Hadoop安全体系介绍以及实践分享

杨威@明略数据


Geekbang.
极客邦科技
全球领先的技术人学习和交流平台
InfoQ
专注中高端技术
人员的社区媒体
EGO EXTRA GEEKS' ORGANIZATION NETWORKS
高端技术人员
学习型社交网络
StuQ
实践驱动的IT职业
学习和服务平台
扫我，码上开启新世界
Geekbang.
InfoQ | EGO NETWORKS | StuQ

InfoQ

促进软件开发领域知识与创新的传播

风险

- 外部风险
  - 恶意用户入侵 – 密码破解
- 内部风险
  - 内部用户 – 违规操作、失误操作、越权访问
- 数据风险
  - 数据被泄漏、篡改、删除
- 服务风险
  - 服务非法使用、资源超限使用、系统可用性风险

Hadoop平台“安全”的两个方面

- 数据安全
  - 保障平台上的数据不被恶意用户所窃取、破坏和泄漏
  - 保障平台上的服务不被恶意操作和使用
  - 网络安全、主机安全、服务安全、数据安全
- 运维安全
  - 即工业界常提的安全生产
  - 提升平台的稳定性和可靠性，保障服务的正常运行
  - 权限独立管理、资源统一管理、服务热备双活

Hadoop 之 数据安全

- 缺乏安全配置的Hadoop平台有哪些隐患？
  - SIMPLE的身份验证机制
    - nobody都可以冒充superuser
  - 基于Linux用户组信息的文件访问控制
    - 本地权限可被恶意用户利用
  - 不经授权的数据访问和粗粒度的数据访问控制
    - 不经授权的获取关键数据
  - 不设防的底层文件存储
    - 偷走文件即偷走了数据内容

不设防的身份验证

- SIMPLE的身份验证机制
  - 认为系统内的所有节点都是可信赖的
  - 认为用户的身份都是他自己所宣称的
- 后果
  - 任意机器可伪装成为集群的一个节点加入
    - 伪装成DataNode ,NodeManager甚至NameNode
    - 破坏集群的正常运转
  - 集群任意节点的任意帐号可伪装成超级用户或者其他用户
    - 任意偷取、破坏数据
    - 任意提交任务占用集群资源
    - 等等你能想到的一切最坏情况

基于Linux用户组信息的文件访问控制

- 模拟Linux的文件系统权限：RWX * (Owner, Group, Other)
- 可利用本地权限获得平台权限
  - 开给某个应用的跳板机上，假设A, B 两个帐号分属不同组，都可以访问Hadoop
  - A将其文件设置为了Group可读
  - 假如B拿到了该跳板机的Root权限，即可把自己的属组改为和A相同，即可访问A的文件

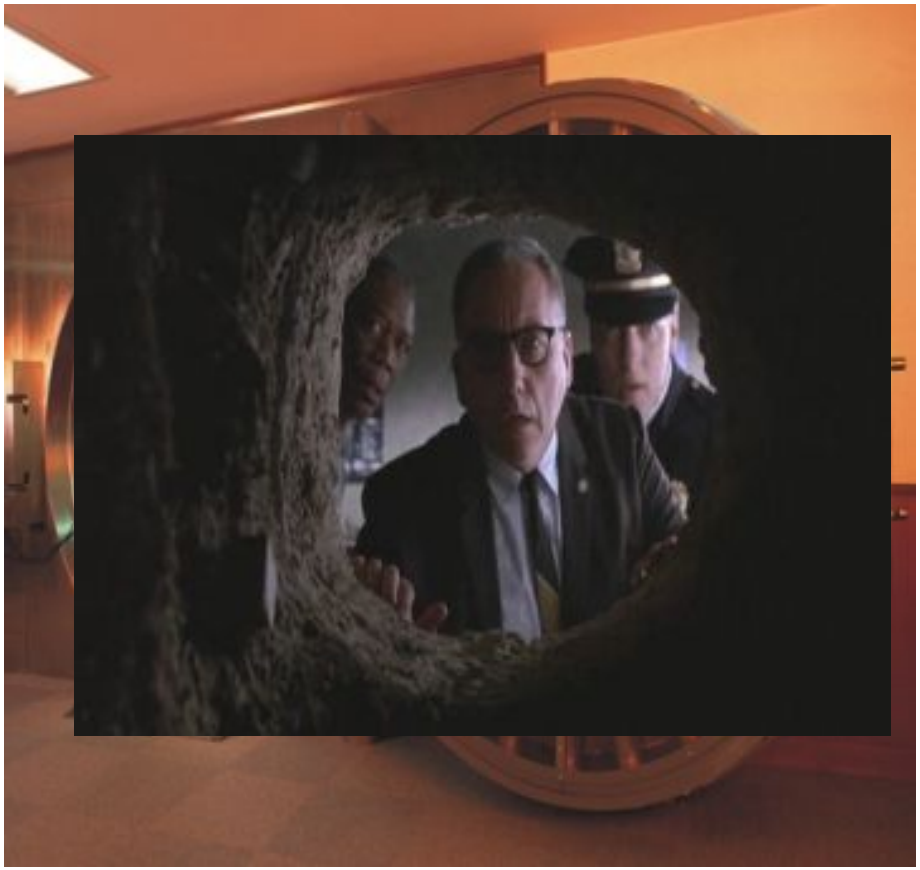
不经授权的数据访问

- Hive的数据表通常没有设置访问权限
- 表级别的权限控制不能限制敏感数据的访问
- 能

ID	Name	Cardbin	Address	Order	Merchant	Transaction
1	张三	8888-9999-6666-1111	北京	123	家乐福	100.00
2	李四	8888-9999-6666-2222	北京	321	沃尔玛	10000.00
3	Tom	8888-9999-6666-3333	深圳	222	海底捞	330.01
4	Cat	8888-9999-6666-4444	深圳	456	香格里拉	299.99

不设防的底层文件存储

- 再好的上层身份认证、权限控制系统都抵不住底层文件直接泄漏
  - HIVE中的数据文件以明文或者近似明文的方式存储
  - 恶意用户可直接copy走文件来绕过上层的权限管理系统
  - 恶意用户甚至可直接copy走DataNode上的DataBlock来窃取数据



