Multi Tenant Security Architecture for Big Data Systems

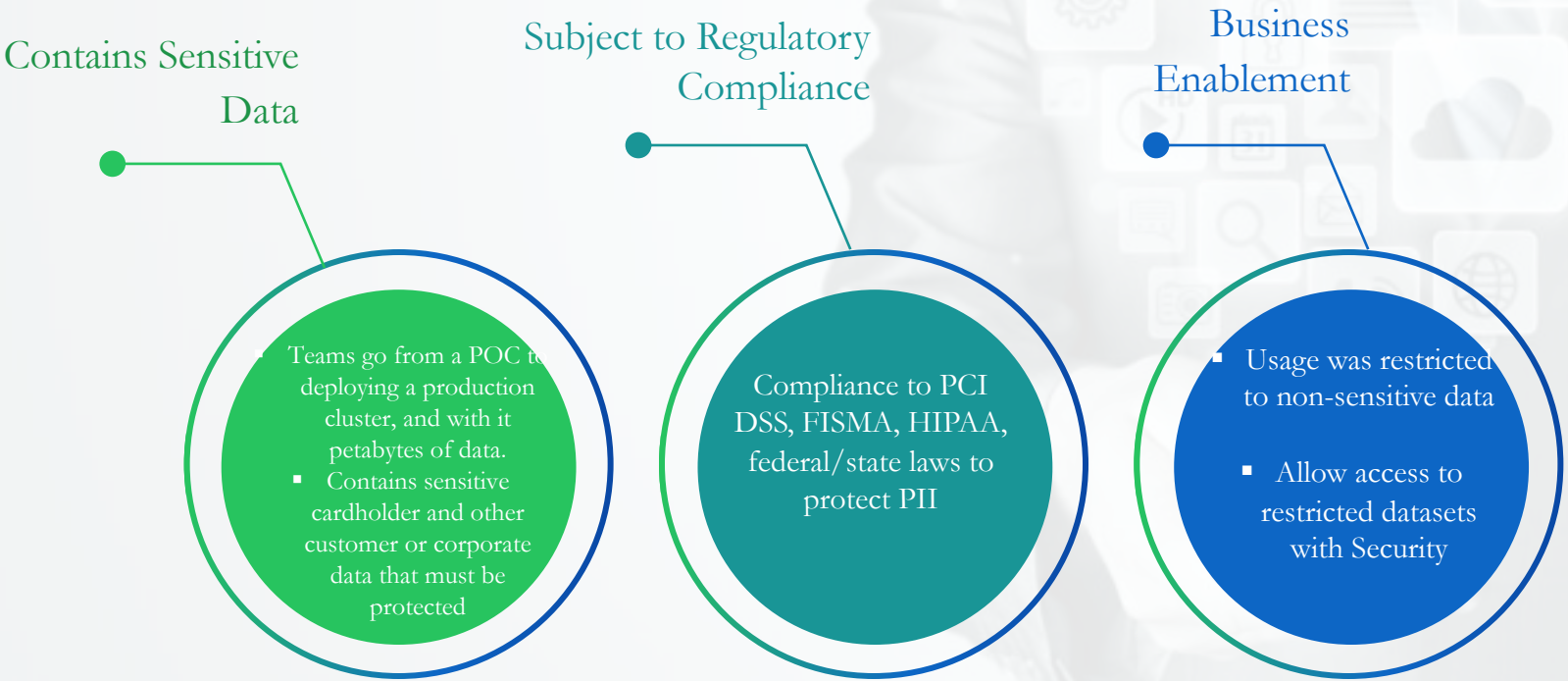
Suresh Yadagotti Jayaram
Sr. IT Technical Architect

What is Big Data

"Big Data refers to datasets whose size and/or structure is beyond the ability of traditional software tools or database systems to store, process, and analyze within reasonable timeframes"

