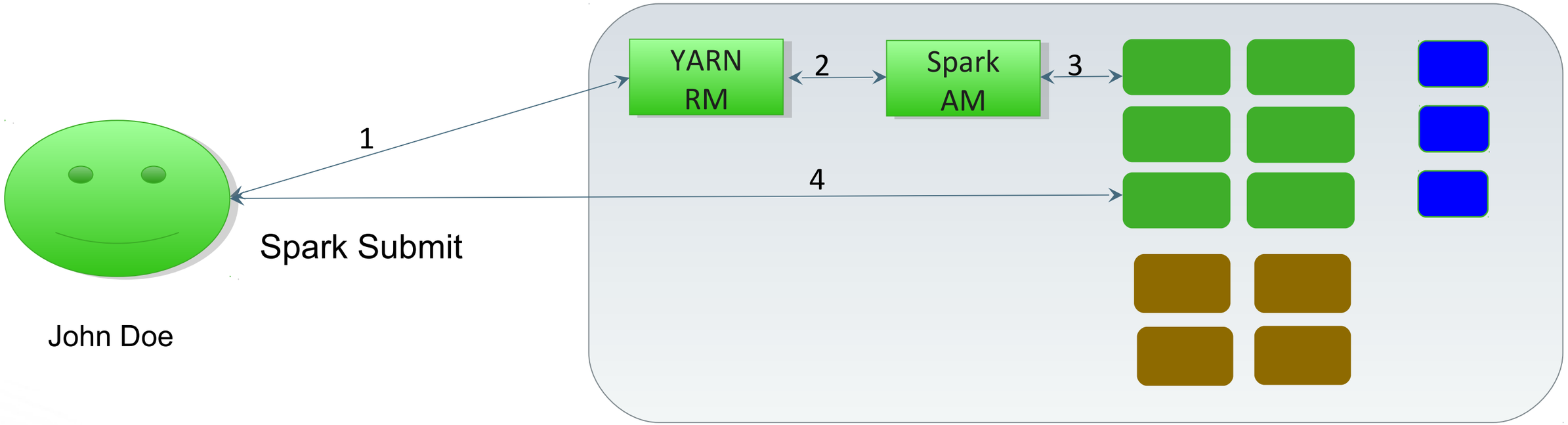


Context: Spark Deployment Modes

- Spark on YARN
  - Spark driver (SparkContext) in YARN AM(yarn-cluster)
  - Spark driver (SparkContext) in local (yarn-client):
    - Spark Shell & Spark Thrift Server runs in yarn-client only



