

A Reference Architecture for Securing Big Data Infrastructures

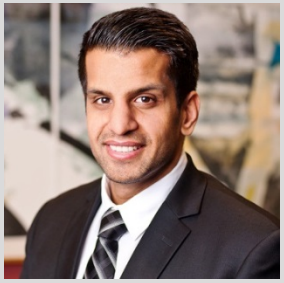
July 4 2015

Introduction

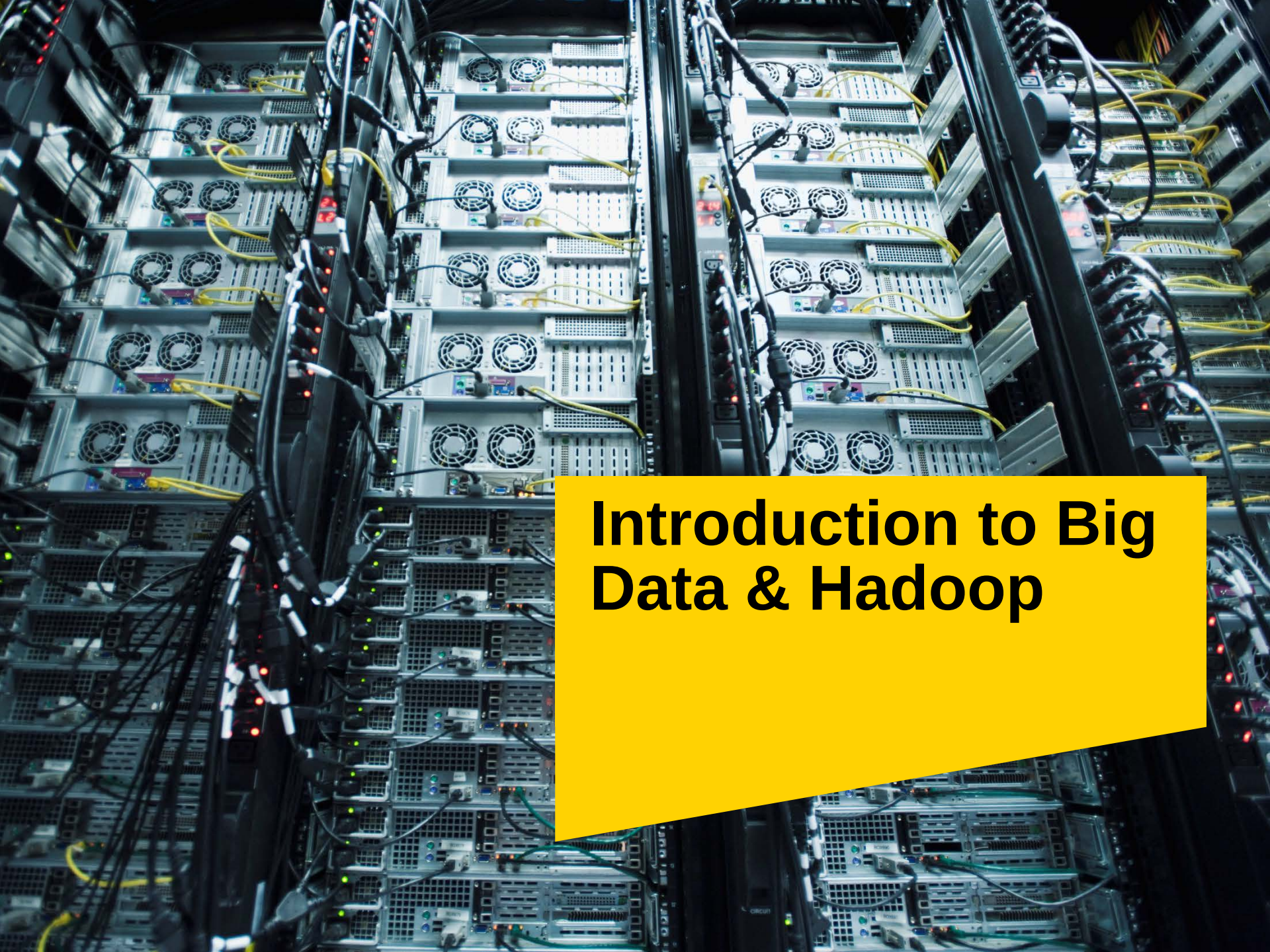
- Enterprise information repositories contain sensitive, and in some cases personally identifiable information that allows organizations to understand lifetime value of their customers, enable journey mapping to provide better and more targeted products and services, and improve their own internal operations by reducing cost of operations and driving profitability.
- Analysis of such large quantities of disparate data is possible today leveraging Hadoop and other Big data technologies. These large data sets create a very attractive target for hackers and are worth a lot of money in the cyber black market.
- Recent data breaches such as the ones at Anthem, Sony and others, drive the need to secure these infrastructures consistently and effectively.



Abhay Raman
Partner – Cyber Security and Resilience
abhay.raman@ca.ey.com
https://www.linkedin.com/in/abhayraman



Shezan Chagani
Manager - Cyber Security and Resilience
shezan.chagani@ca.ey.com
ca.linkedin.com/in/shezanchagani