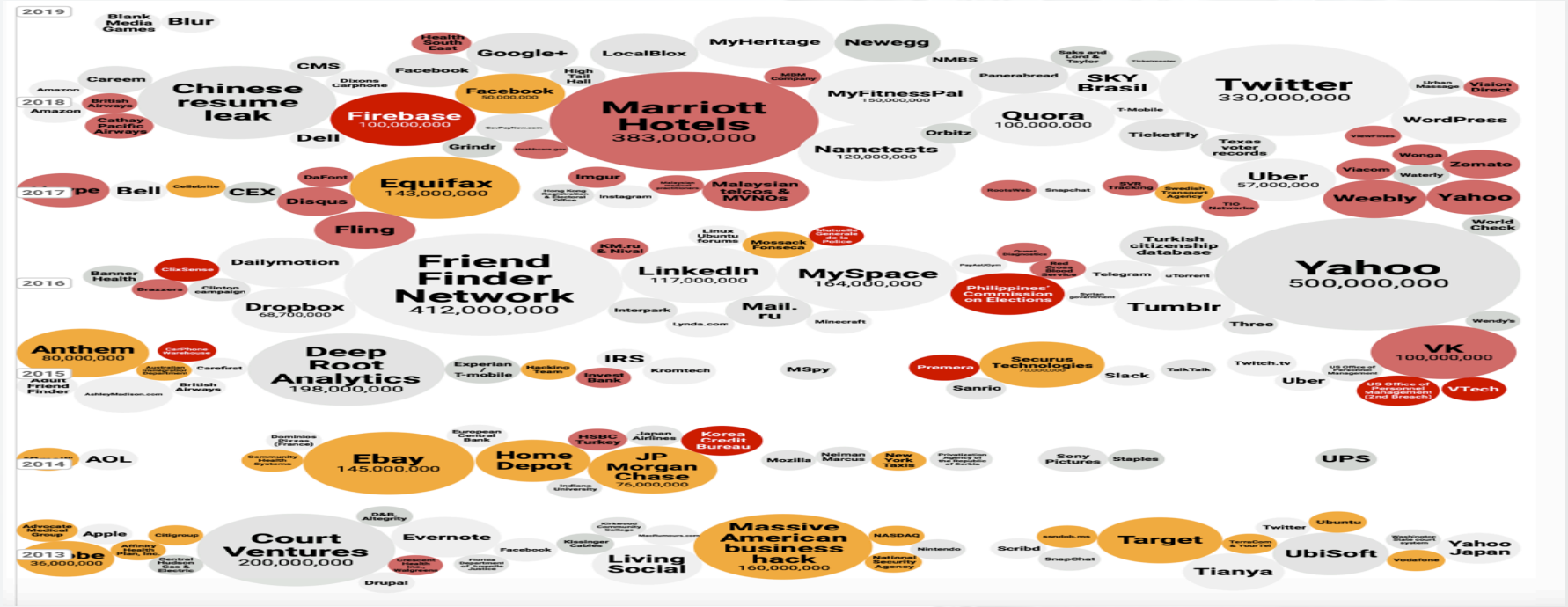
HADOOP is a computing environment built on top of a distributed clustered file system (HDFS) that was designed specifically for large scale data operations (e.g. MapReduce)

Reasons for securing data in Big Data systems



Data Breaches & Hacks

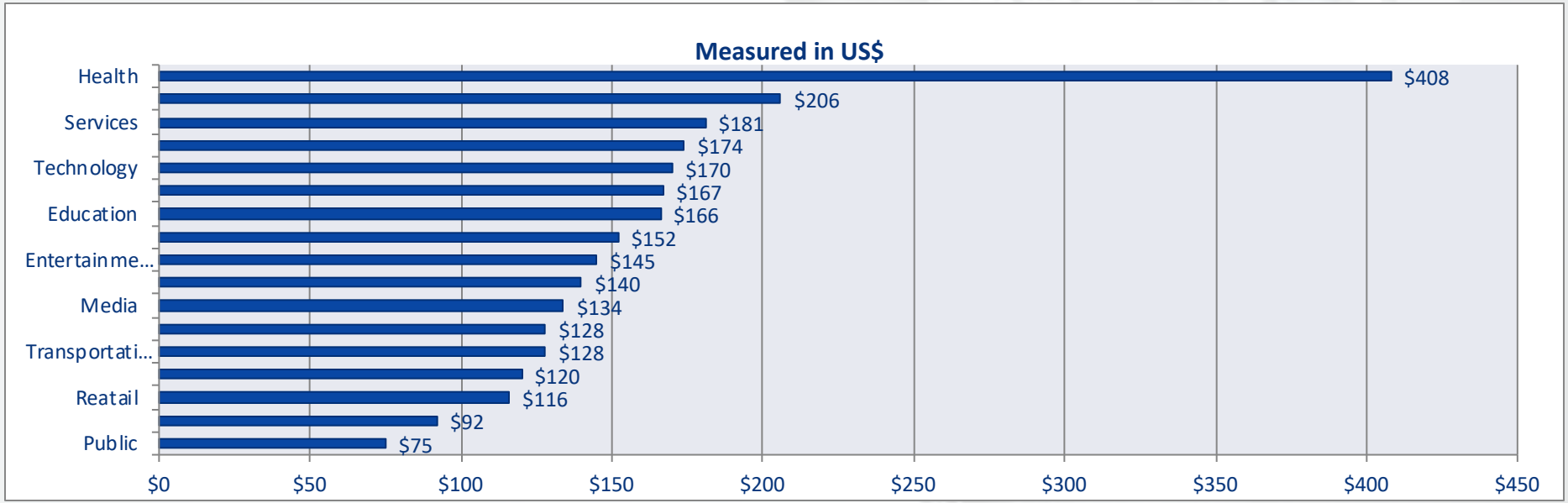
Different kinds of PII, financial data, and IP breached. Healthcare, Retail, Federal Govt., Financial Institutions, Tech companies etc.