Spark on YARN



Spark – Security – Four Pillars

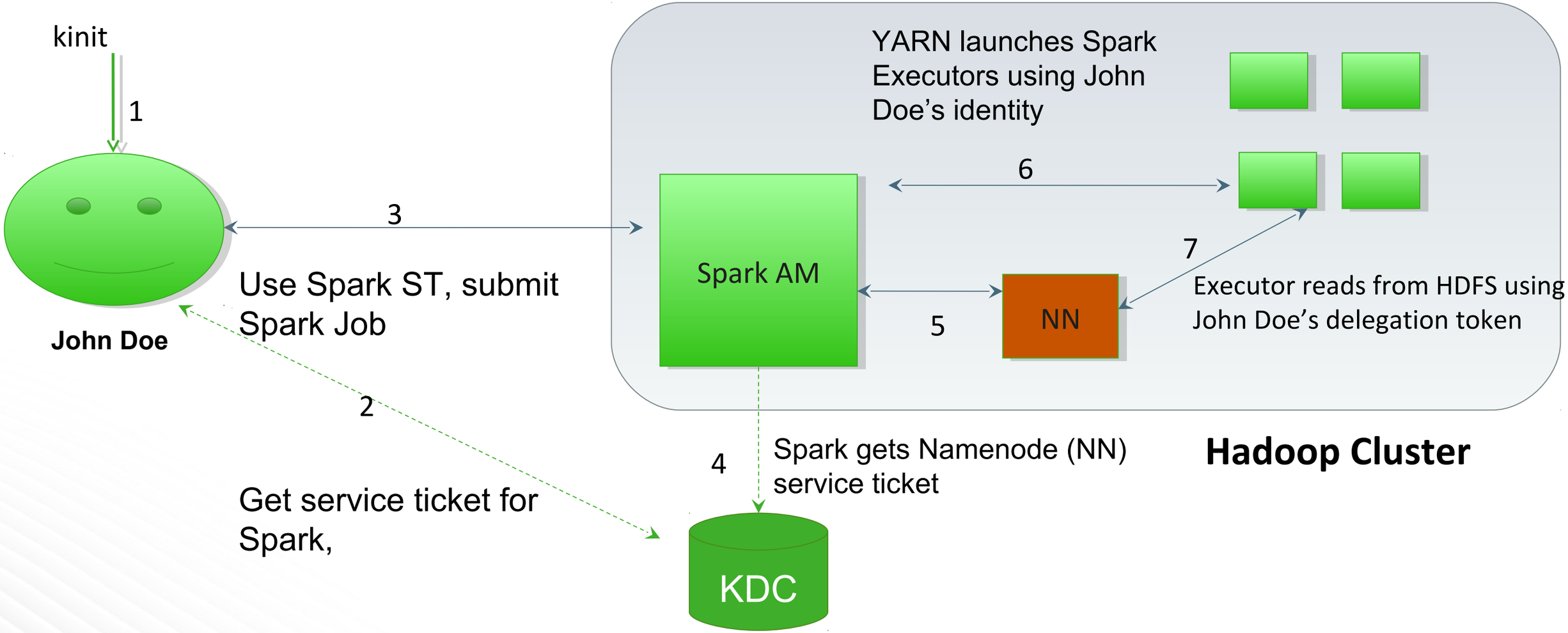
- Authentication
- Authorization
- Audit
- Encryption

Ensure network is secure

Spark leverages Kerberos on YARN