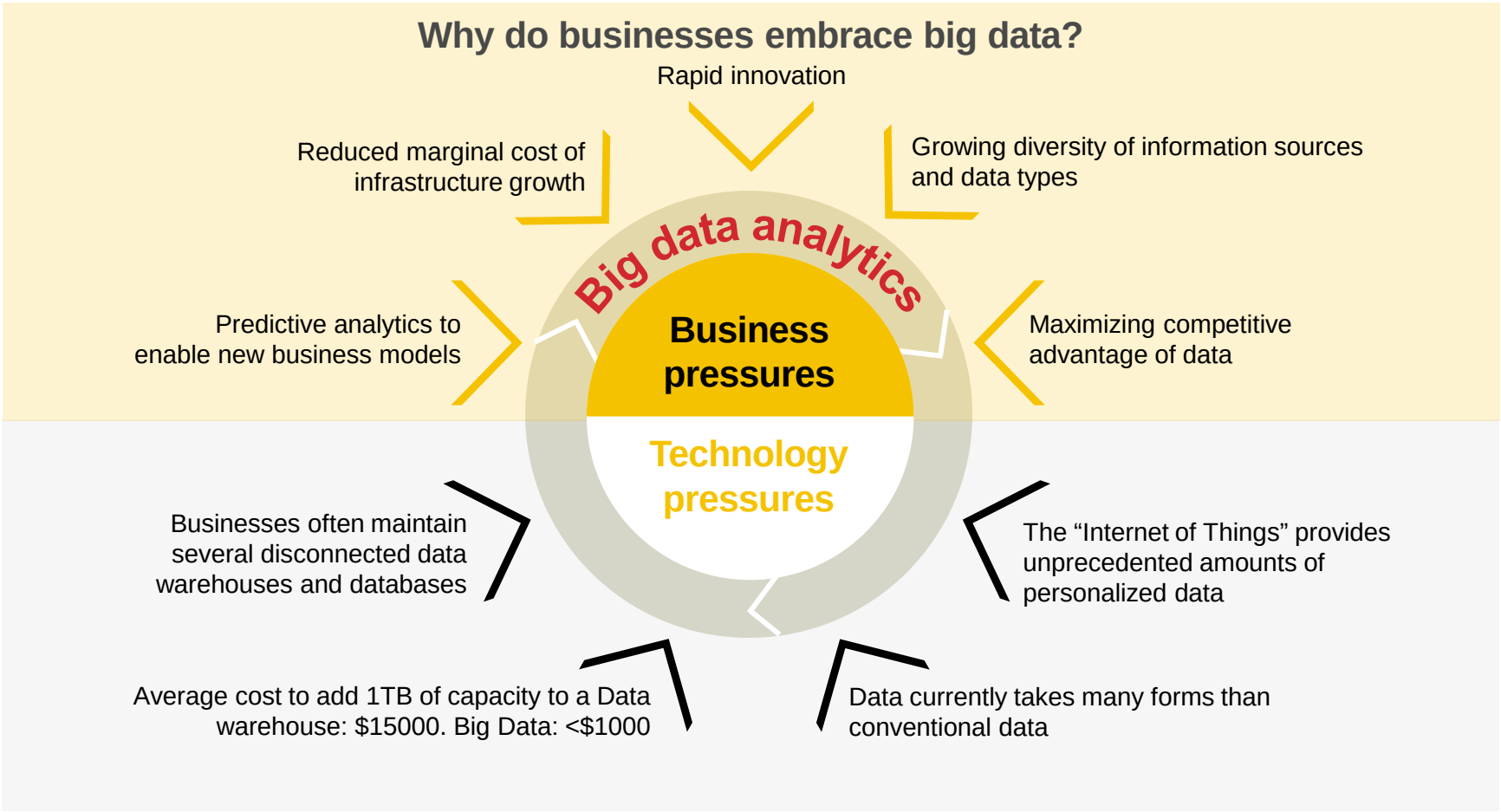


Introduction to Big Data & Hadoop

What is Big Data

- Big data is defined as the collection, storage, and analysis of disparate data sets that are so large that conventional relational databases are unable to deliver results in a timely or cost effective fashion.
- Hadoop is a flavour of Big Data analytics which uses a parallel storage architecture (HDFS) and analytics architecture (MapReduce) to process data using commodity hardware.
- The history of Hadoop dates back 12 years to Yahoo’s search indexing woes

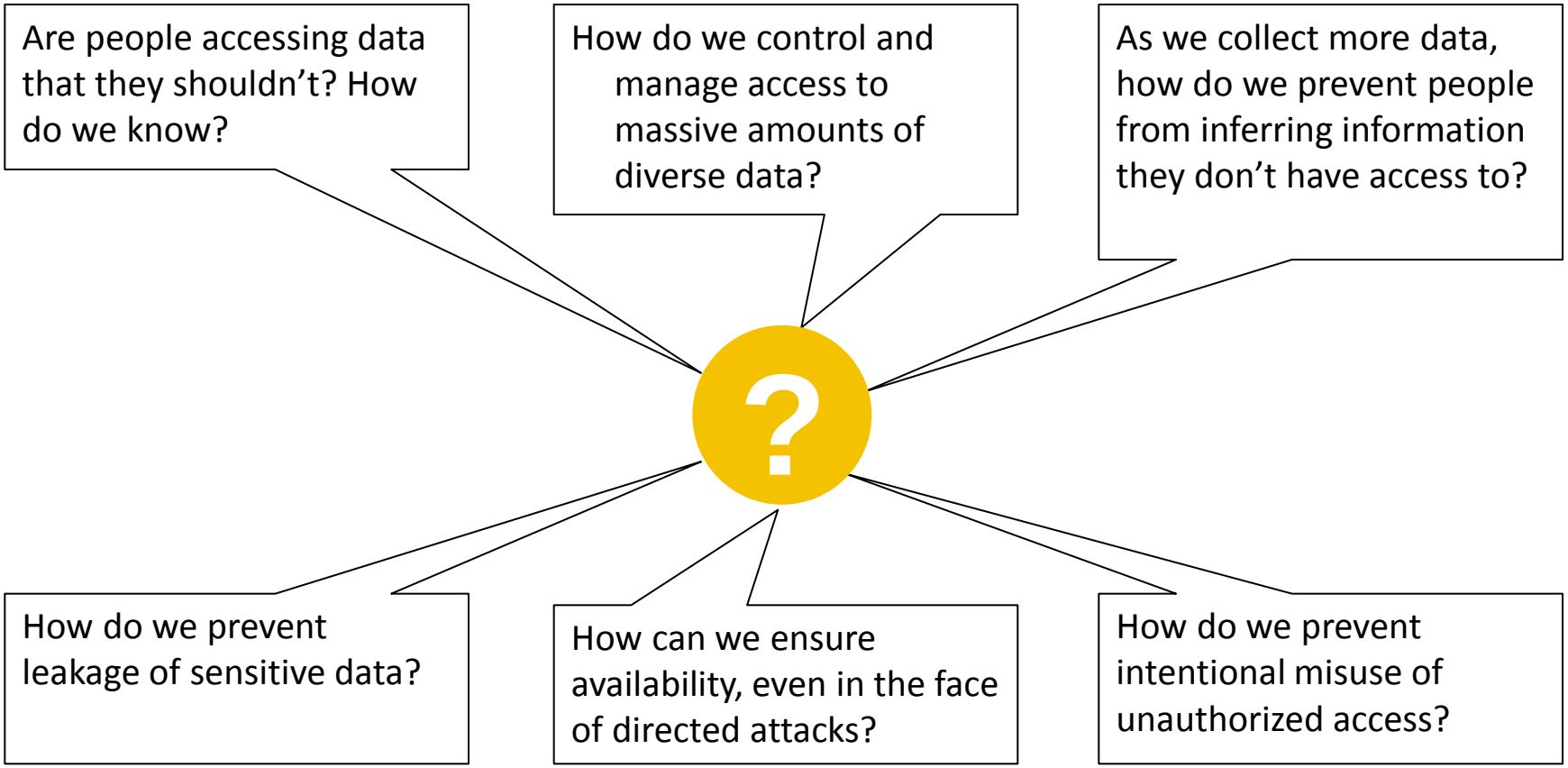
What is Driving Big Data Analytics



Big Data allows us to answer “What was the behavior that lead to the transaction?” and “How do we maximize that behavior to grow our business?”


Security Challenges

Security Challenges with Big Data environments



Common Hadoop Security Issues

Focus on perimeter security

- Perimeter security in typical RDMS implementation has always been heavily used.
- This same model is often used in Hadoop and internalized security is often missed.