Hadoop平台之安全体系

操作安全	安全审计 安全运维
服务安全	服务访问授权 服务权限分割
数据安全	数据传输加密 底层数据加密 数据访问授权
周边安全	网络安全、系统安全 身份系统、安全认证

Hadoop平台之安全体系

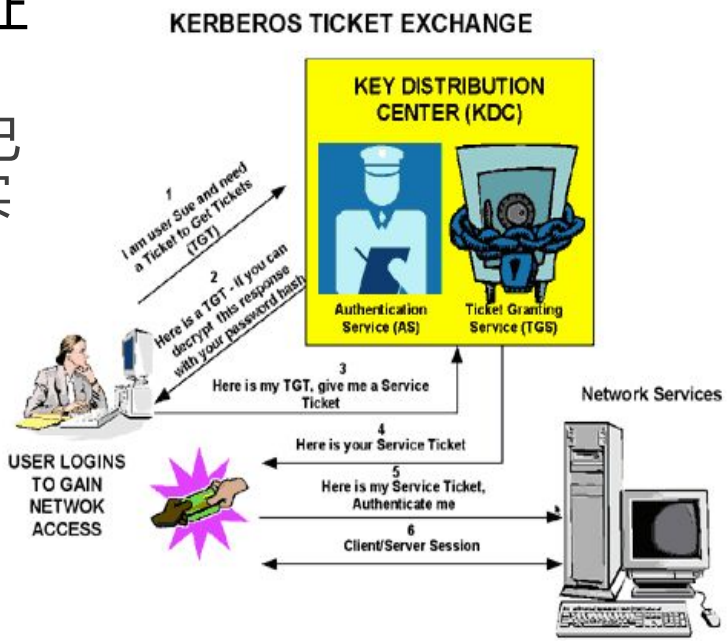
- 身份认证
  - Kerberos
- 身份管理
  - LDAP
- 授权访问
  - 服务授权访问：Policy File
  - 文件授权访问：HDFS ACL
  - 数据授权访问：SENTRY
- 安全审计
  - 审计日志
- 数据加密
  - HDFS crypto
  - Hadoop Key Management Server（KMS）
- 传输加密
- 通信加密
- REST加密与认证

身份认证

- 要证明你是你
  - 你有打开锁的钥匙
  - 你能对上暗号

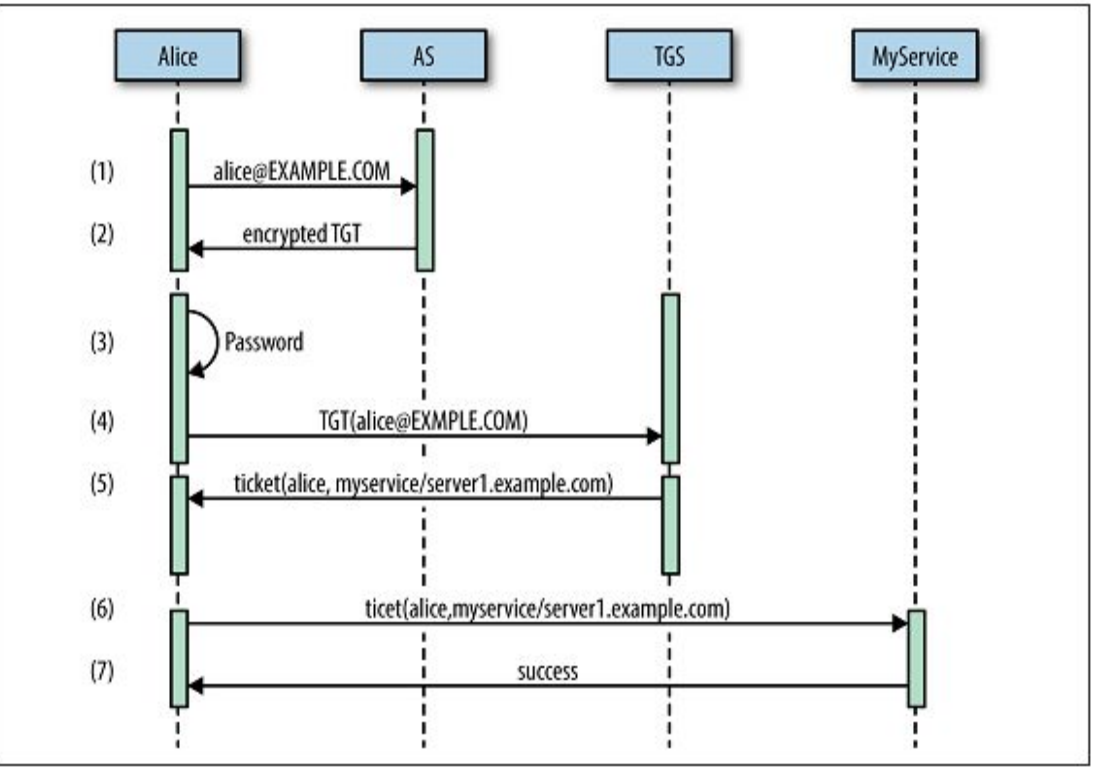
身份认证

- 基于Kerberos的服务与用户的身份认证机制
  - 80年代中期产生，93年加入RFC协议，已成为网络中可信任的第三认证服务的事实标准
  - Principal, Realm
  - KDC(Key distribution center), AS(Authentication service), TGS(Ticket-granting service)



- 基于口令/密钥认证
- 一次登录，全局认证，适用于分布式系统

身份认证



当前需要进行Keberos认证的组件

- Zookeeper
- HDFS
- YARN
- HBase
- Hive
- Impala
- Oozie
- Sqoop2
- Flume
- HUE
- Kafka(on the way)

身份认证

- **Kerberos登录方式**
  - 命令式

```
[hdfs@d196 ~]$ kinit -k -t .secret/hdfs.keytab hdfs/196.mininglamp.com
```

  - 程序代码
    - 帐号密码

```
LoginContext context = new LoginContext("loginContext", null,
    new UsernamePasswordCallbackHandler(principal, password), new KerberosConfiguration());
context.login();
return context.getSubject();
```

    - Keytab文件

```
ugi = UserGroupInformation.loginUserFromKeytabAndReturnUGI(princ, keyTab);
```