HORTONWORKS

Kerberos authentication within Spark



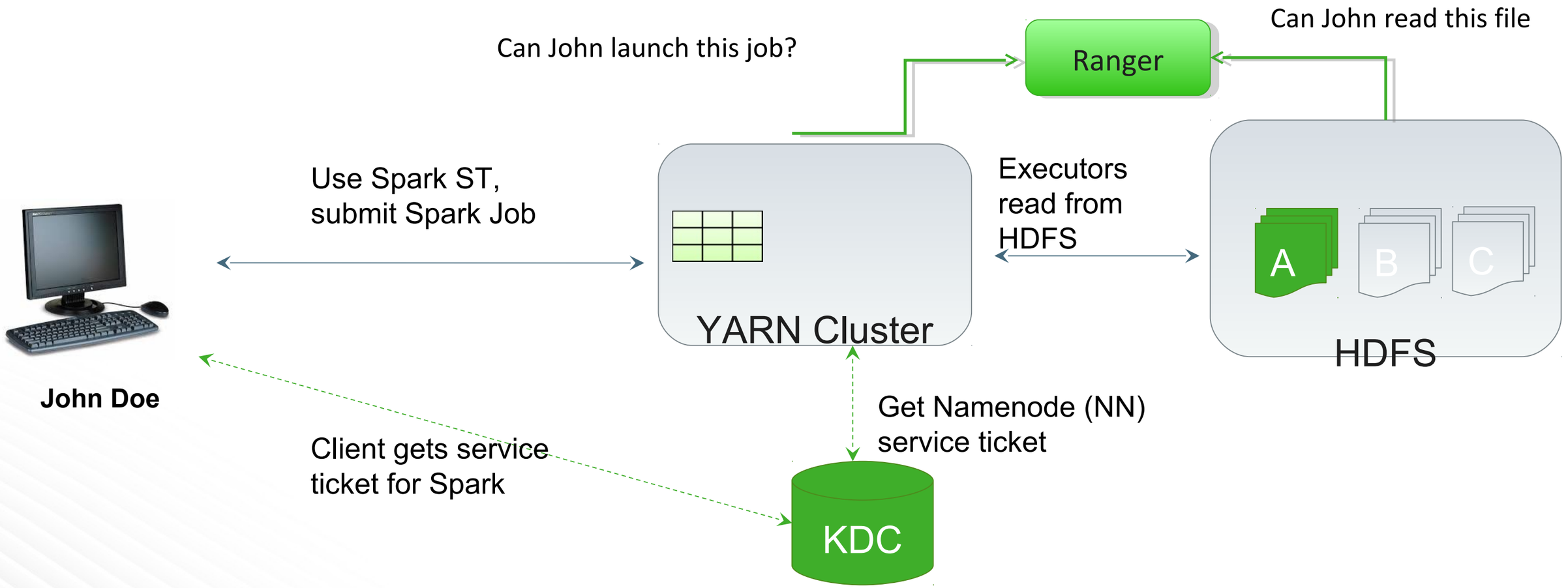
Spark – Kerberos - Example

```
kinit -kt /etc/security/keytabs/johndoe.keytab johndoe@
EXAMPLE.COM
```