Granularity and uniformity of authorization

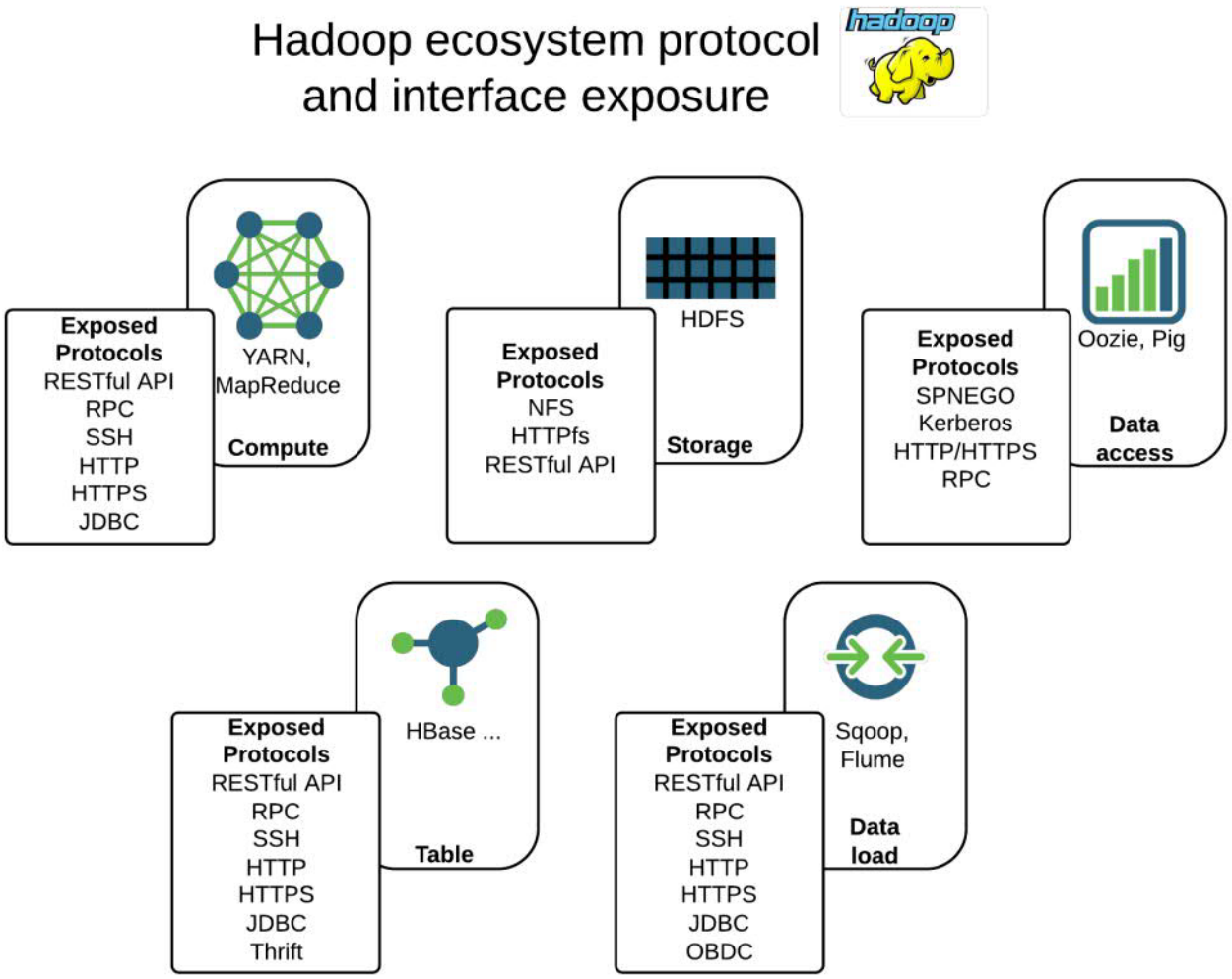
- Components within the Hadoop ecosystem offer vastly different capabilities for authorization of transactions
- Introduces the possibility of escalated privileges/access by oversight or gap.

Organizations typically miss Security “gotcha’s” during configuration

- Hadoop requires additional configurations to secure the ecosystem; which are often missed

Hadoop Attack Surface

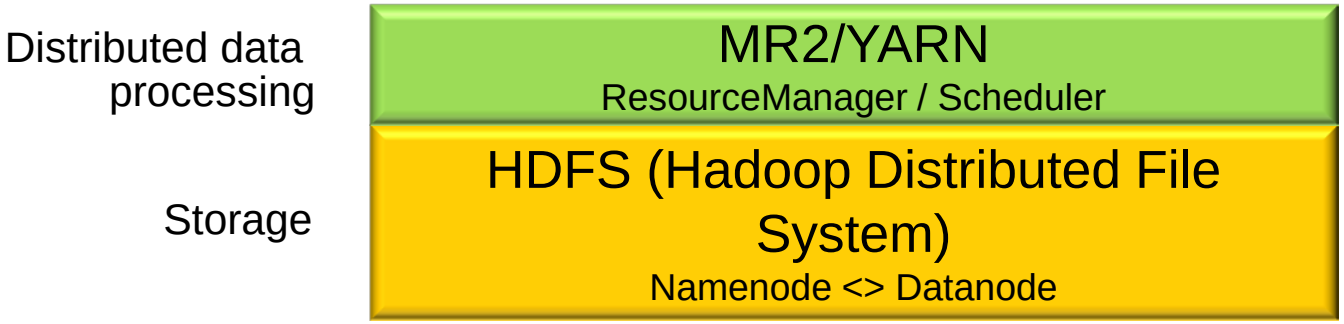
- Hadoop contains a wide variety of components and protocols
  - Some interfaces are not authenticated
  - Interfaces across components not standardized
- Many interchangeable components which exacerbate the different level of security approaches
- Project Rhino to help standardize