独立身份系统

- 与Linux用户组信息无关的用户管理
- 所有的权限控制基于此用户管理系统统一进行

- 基于Kerberos的用户身份
  - Kerberos Pricipal -> Hadoop User
  - Etlman/hadoop@mininglamp.com

```
<property>
  <name>hadoop.security.auth_to_local</name>
    <value>
      RULE:[1:$1.$0]
      RULE:[2:$1@$0]
      DEFAULT
    </value>
</property>
```

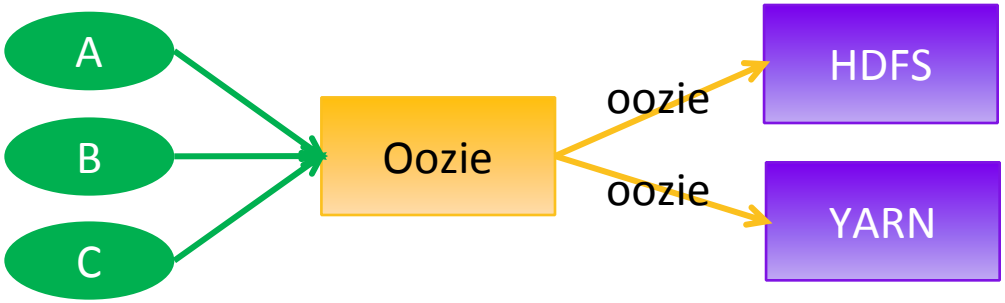
独立身份系统

- 基于LDAP独立管理的用户组信息
  - Hadoop User -> Hadoop Group

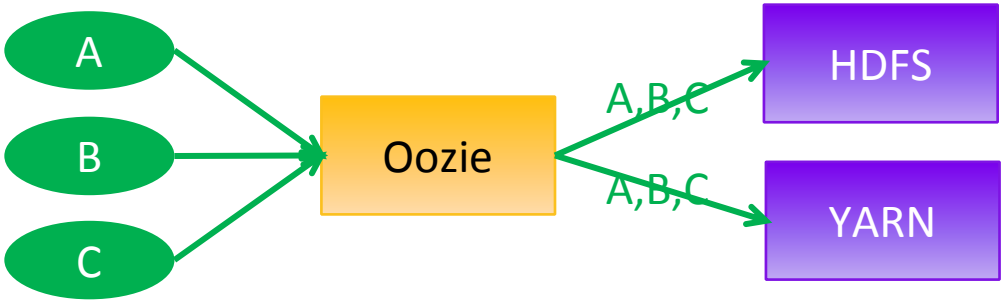
```
<property>
  <name>hadoop.security.group.mapping</name>
  <value>org.apache.hadoop.security.LdapGroupsMapping</value>
</property>
<property>
  <name>hadoop.security.group.mapping.ldap.url</name>
  <value>ldap://ldap.mininglamp.com</value>
</property>
<property>
  <name>hadoop.security.group.mapping.ldap.bind.user</name>
  <value>Hadoop@ldap.mininglamp.com</value>
</property>
<property>
  <name>hadoop.security.group.mapping.ldap.bind.password</name>
  <value>password</value>
</property>
```

  - hadoop.security.groups.cache.secs
  - Hadoop.security.group.mapping.ldap.bind.passwordfile

Impersonation - Proxy User




```
<property>
  <name>hadoop.proxyuser.oozie.hosts</name>
  <value>*</value>
</property>
<property>
  <name>hadoop.proxyuser.oozie.groups</name>
  <value>*</value>
</property>
```

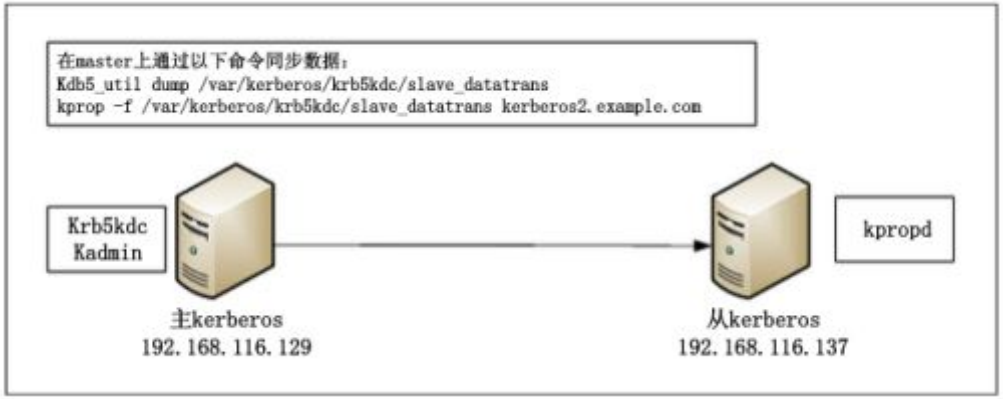


可动态生效，无需重启服务

bin/hdfs dfsadmin –refreshSuperUserGroupsConfiguration
bin/yarn rmadmin –refreshSuperUserGroupsConfiguration

一些建议

- Kerberos备份



- Kerberos应配合LinuxContainerExecutor一起使用
  - 不使用LinuxContainerExecutor，所有的container会以yarn用户执行， container之间可以互相访问本地目录
  - 每个机器上运行的任务是以提交任务的用户UID执行， container之间不可互相访问数据
- LinuxContainerExcecutor可考虑配合cgroups
  - 严格的cpu、内存使用限制

授权访问

- 通过了身份认证，只证明了你是你
- 想要使用平台的服务，还需进行授权访问
  - 获得你的详细信息，比如你是谁，你是哪个组的
  - 服务授权，授权你使用哪个服务的哪些功能
  - 数据授权，授权你访问哪些文件、目录，哪个表的哪些行、哪些列

服务授权访问

- 限制用户使用哪些服务的哪些功能
- 覆盖了主要的存储(HDFS、Hbase)和计算资源(Yarn ,Oozie)的使用
- 基于用户和组的访问控制列表
- 细粒度的访问权限控制
  - 控制能否访问NameNode/DataNode
  - 能否提交任务/杀死任务/查看任务状态
  - Hbase表查询/表管理操作等
  - Oozie任务提交、查看操作等
- 服务访问控制动态生效

一些建议

- HDFS服务的授权访问
  - 关闭HDFS超级用户的权限
  - 增加集群管理员组、HDFS文件系统超级用户组、HDFS文件系统用户组
  - 审计需求
  - 便于管理

```
<property>
  <name>dfs.cluster.administrators</name>
  <value>hdfs hadoop-admins</value>
</property>
```