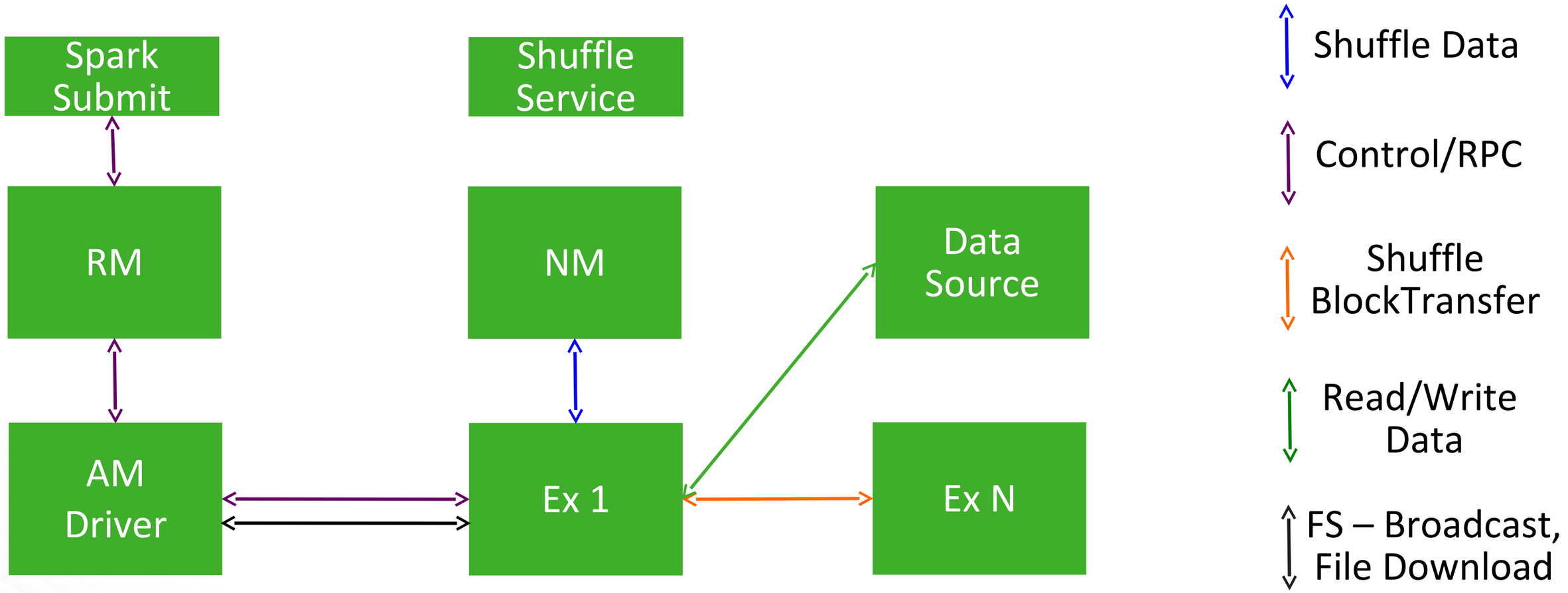
```
./bin/spark-submit --class org.apache.spark.examples.SparkPi
--master yarn-cluster  --num-executors 3 --driver-memory 512m
--executor-memory 512m   --executor-cores 1  lib/spark-
examples*.jar 10
```