Building a Secure
Hadoop Ecosystem

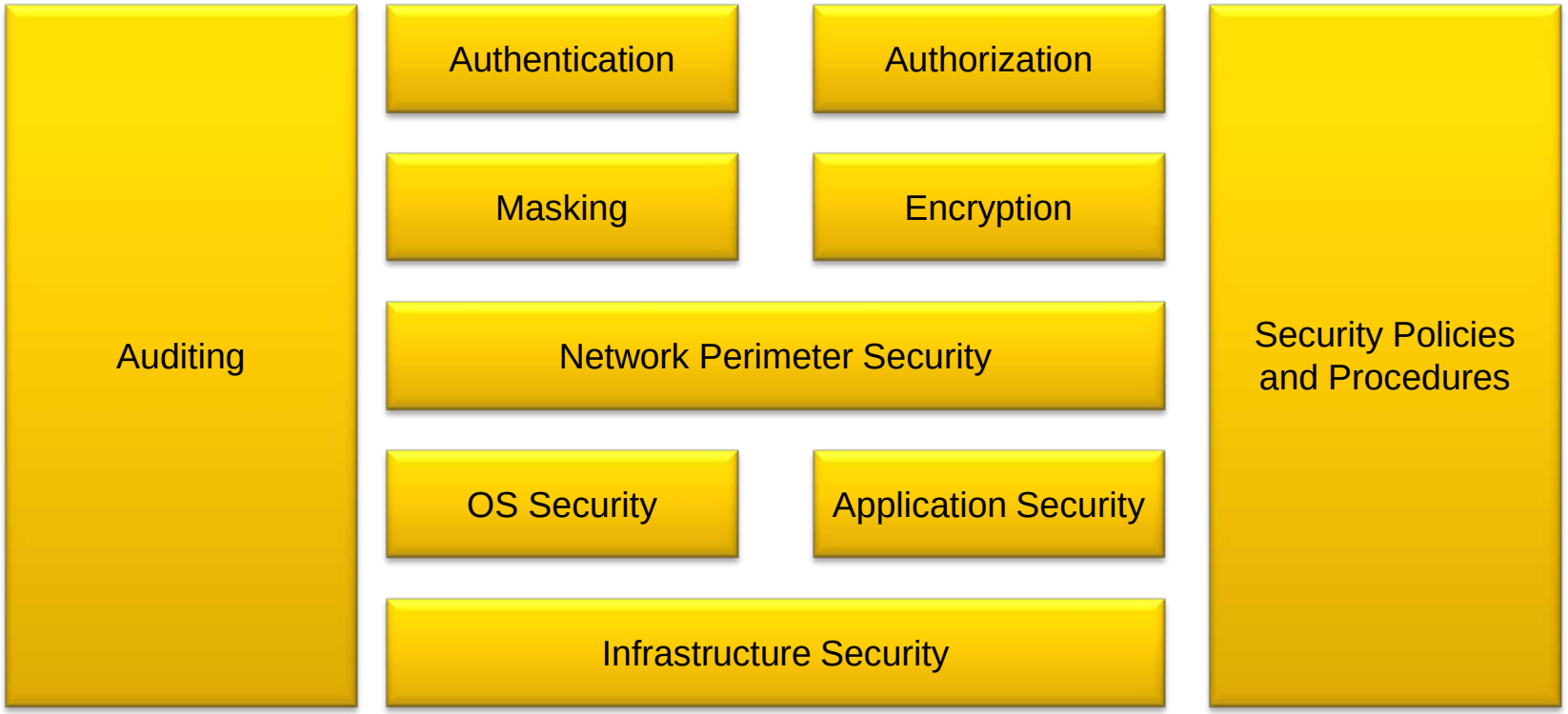
Hadoop Architecture

- Let's review key components in Hadoop to lay the ground of the architecture:
  - HDFS (Hadoop Distributed File System)
    - Namenode – track metadata of file stores (filenames, locations, permissions)
    - Datenode – actual storage of blocks of data in HDFS (retrieval)
  - MR2/YARN
    - MapReduce (Mappers and Reducers) – programming model
    - YARN (Yet Another Resource Negotiator) – Resource Management
- MR1 components not covered in this presentation (e.g. Jobtracker/Tasktracker)



Framework to a Secure Hadoop Infrastructure

The below reference architecture summarizes the key security pillars that need to be considered for any Hadoop Big Data Ecosystem



Authentication

- Hadoop by default enables no authentication
- Challenges with distributed computing
  - Multi-step process to access a file (many points of security)
  - MapReduce jobs are not run in real time
  - Multi-tenancy concerns

Recommendations:

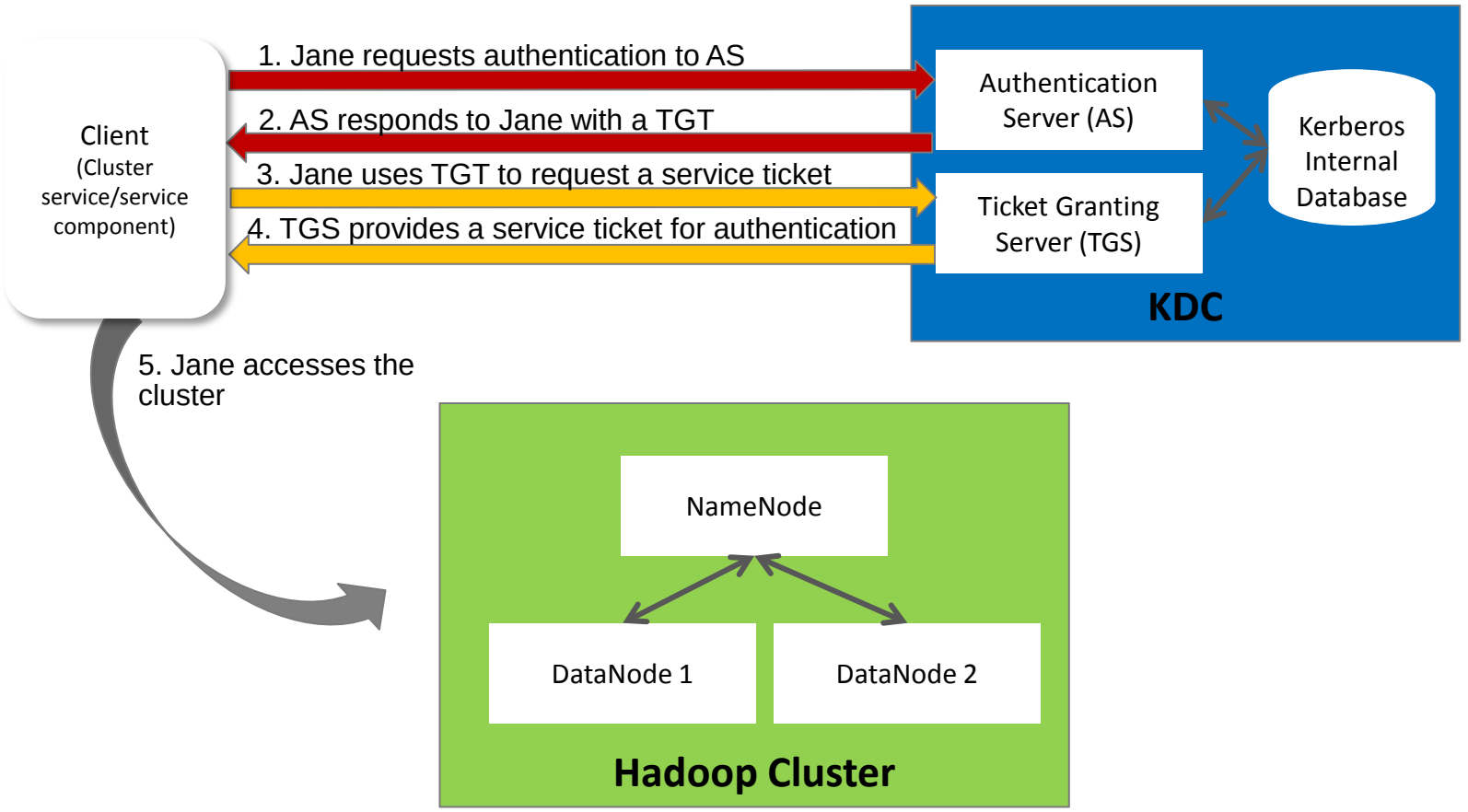
- Kerberize the cluster
- Integrate directory services

Why Kerberos?