一些建议

- YARN服务的授权访问
  - 增加YARN集群管理员组、YARN Scheduler管理员组、YARN用户组
  - 增加job historyserver的管理员组
  - 要求用户提交任务时指定队列

```
<property>
  <name>yarn.resourcemanager.scheduler.class</name>
  <value>
    org.apache.hadoop.yarn.server.resourcemanager.scheduler.fair.FairScheduler
  </value>
</property>
<property>
  <name>yarn.scheduler.fair.user-as-default-queue</name>
  <value>false</value>
</property>
<property>
  <name>yarn.scheduler.fair.allow-undeclared-pools</name>
  <value>false</value>
</property>
```

```
<property>
  <name>mapreduce.jobhistory.admin.acl</name>
  <value>mapred hadoop-admins</value>
</property>
```

数据授权访问

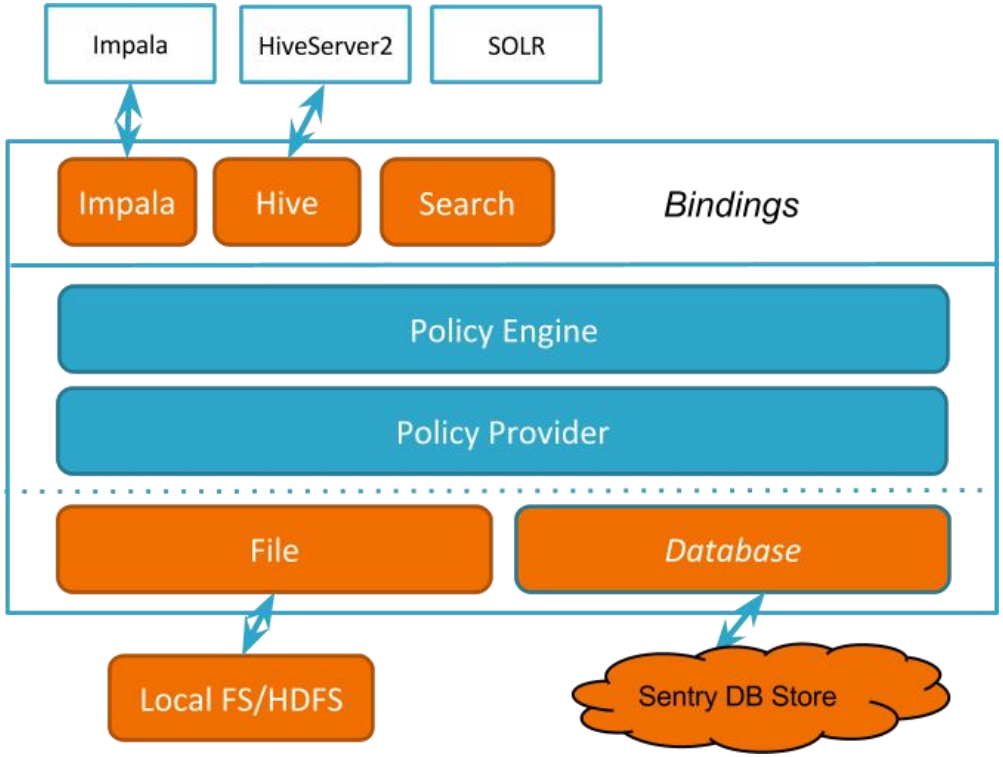
- HDFS ACL

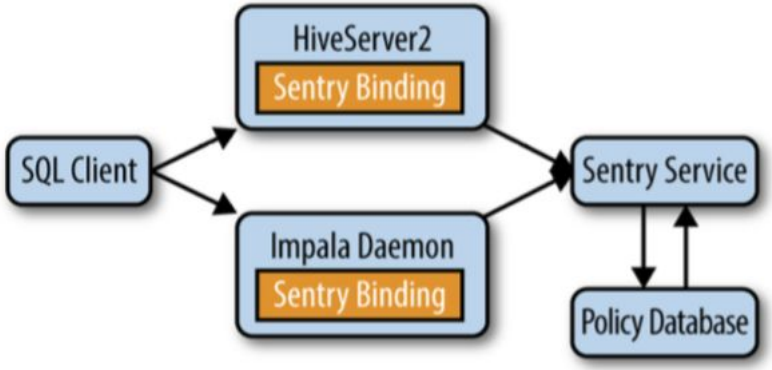
unnamed user (file owner)	文件的拥有者
unnamed group (file group)	文件的所属组
named user	除了文件的拥有者和拥有组之外，的其它用户
named group	除了文件的拥有者和拥有组之外，的其它用户
mask	权限掩码，用于过滤named user和named group的权限

```
[hadoop@master ~]$ hdfs dfs -getfacl /input/acl        【初始权限】
# file: /input/acl
# owner: hadoop
# group: supergroup
user::rwx
group::r-x
mask::r-x
other::r-x
[hadoop@master ~]$ hdfs dfs -setfacl -m user:mapred:rwx /input/acl
[hadoop@master ~]$ hdfs dfs -getfacl /input/acl            【mapred用户拥有rwx权限，但mask为r-x，则
mask自动改为rwx】
# file: /input/acl
# owner: hadoop
# group: supergroup
user::rwx
user:mapred:rwx
group::r-x
mask::rwx
other::r-x
```

数据授权访问

- Hive、Impala – Sentry
  - Apache Sentry 是一个加强的细粒度的基于角色的授权系统，针对存储在 Hadoop 集群中的数据和元数据。





数据授权访问

- Hive、Impala – Sentry
  - Apache Sentry 是一个加强的细粒度的基于角色的授权系统，针对存储在 Hadoop 集群中的数据和元数据。

	Hiveserver2	Sentry
用户/组/角色	1. 权限可以被授予给用户，也可以授予给角色； 2. 用户被指定属于一个或者多个角色； 3. 存在两个默认角色：Public和Admin：可以对Public角色授权，使得所有用户均具有该权限；在配置文件中指明属于Admin角色的	1. 权限只能被授予给角色 2. 角色被指定属于一个或者个多组； 3. 可以设定一个Sentry管理员所属的组：所有属于管理员组的用户都具有管理员的能力；
授权对象	数据库/表/视图	服务器/数据库/表/视图/列/URI
授权级别	SELECT, INSERT, UPDATE, DELETE, ALL	SELECT, INSERT, ALL