HORTONWORKS

Spark – Authorization



Encryption: Spark – Communication Channels



Spark Communication Encryption Settings

Shuffle Data

NM > Ex leverages YARN based SSL

Control/RPC

spark.authenticate = true. Leverage YARN to distribute keys

Shuffle BlockTransfer

spark.authenticate.enableSaslEncryption= true

Read/Write Data

Depends on Data Source, For HDFS RPC (RC4 | 3DES) or SSL for WebHDFS

FS – Broadcast, File Download

spark.ssl.enabled = true

HORTONWORKS

Sharp Edges with Spark Security

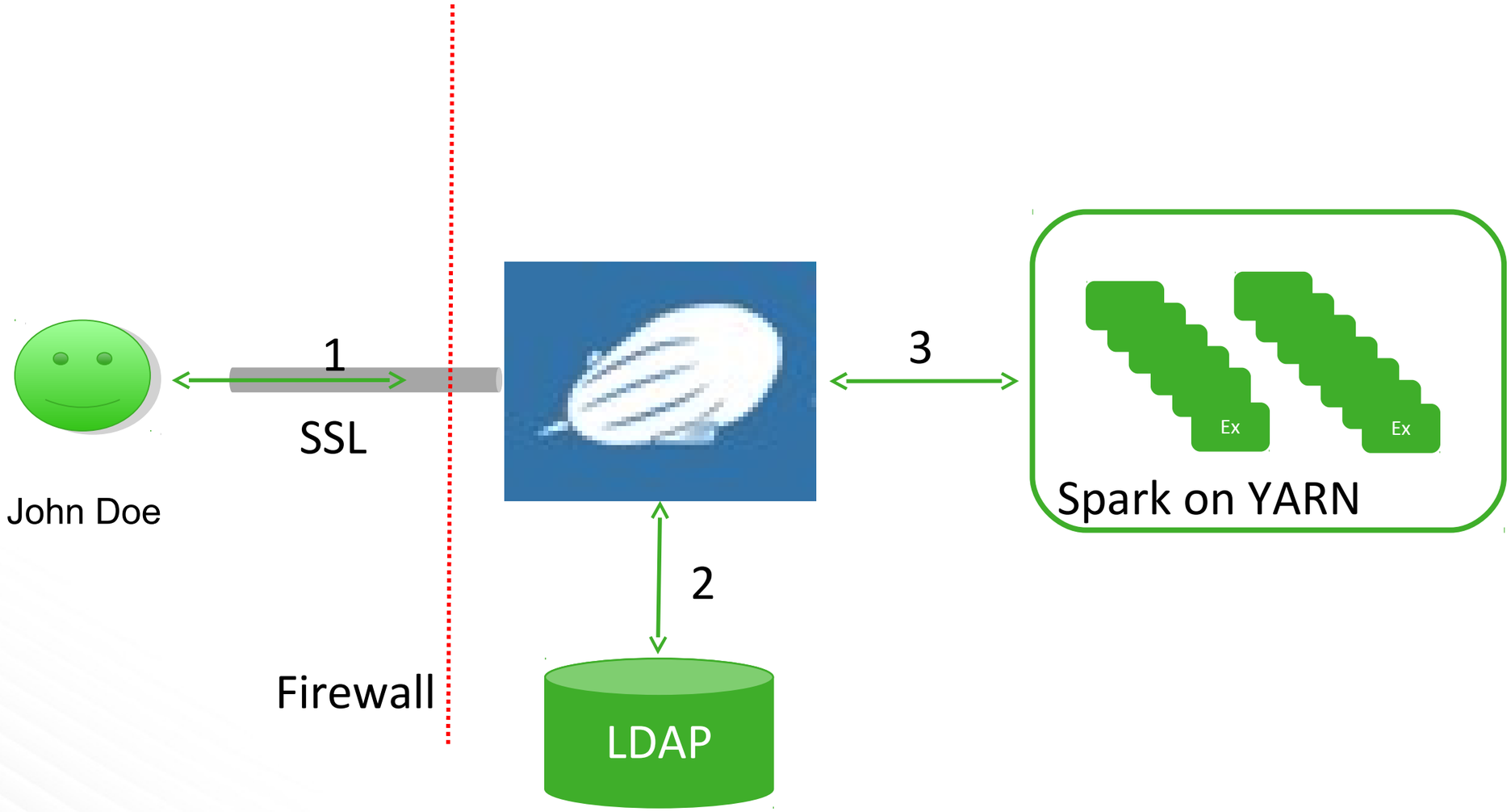
- SparkSQL – Only coarse grain access control today
- Client -> Spark Thrift Server > Spark Executors – No identity propagation on 2nd hop
  - Lowers security, forces STS to run as Hive user to read all data
  - Use SparkSQL via shell or programmatic API
  - https://issues.apache.org/jira/browse/SPARK-5159
- Spark Stream + Kafka + Kerberos
  - Issues fixed in HDP 2.4.x
  - No SSL support yet
- Spark Shuffle > Only SASL, no SSL support
- Spark Shuffle > No encryption for spill to disk or intermediate data