- Hadoop clusters were initially perceived as an open trusted network with no strong security measures
- When shipped Hadoop provides two types of authentication:
  - simple auth– uses UID of OS to determine username and passed user into Hadoop via client side libraries
  - Kerberos – third party authentication
- Kerberos should be utilized as the authentication mechanism in any multi tenant/production cluster

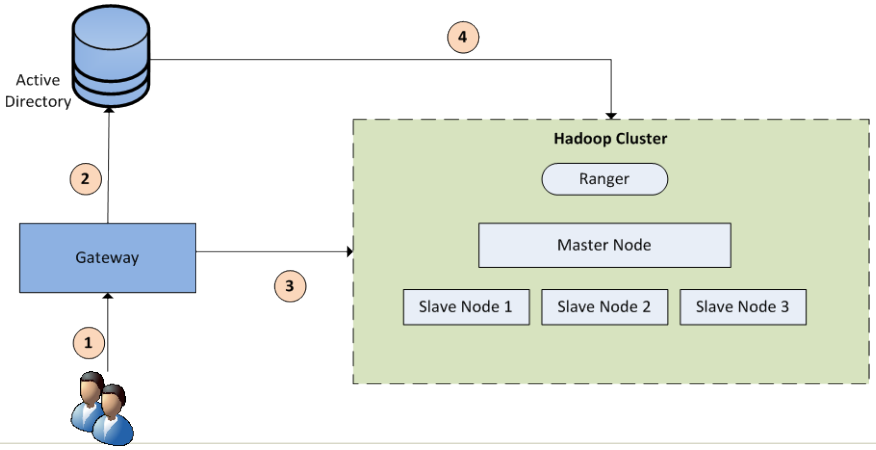
Overview of a Kerberos Flow

- Each user and component is a principle in the KDC (UPN & SPN)
- Below diagram provides an overview of a simple Kerberos flow in a Hadoop cluster:



Integrating an Enterprise Directory

- Leverage your enterprise users/groups by integrating to Active Directory/LDAP
- Seven options, but two popular options exist to integrate:
  - **LdapGroupsMapping**
  - **JniBasedUnixGroupsMappingWithFallback** (default)
- Recommend JNI (Java Native Interface) with Shell backup for a few reasons:
  - Unix SSSD OS Integration – Proven System Security Services Daemon's scalability, caching, and offline access
  - Lose internal Hadoop groups with LdapGroupsMapping option
- Our experience has found 600 seconds as the best caching timeframe



Authorization

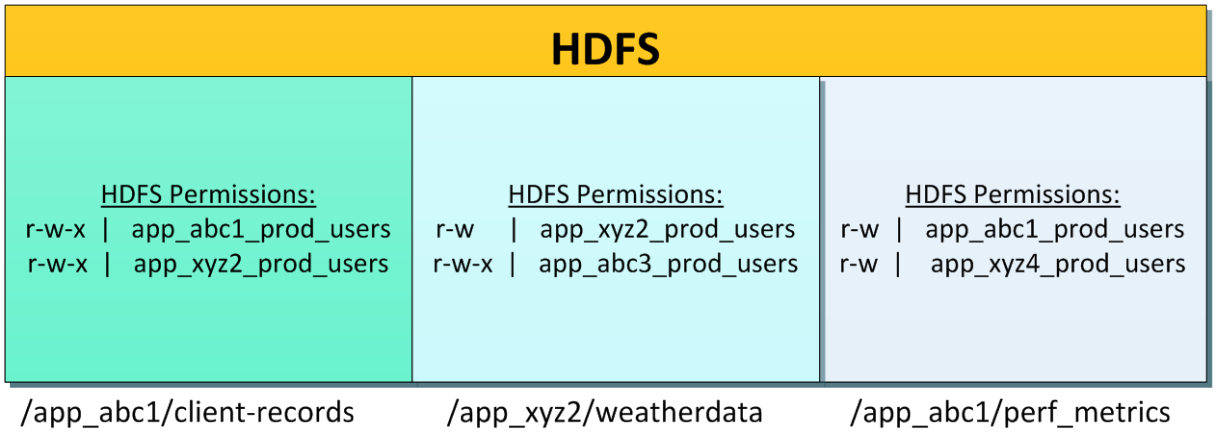
- Need to model the actions that can be performed once a user has been authenticated
- Dimensions from clients on protecting access between departments, multi level authorization, and queue management
- Greatest challenge: Each component is unique in the service it provides so therefore the authorization it uses is unique to each component

Recommendations

- Enable HDFS permissions
- Build out service level authorizations
- Secure jobs between users
- Build out queue modeling across the cluster
- Authorization centralization

HDFS Authorizations

- Multi tenancy requires the maintenance of many groups
  - With Hadoop 2.4 extended ACL's are used HDFS - This allows applying multiple group management
- Clients struggle with managing group sizes - ACL's have a maximum of 32 entries (with 4 taken by default)
  - Errors will be thrown by the NodeManager →"setfacl: Invalid ACL: ACL has 33 entries, which exceeds maximum of 32"



Service Level Authorizations

- Authorization is a multi step process in Hadoop – can't just protect the data, need to protect components that provide access to the data (services)
- At a minimum, service level authorization should be applied at the YARN level
- YARN Authorization:
  - YARN provides limits on job queues to ensure access is controlled to resources, and set priorities as well on job execution
  - Important:
    - yarn admin configurations should only be for admin groups
    - yarn acl's for job executions for end user groups

```
<property>
<name>yarn.acl.enable</name>
<value>true</value>
</property>
<property>
<name>yarn.admin.acl</name>
<value>yarn hadoop-admins</value>
</property>
```

Capacity Schedulers

- YARN provides access to job queues and managing resources
- Two schedulers for YARN: CapacityScheduler, and FairScheduler
- Queues split up totalling 100
- Allows queuing for normal jobs, but also controls administrative access, such as being able to kill jobs of other users

```
<property>
<name>yarn.scheduler.capacity.root.acl_submit_applications</name>
<value> hadoop_users</value>
</property>
<property>
<name>yarn.scheduler.capacity.root.acl_administer_applications</name>
<value> hadoop-admins</value>
</property>
<property>
<name>yarn.scheduler.capacity.root.queues</name>
<value>prod,default</value>
</property>
```