数据授权访问

Acre是明略数据研发的在Hive、Impala、Spark SQL上支持行列（Cell）级别访问控制的组件

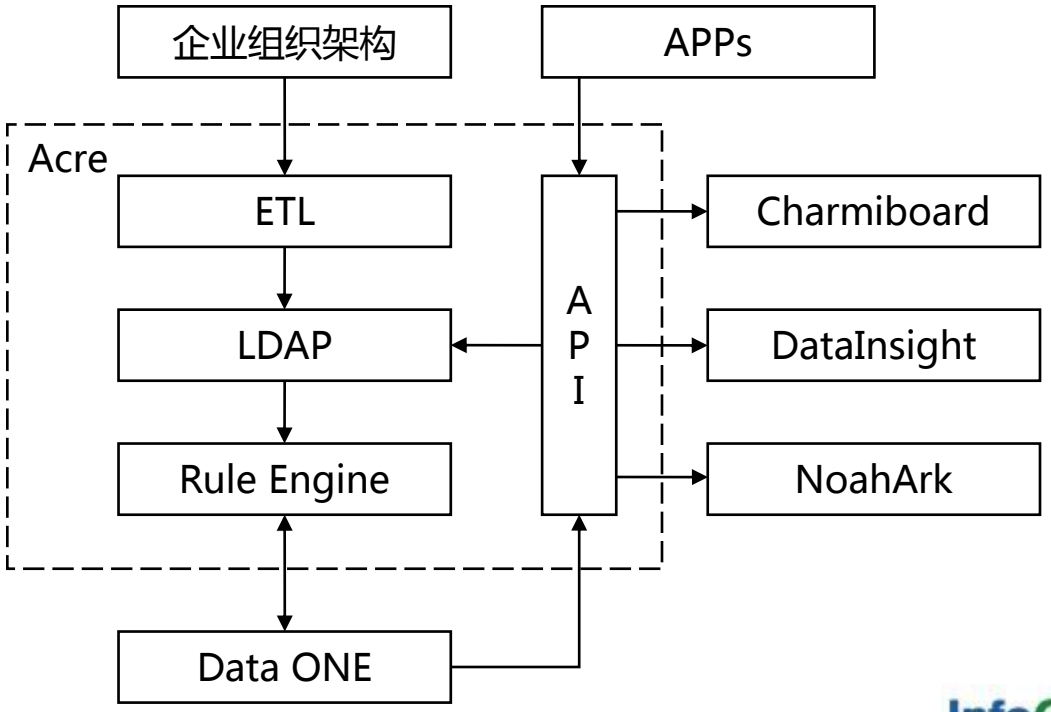
特性

• 支持基于ACL和RBAC的混合授权模型 (Note: 可以限制开发人员)

•支持Hive、Impala、Spark的统一授权管理

• 支持行列（Cell）基本访问控制

• 基于访问过滤的实现无需修改原数据库/表结构



行为审计

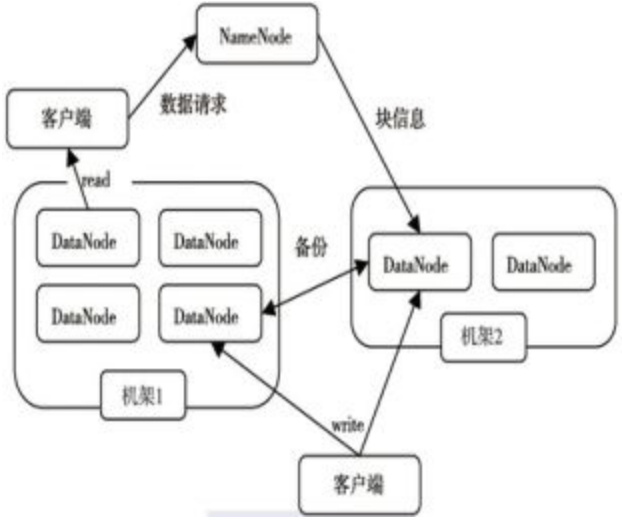
- 凡做过必留下痕迹
- 管理操作审计
  - 服务操作审计，包括安装、配置、卸载、启停、管理等操作
  - 机器操作审计，包含安装、卸载、管理等
- 服务访问审计
  - HDFS
  - Hbase
  - Hive
  - YARN

数据加密

- 偷到也白给
- 基于HDFS crypto提供静态加密区（目录）
- 基于 Hadoop KMS (Key Management Server)提供加密密钥管理
- 对应用透明的文件系统层加密
  - Hive、MR、Hbase、HDFS操作等均透明
- 超级用户也无法获取解密数据
  - 只有通过Kerberos认证的数据区拥有者，且通过HDFS的权限认证，方可访问数据

通信加密

- RPC通信
  - hadoop.rpc.protection = privacy
- 数据传输
  - dfs.encrypt.data.transfer = true“
  - dfs.encrypt.data.transfer.algorith =3des
  - dfs.encrypt.data.transfer.cipher.suites = AES
- 安全DataNode
  - 基于SASL的身份认证和数据加密机制
    - SASL可以认为是认证+加密两部分组成
    - 通过Kerberos进行身份认证
    - 通过Digest-Md5进行数据加密