Fine grained Security to SparkSQL
http://bit.ly/2bLghGz
http://bit.ly/2bTX7Pm

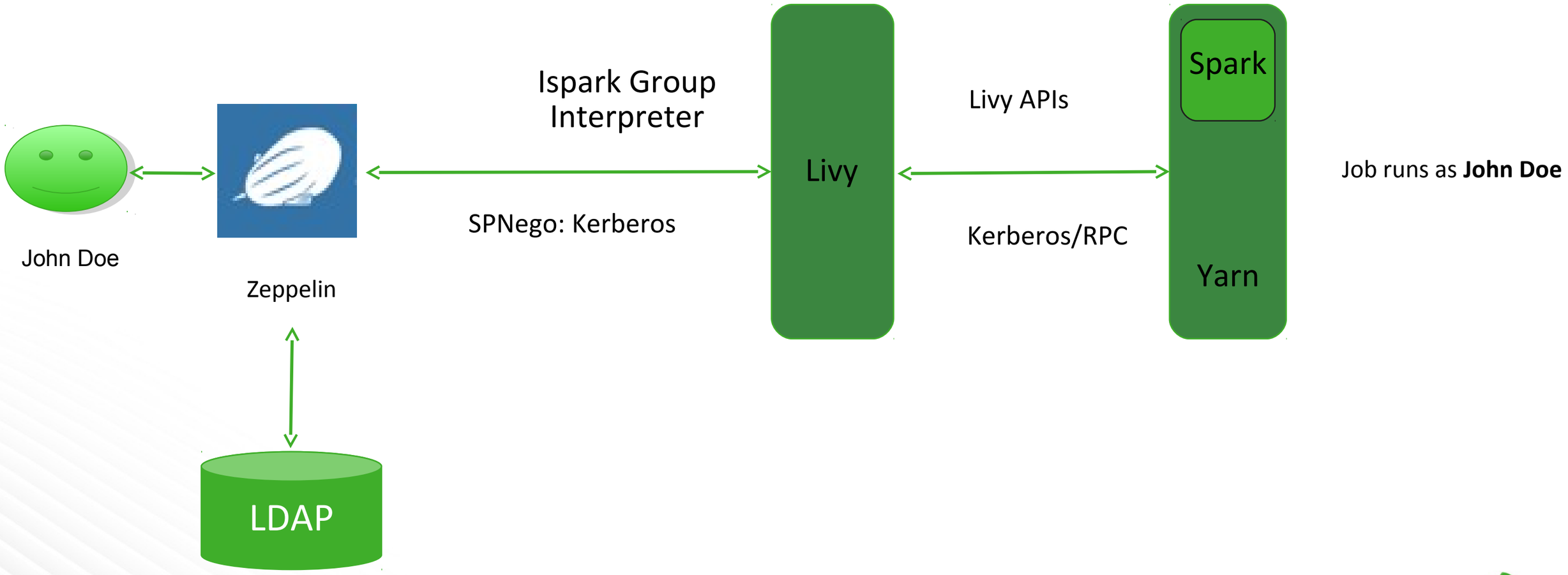
Apache Zeppelin Security

HORTONWORKS

Apache Zeppelin: Authentication + SSL



Zeppelin + Livy E2E Security



Apache Zeppelin: Authorization

Authorization in Zeppelin

- Note level authorization
  - Grant Permissions (Owner, Reader, Writer) to users/groups on Notes
  - LDAP Group integration
- Zeppelin UI Authorization
  - Allow only admins to configure interpreter
  - Configured in shiro.ini

```
[urls]
/api/interpreter/** = authc, roles[admin]

/api/configurations/** = authc, roles[admin]

/api/credential/** = authc, roles[admin]
```

Authorization at Data Level

- For Spark with Zeppelin > Livy > Spark
  - Identity Propagation Jobs run as End-User
- For Hive with Zeppelin > JDBC interpreter
- Shell Interpreter
  - Runs as end-user

HORTONWORKS

Apache Zeppelin: Credentials

Credentials in Zeppelin

- LDAP/AD account
  - Zeppelin leverages Hadoop Credential API
- Interpreter Credentials
  - Not solved yet
- Credentials

This is still an open issue

Apache Zeppelin: AD Authentication

Zeppelin leverages Apache Shiro for authentication/authorization

Configure Zeppelin to Authenticate users

1. /etc/zeppelin/conf/shiro.ini

```
[urls]
/api/version = anon
/** = authc
```


HORTONWORKS

Apache Zeppelin: AD Authentication

Skip step 1 if securing LDAP password is not an issue

Active Directory Authentication

1. Create an entry for AD credential
   - Zeppelin leverages Hadoop Credential API
   - >hadoop credential create
   - activeDirectoryRealm.systemPassword -provider jceks://etc/zeppelin/conf/credentials.jceks
   - chmod 400 with only Zeppelin process r/w access, no other user allowed accessCredentials

1. Configure Zeppelin to use AD

```
activeDirectoryRealm = org.apache.zeppelin.server.ActiveDirectoryGroupRealm
activeDirectoryRealm.systemUsername = CN=Administrator,CN=Users,DC=HWQE,DC=HORTONWORKS,DC=COM
#activeDirectoryRealm.systemPassword = Password1!
activeDirectoryRealm.hadoopSecurityCredentialPath = jceks://etc/zeppelin/conf/credentials.jceks
activeDirectoryRealm.searchBase = CN=Users,DC=HWQE,DC=HORTONWORKS,DC=COM
activeDirectoryRealm.url = ldap://ad-nano.qe.hortonworks.com:389
activeDirectoryRealm.groupRolesMap =
"CN=admin,OU=groups,DC=HWQE,DC=HORTONWORKS,DC=COM":"admin","CN=finance,OU=groups,DC=HWQE,DC=HORTONWORK
S,DC=COM":"finance","CN=zeppelin,OU=groups,DC=HWQE,DC=HORTONWORKS,DC=COM":"zeppelin”
activeDirectoryRealm.authorizationCachingEnabled = true
```