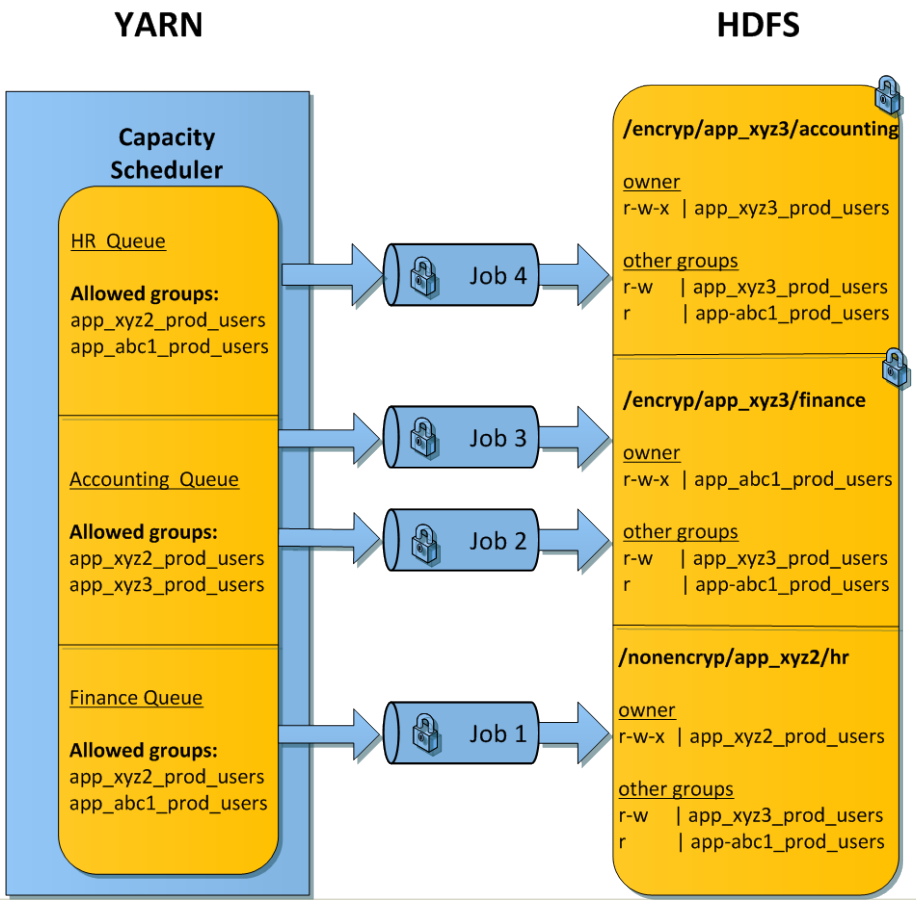
Securing Job Executions

- Multi-tenancy is a requirement of most Hadoop implementations, thus job access security between users is very important
- LinuxContainerExecutor should be used to secure job execution within any Hadoop environment
- Uses a setuid when launching YARN containers
  - allows NodeManager to run containers using UID of the person that submitted the job
- All containers would run as “yarn” allowing local files to be read between users
- Can also set restrictions: min uid, allowed system users, ban users

yarn.nodemanager.linux-container-executor.group=*yarn*
min.user.id=*1000*
allowed.system.users=*nobody,impala,hive,llama*
banned.users=*root,hdfs,yarn,mapred,bin*

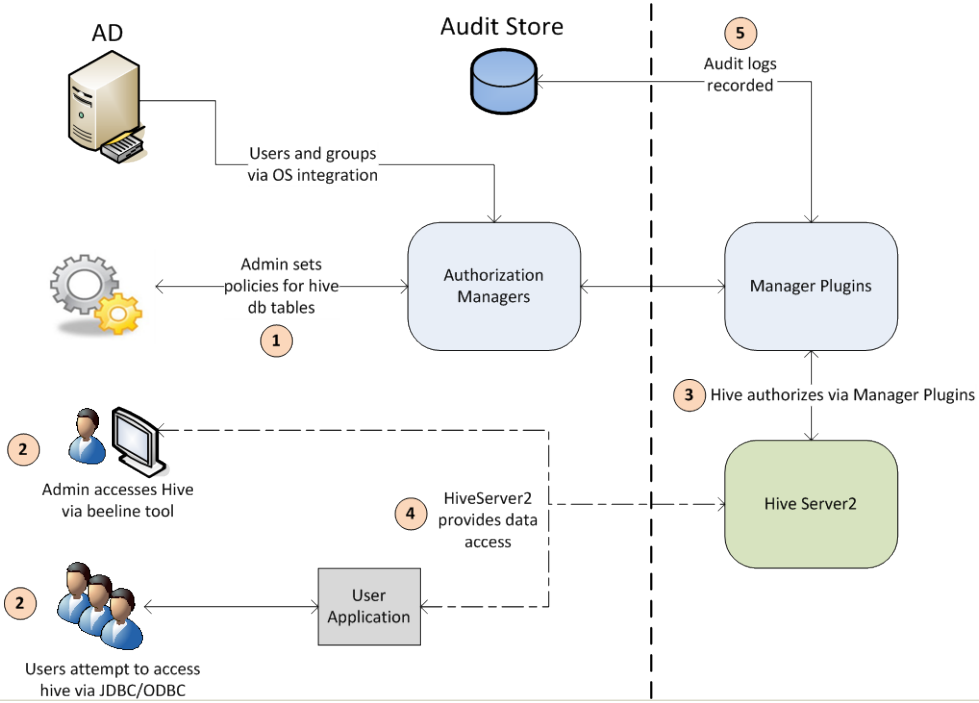
Container Execution - Spinning off Jobs to HDFS

- The below diagrams depicts how jobs would be spun off by YARN which would be separated from other jobs operating within the cluster in parallel:



Centralization of Authorization

- Decentralization of authorization creates risk (stale ACL's, mistakes in addition)
- Authorization manager examples: Apache Ranger or Cloudera Manager
- These managers include their own directories that are integrated with various components



Audit

- Auditing measures shouldn't be viewed as a way to satisfy your security compliance measures or to meet regulatory conditions – they can help stop security breaches before they happen
- Auditing completes a security model by providing records of what happened which can be used for:
  - Active Auditing
    - Auditing used in conjunction with an alerting mechanism
  - Passive Auditing
    - Refers to auditing that does not generate an alert
- Challenge: Disparate components create disparate log files to manage

Recommendations

- Utilize HDFS, and MapReduce audit logs
- Utilize log aggregation across nodes

HDFS & MapReduce Audit Logging

- HDFS and MapReduce are critical in providing programming logic + data storage
- Two main audit loggers in HDFS:
  - **hdfs-audit.log** –user activity such as new file creation, permissions changes, directory listing requests, etc
  - **SecurityAuth-hdfs.audi**t – audits service level authorization into the HDFS
- MapReduce also utilizes two main audit loggers
  - **Mapred-audit.log** – audits user activity such as job submissions
  - **SecurityAuthMapRed.audi**t – where service level authorization is turned on similar to HDFS's logger