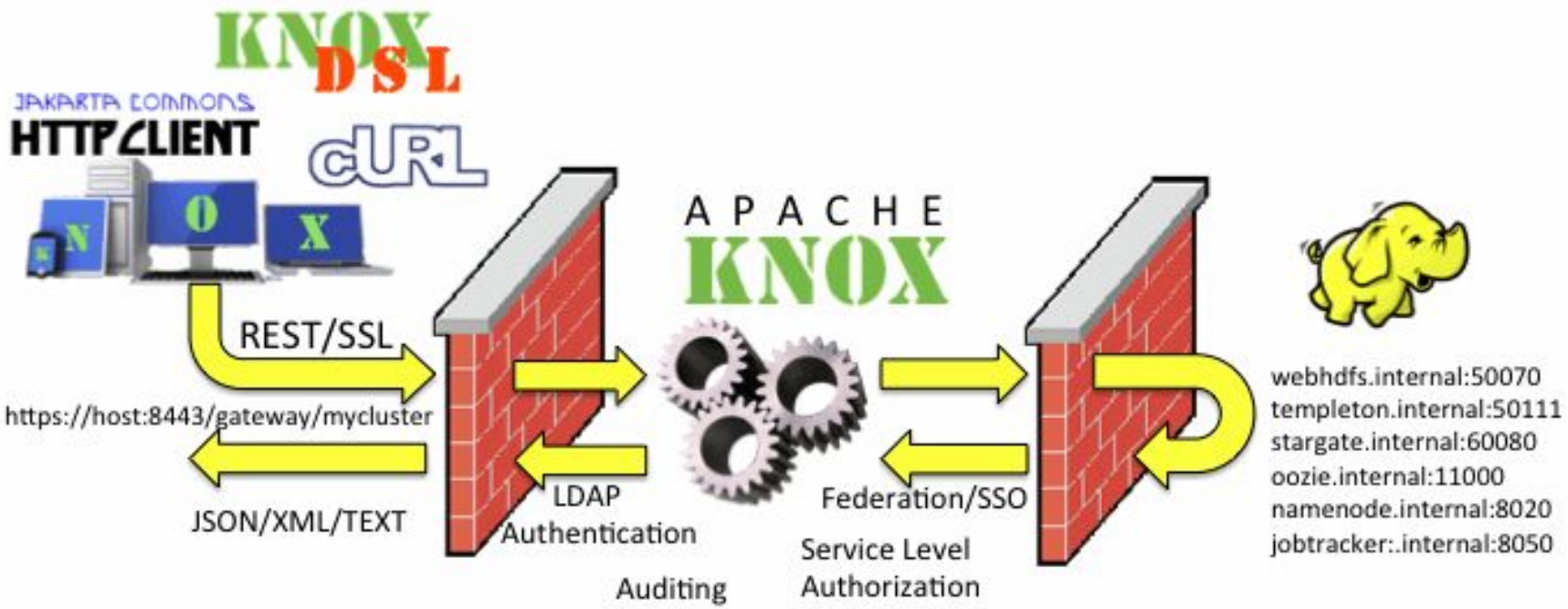


接口保护

- API接口
  - Kerberos认证
  - Hadoop.rpc.protection
- REST接口
  - Httpfs - HTTPS
  - KNox

Advantage	Description
Simplified access	Entend Hadoop's REST/HTTP services by encapsulating Kerberos within the cluster
Enhanced security	Expose Hadoop's REST/HTTP services without revealing network details, with SSL provided out of box
Centralized control	Centrally enforce REST API security and route requests to multiple Hadoop clusters
Enterprise integration	Support LDAP, Active Directory, SSO, SAML and other authentication systems

KNox

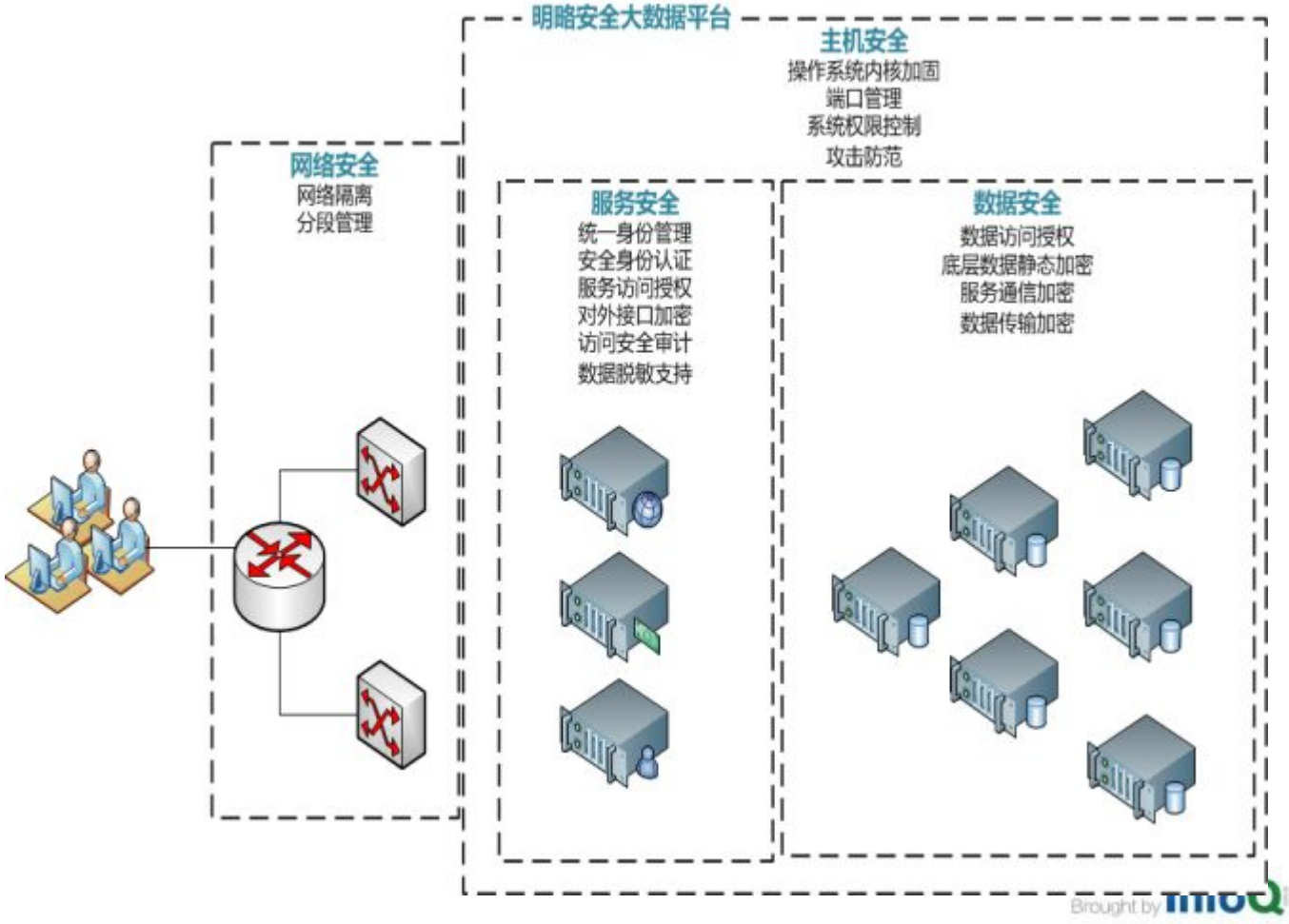

KNOX
DSL
JAKARTA COMMONS
HTTPCLIENT
cURL
REST/SSL
https://host:8443/gateway/mycluster
JSON/XML/TEXT
APACHE
KNOX
LDAP
Authentication
Auditing
Federation/SSO
Service Level
Authorization
webhdfs.internal:50070
templeton.internal:50111
stargate.internal:60080
oozie.internal:11000
namenode.internal:8020
jobtracker:.internal:8050