Spark Performance

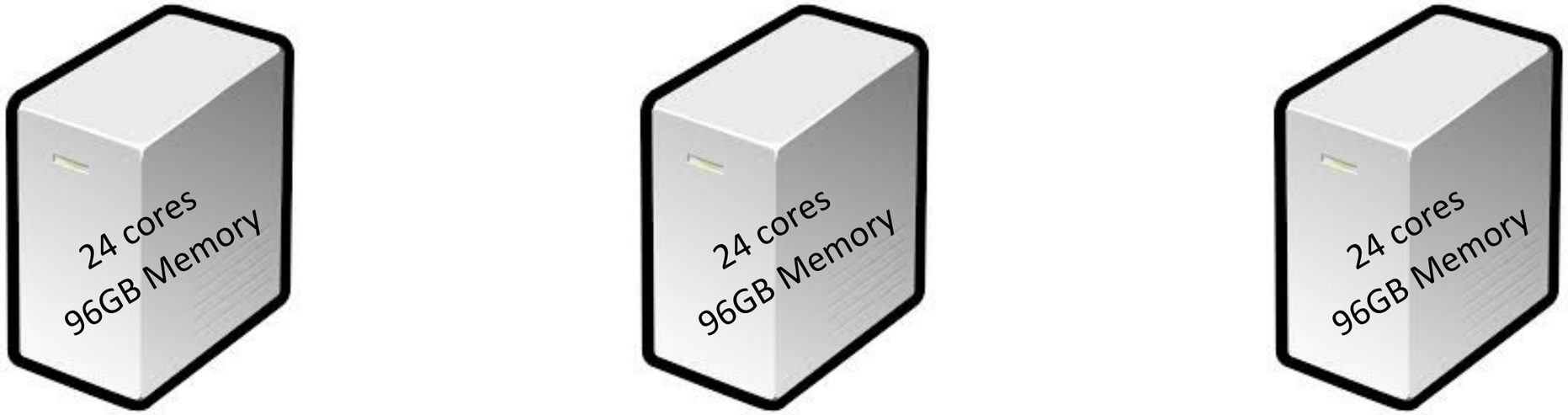
HORTONWORKS®

#1 Big or Small Executor ?

```
spark-submit --master yarn \
    --deploy-mode client \
    --num-executors ? \
    --executor-cores ? \
    --executor-memory ? \
    --class MySimpleApp \
    mySimpleApp.jar \
    arg1 arg2
```