MapRed

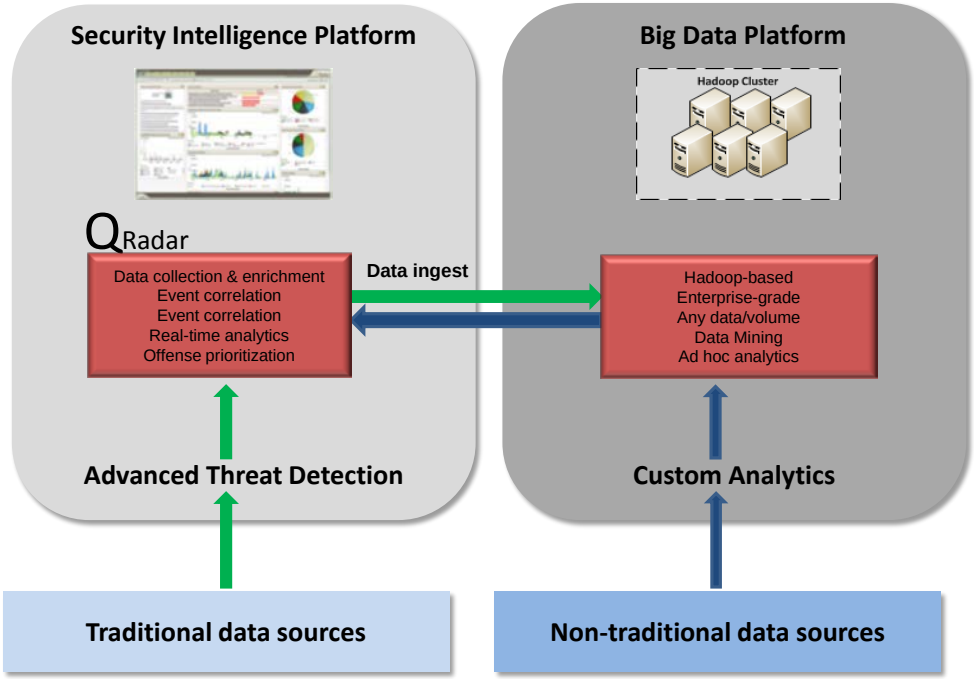
- 2014-03-12 18:11:46,363 INFO mapred.AuditLogger: USER=bob IP=10.1.1.1 OPERATION=SUBMIT_JOB
- TARGET=job_201403112320_0001 RESULT=SUCCESS

SecurityAuthMapRed

- 2014-03-12 18:46:25,200 INFO SecurityLogger.org.apache.hadoop.ipc.ServerAuth successful for bob@EXAMPLE.COM (auth:SIMPLE)
- 2014-03-12 18:46:25,239 INFO SecurityLogger.org.apache.hadoop.security.authorize.ServiceAuthorizationManager: Authorization successful for bob@EXAMPLE.COM (auth:KERBEROS) for protocol=interface org.apache.hadoop.mapred.JobSubmissionProtocol

Log Aggregation

- Many other components in cluster will also have logging turned on
  - Correlating these events can be a huge undertaking
- Logs be aggregated across one common storage area for analysis
- Examples with clients: Apache Ranger or IBM Qradar integration (or both)
- Some clients reinject data for analytics within Hadoop itself for passive auditing



Data Encryption

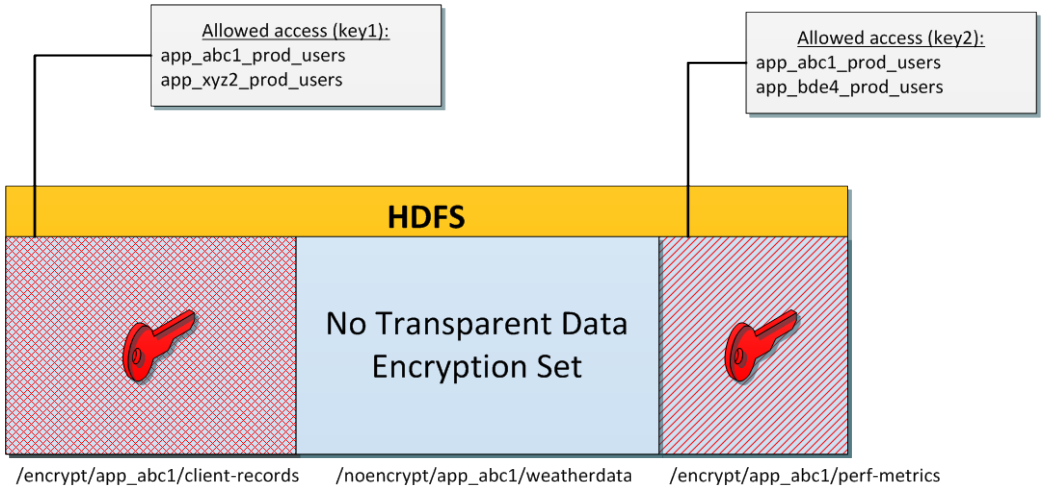
- There are two categories of data within Hadoop that are the focus of data protection.
  - Sensitive data that has been pushed into Hadoop (business data or customer data). Data was brought into Hadoop for analysis
  - Insights which is data that has already been analyzed. Such information if exposed can lead to great losses as correlation has already been established
- This data is secured by focusing on data at rest and data in motion

Recommendations

- Data at rest: Utilize encryption zones
- Data in motion: encrypt standard Hadoop connections (RPC, SASL, etc)

Data at Rest

- Three options when it comes to securing data at rest:
  - Volume level encryption – least favourable
  - Block level encryption – Future state and still in incubation
  - HDFS Transparent Data Encryption (TDE) – Recommended approach
- HDFS TDE - data is secured within HDFS across the cluster – this requires:
  - Setup a KMS (Key Management Server) to house secure access to keys
  - Set the Key Provider API's to allow components access to KMS
  - Finally set encryption zones using the keys generated