Web Console

- NameNode, ResourceManager Web 界面
  - 支持HTTP　SPNEGO验证机制
  - SPNEGO底层基于Kerberos，是使用标准HTTP协议为WEB应用进行Kerberos扩展的一种机制

平台安全的更多层次

- 安全性
  - 网络安全
  - 主机安全
  - 服务安全
  - 数据安全



Hadoop平台“安全”的两个方面

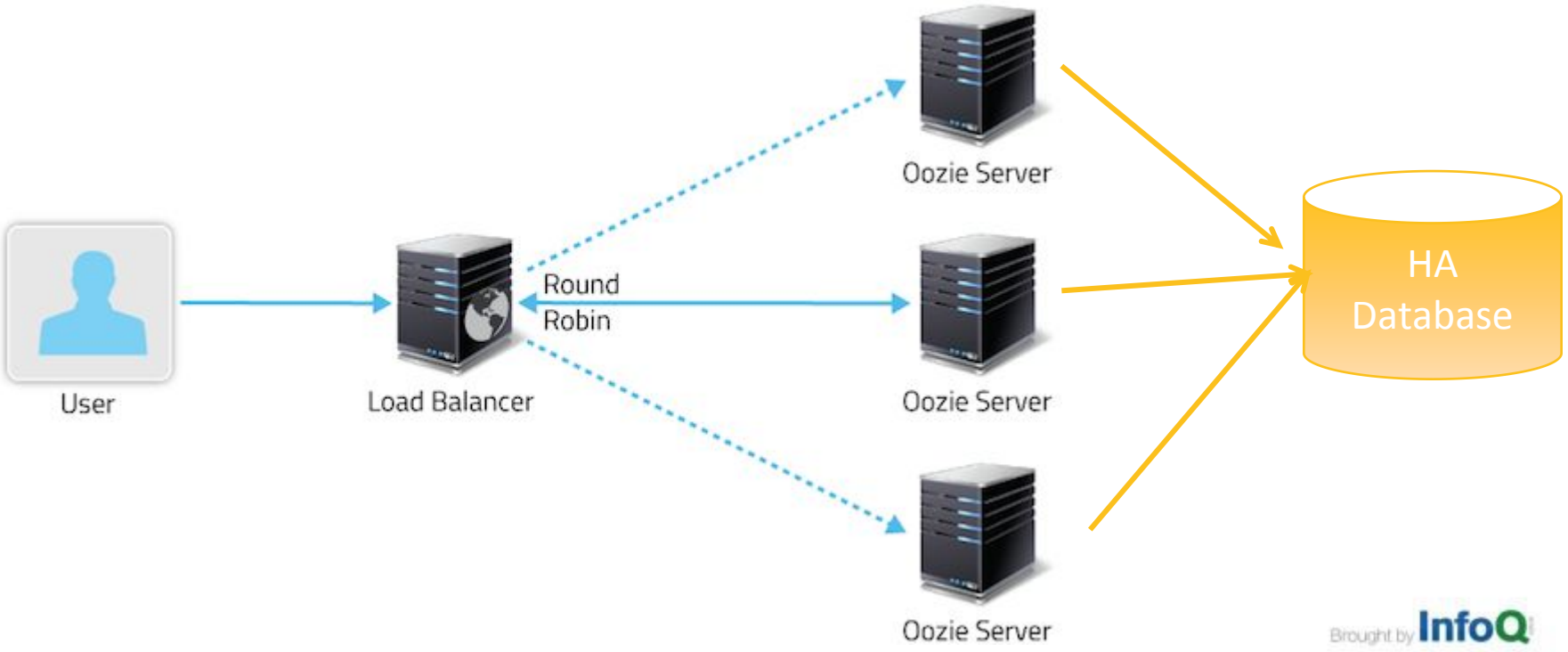
- 数据安全
  - 保障平台上的数据不被恶意用户所窃取、破坏和泄漏
  - 保障平台上的服务不被恶意操作和使用
  - 网络安全、主机安全、服务安全、数据安全
- 运维安全
  - 即工业界常提的安全生产
  - 提升平台的稳定性和可靠性，保障服务的正常运行
  - 权限独立管理、资源统一管理、服务热备双活

服务高可用

- HDFS NameNode HA
- YARN ResourceManager HA
  - 基于Zookeeper, 内置 “zkfc”, ActiveStandbyElector
  - 只负责ApplicationMaster的状态维护和容错恢复，而AppMaster内部的容错需要自行管理
  - RM切换会导致常驻服务重启，如Storm, HBase, thriftserver
  - Slider项目
- HBase Master HA
  - 基于Zookeeper
  - ActiveMasterManagement