HORTONWORKS

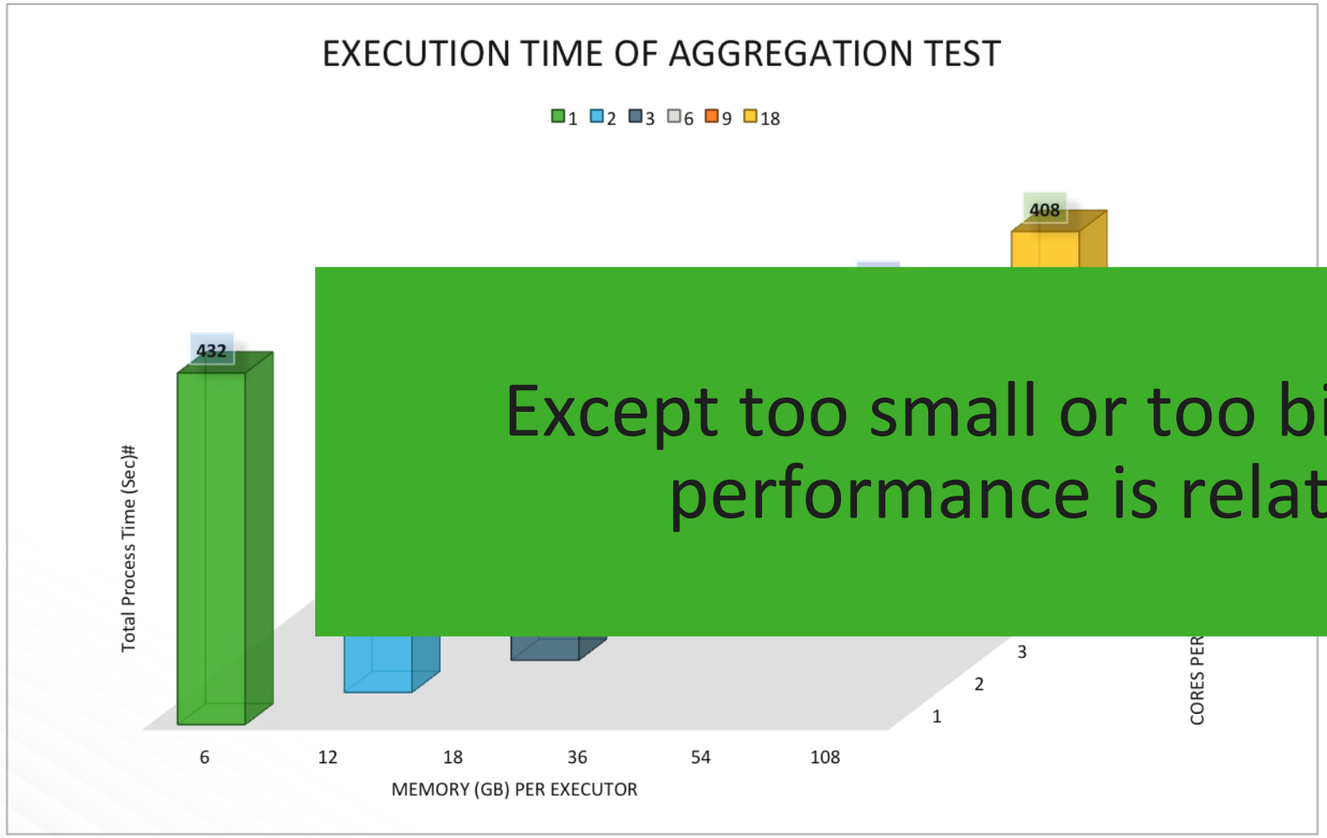
Show the Details



- Cluster resource: 72 cores + 288GB Memory

--executor-cores	--executor-memory	--num-executor
1	4GB	72
2	8GB	36
3	12GB	24
⋮	⋮	⋮
24	96GB	3

Benchmark with Different Combinations



Cores Per E#	Memory Per E (GB)	E Per Node
1	6	18
		9
		6
		3
		2
18	108	1

Except too small or too big executor, the performance is relatively close

The lower the better

18 cores 108GB memory per node

#E stands for Executor

Big Or Small Executor ?

- Avoid too small or too big executor.
  - Small executor will decrease CPU and memory efficiency.
  - Big executor will introduce heavy GC overhead.
- Usually 3 ~ 6 cores and 10 ~ 40 GB of memory per executor is a preferable choice.

Any Other Thing ?

- Executor memory != Container memory
- Container memory = executor memory + overhead memory (10% of executory memory)
- Leave some resources to os and other services

--executor-cores	--executor-memory	--num-executor
3	10GB ~~(12GB)~~	24
4	13GB ~~(16GB)~~	18
6	20GB ~~(24GB)~~	12

- Enable CPU scheduling if you want to constrain CPU usage[#]

#CPU scheduling -
http://hortonworks.com/blog/managing-cpu-resources-in-

HORTONWORKS

Multi-tenancy for Spark

Cluster resource Utilization

- Leverage YARN queues
  - Set user quotas
  - Set default yarn-queue in spark-defaults
  - User can override for each job
- Leverage Dynamic Resource Allocation
  - Specify range of executors a job uses
  - This needs shuffle service to be used

Thank You

Vinay Shukla
@neomythos