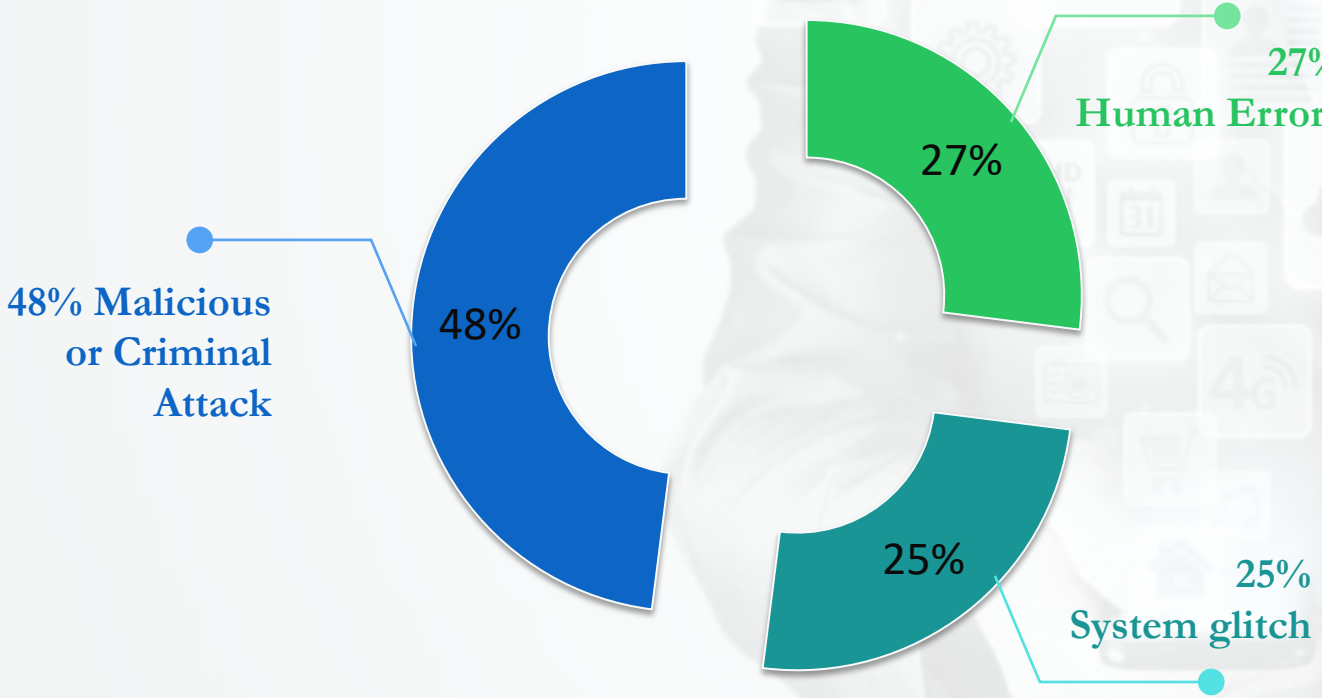
Per capita cost – Industry Sector

Certain industries have higher data breach costs. compares 2018 year's per capita costs for the consolidated sample by industry classification.