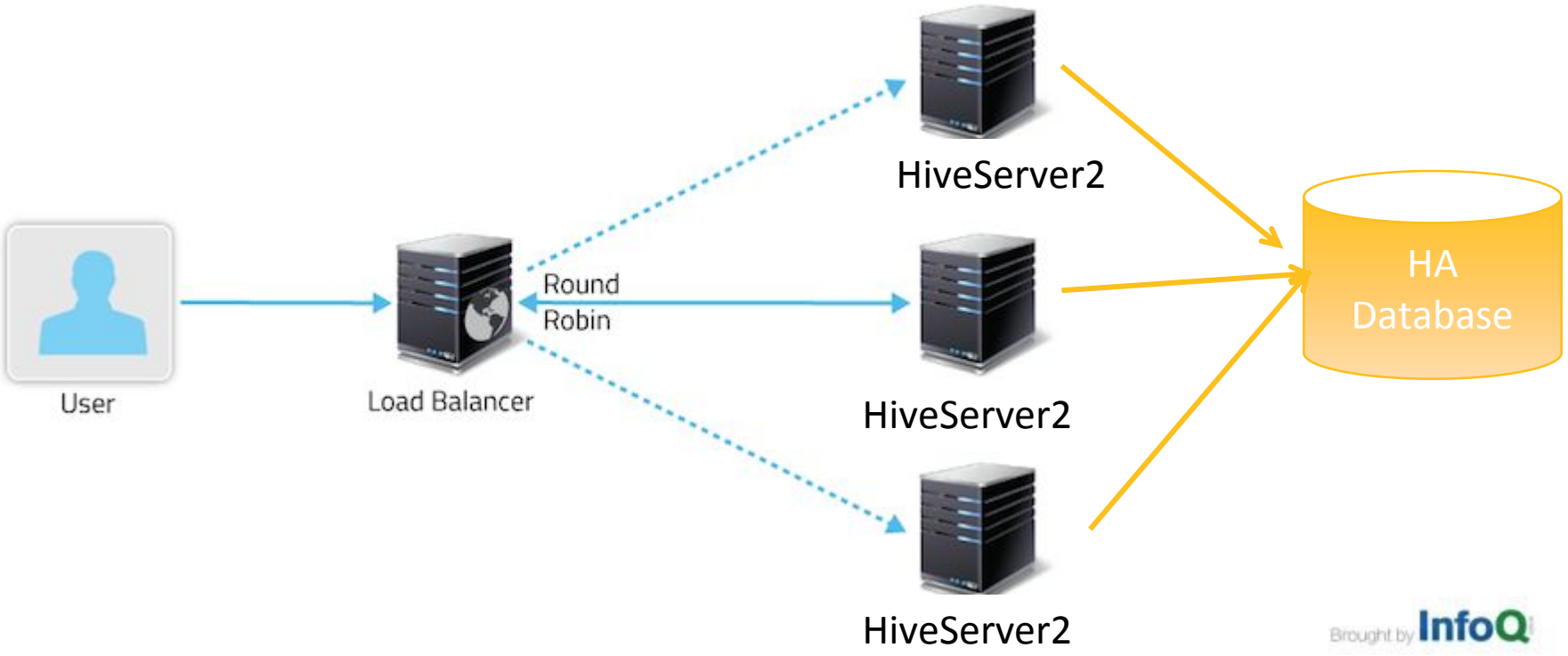
服务高可用

- 服务热备
  - Oozie



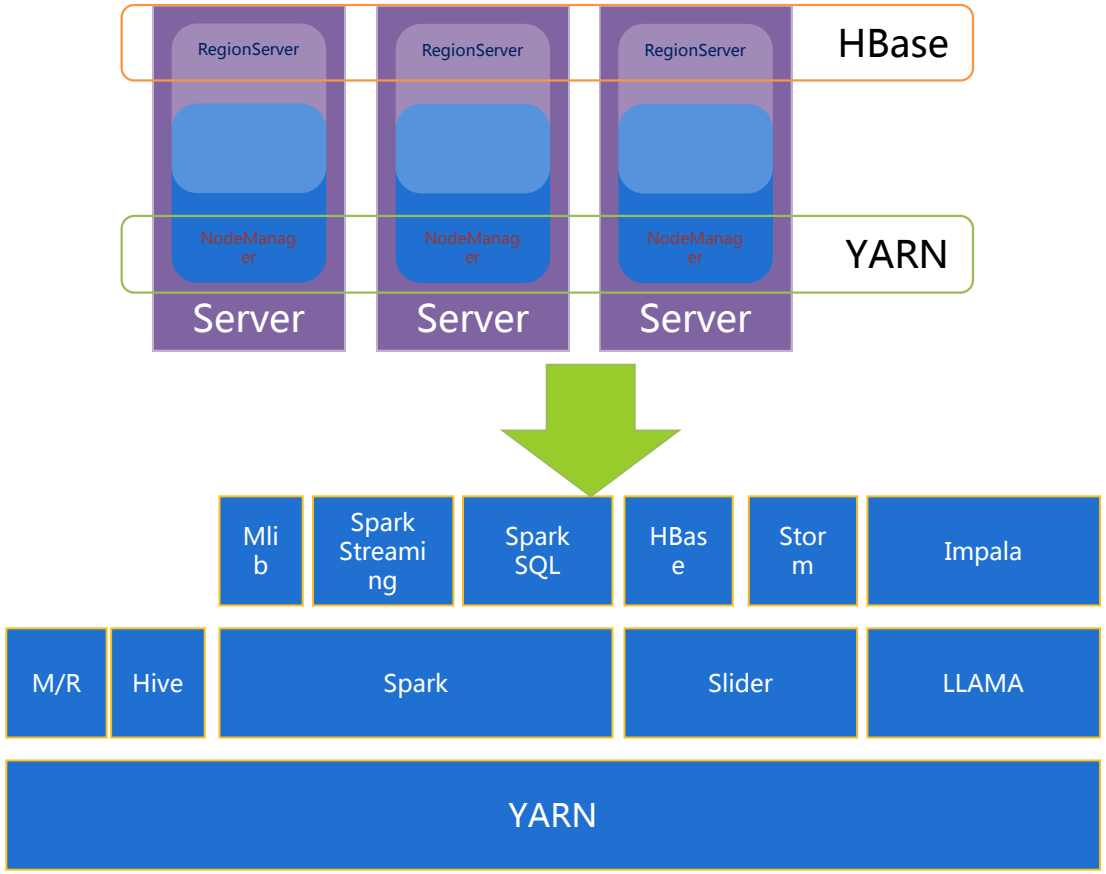
服务高可用

- HiveServer2 HA
  - HAProxy



平台可靠性

- Everything On Yarn，资源统一管理和分配，保障关键服务的资源使用



如何评估安全需求

- 用户
  - 业务线 – 开发者/数据分析师/平台管理员/安全审计人员…
  - 访问方式 – Shell登录/API访问/WEB访问
  - 使用服务 – HDFS/HBase/Hive..
  - 访问数据 – 日志文件/交易数据/个人身份信息…
- 环境
  - 网络环境 – 内网、外网，网络结构
  - 系统环境 – 操作系统、端口设定、漏洞补丁等
  - 云服务

THANKS