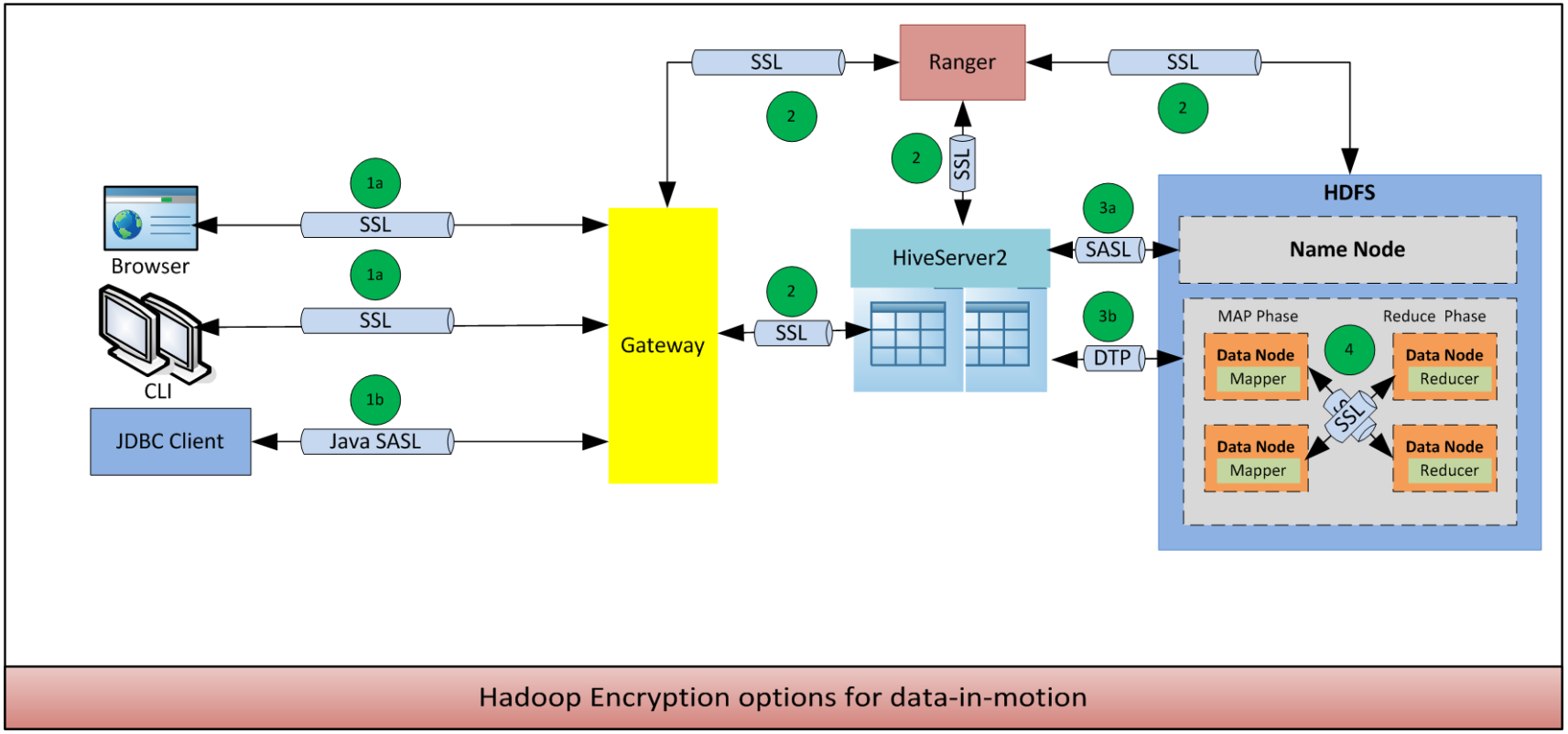


Data in Motion

- Data in motion is secured between all components at the following levels:
  - **At the client**
    - **RPC encryption** –Client communication to NameNode occur over RPC which uses SASL (Simple Authentication and Security Layer)
    - **Data Transfer Protocol** –Client communication to DataNode occurs over DTP which supports encryption
  - **User mechanisms** (e.g. browsers/command line tools)
    - **HTTPS encryption** – Interactions from users occur through browsers or Command Line Interfaces
    - **JDBC (Java Database Connectivity)** – JDBC connections such as HiveServer2 can utilize encryption by using Java ASL QOP (Quality of Protection)
  - **Secure the shuffle**
    - Communication between the “Mappers” and “Reducers” is called the shuffle which occurs over HTTP

Data in Motion Encrypted Environment

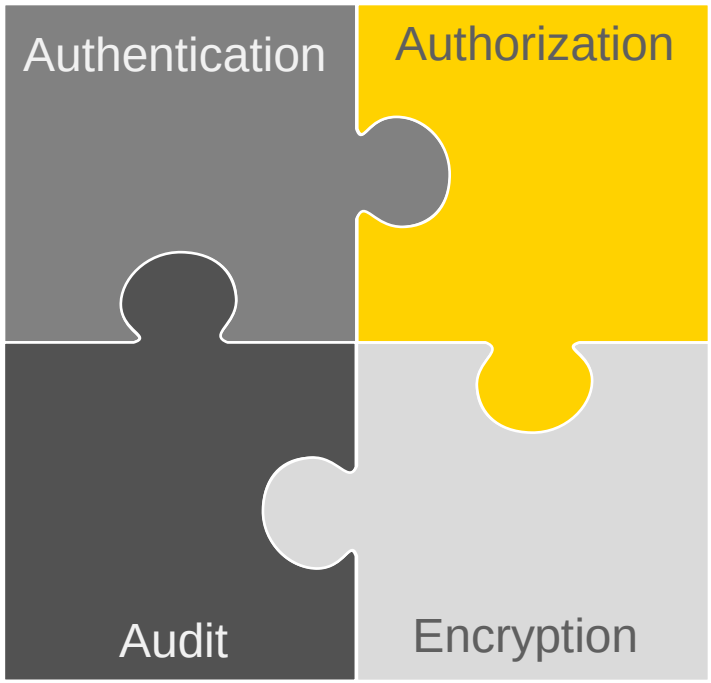
- Below is an illustration of a these data in motion encryption standards set throughout a cluster:



In Closing…

- Extent of security configurations should always be specific to your requirements
- Configurations in this presentation included:

Authentication:
- Kerberos
- Active Directory



Authorization:
- File system level
- Service level
- Secure job submissions
- Centralizing authorization

Auditing:
- Logging with HDFS and MapReduce
- Centralizing auditing via SIEM's

Encryption:
- Data at rest: HDFS TDE and future state encryption considerations
- Data in motion: securing protocols across the cluster

Thank You

References

1. http://www.computerweekly.com/feature/How-to-tackle-big-data-from-a-security-point-of-view
2. http://www.forbes.com/sites/kurtmarko/2014/11/09/big-data-cyber-security/
3. https://downloads.cloudsecurityalliance.org/initiatives/bdwg/Big_Data_Analytics_for_Security_Intelligence.pdf
4. http://www.isaca.org/groups/professional-english/big-data/groupdocuments/big_data_top_ten_v1.pdf
5. https://cloudsecurityalliance.org/media/news/csa-releases-the-expanded-top-ten-big-data-security-privacy-challenges/
6. http://www.csoonline.com/article/2130877/data-protection/data-protection-the-15-worst-data-security-breaches-of-the-21st-century.html
7. http://www.eweek.com/security/huge-federal-data-breach-a-prelude-to-even-more-dangerous-exploits.html
8. http://www.hpenterprisesecurity.com/collateral/whitepaper/BigSecurityforBigData0213.pdf
9. http://searchsecurity.techtarget.com/answer/What-is-big-data-Understanding-big-data-security-issues
10. http://www.renci.org/wp-content/uploads/2014/02/0213WhitePaper-SMW.pdf
11. http://www.computerworld.com/article/2498601/business-intelligence/it-must-prepare-for-hadoop-security-issues.html
12. https://cloudsecurityalliance.org/download/big-data-analytics-for-security-intelligence/
13. http://searchsecurity.techtarget.com/tip/Security-big-data-Preparing-for-a-big-data-collection-implementation
14. http://www.ijcsit.com/docs/Volume%205/vol5issue02/ijcsit20140502263.pdf
15. Hadoop Security, Ben Spicey & Joey Echeverria
16. Professional Hadoop Solutions, Boris Lubilnsky & Kevin T. Smith & Alexey Yakubovich