As can be seen, heavily regulated industries such as healthcare and financial organizations have a per capita data breach cost substantially higher than the overall mean.



Root Causes


27%
Human Errors
27%
48% Malicious
or Criminal
Attack
48%
25%
25%
System glitch

Goals of an Attacker

01

The primary goal is to obtain sensitive data that sits in Organization Databases

02

This could include different kinds of regulated data (e.g. Payment data, Heath data) or other personally identifiable data (PII)

03

Other attacks could include attacks attempting to destroy or modify data or prevent availability of this platform.

Threats – Counter measures

Attacker attempts to gain privileges to access data

Unauthorized access

- Authentication
- Authorization
- Auditing

Host Level Data at Rest Attacks

- Application Level
- HDFS level
- File System/Volume level

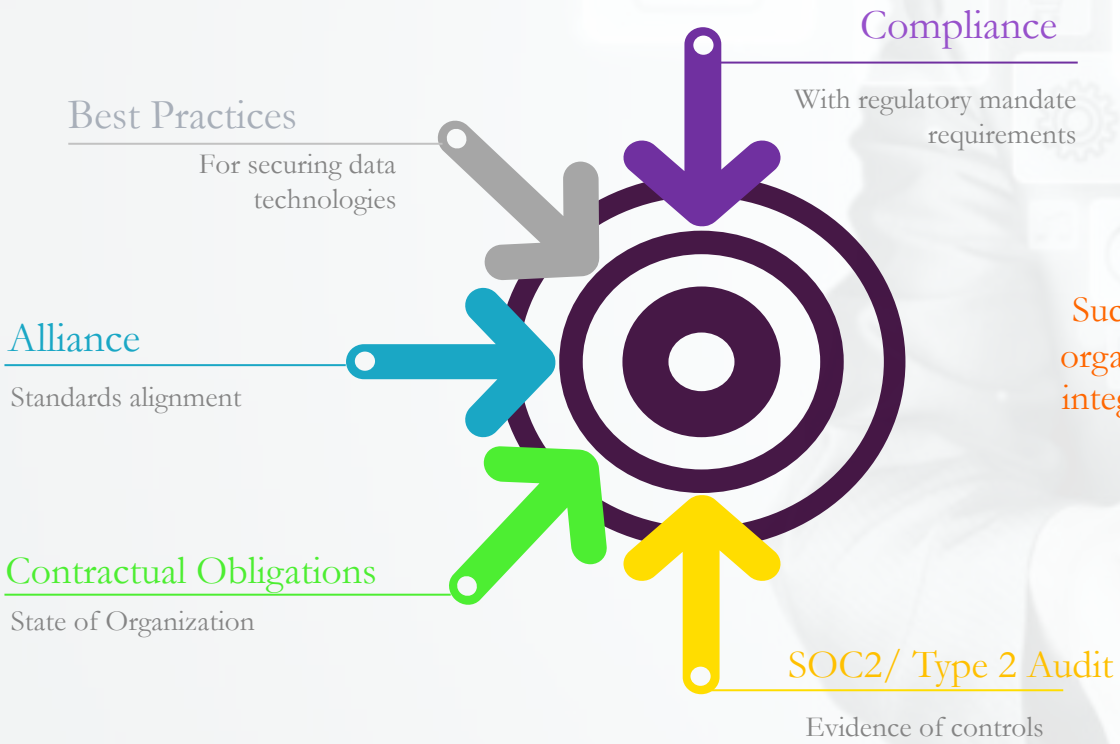
Network Based Attacks

- Transport Layer Security
- SASL Encryption

Infrastructure Security

- Automation
- SELinux

Security Objectives

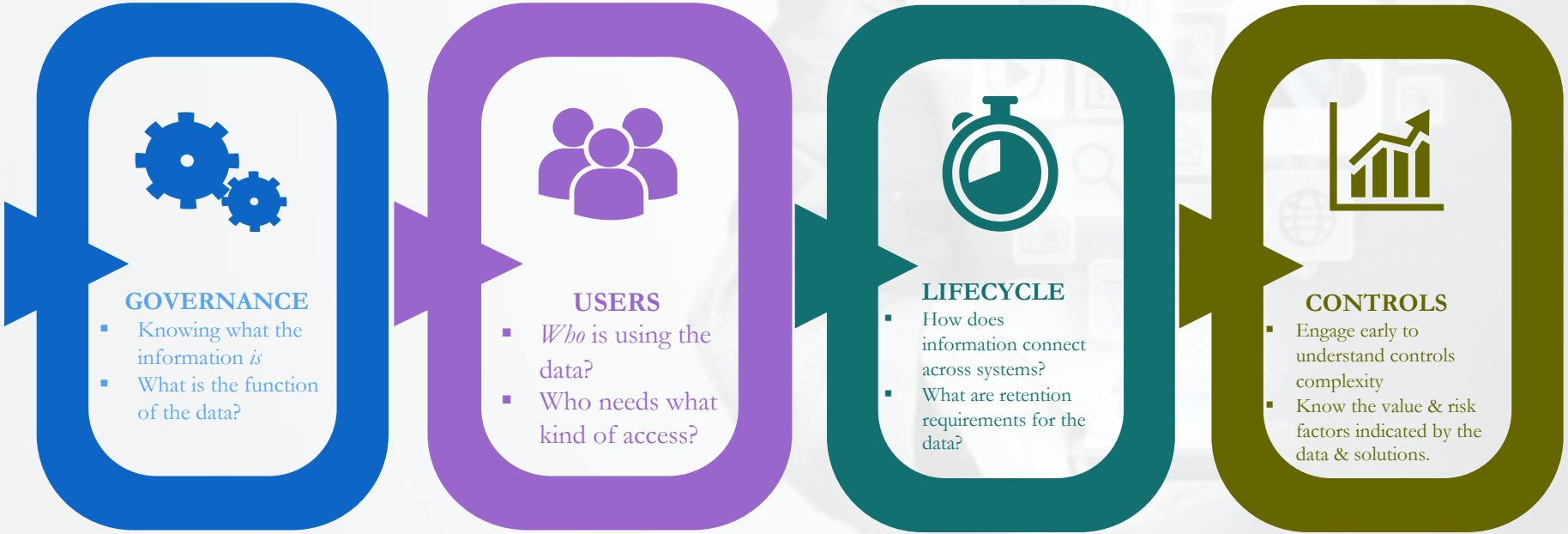


“It’s all about the data.”

Successful implementation of Data Lakes in organizations will demonstrate confidentiality, integrity, and availability across the enterprise.

Achieve Secure Data Enablement

By understanding the key criteria:



Data level hierarchy & OBJECTIVES

Enterprise Level

Enterpriseis the highest level and any data stored at this level is visible / available for all the tenants (geographical data, code sets, etc.)

Tenant Level

To minimize the impact to the existing legacy systems and home-grown services, we will use the additional attributes like “Tenant ID” and “Data Delimiters” to identify which records belong to which tenant. Members can have multiple records in the same system with different Tenant ID’s in case s/he purchased products from more than one tenant.

Domain Level

Application Layer/Domains to control access and/or capabilities (such as LOB, group, segment, or other data restrictions or classifications) within the tenants they use. Application layer to control what the constituent experiences, what data they can access, and how.

Database/Table

Every data set will include audit attributes such as:

- Who is providing the data? ,
- What data is being collected ?,
- When the data is collected?,
- Where the data is collected from?
- Why is the data collected ?

Enterprise level objectives

Enterprise Level Data will…

- Be visible & available to ALL tenants
  - Data Classified, labeled, or segregated in a manner that indicates it has been approved for enterprise wide use which may include Geographical data, code sets, etc.
  - Data Classified as Public
- Support both internal and external users depending on classification
- Internal users get access through an application Id or directly with User Id

Tenant level OBJECTIVES

Enterprise

Tenant Level Data will…

- Support multiple tenants
- Be segregated logically (tagged, labeled, or container segregated based on tenant ID or data delimiters, not physically where possible based on controls objectives for organizations
- Be co-mingled; all applications are storing data together with the following defaults:
  - Logical separation when applicable (controlled by Ranger Policies and data object implementation)
  - Default = Applications (Different Log Locations). Services (Ex; Ranger. Same Log locations).
- Use an additional fields: Tenant ID and Data Delimiters
  - This minimizes impact to existing legacy systems and home-grown services
  - Tenant IDs and Data Delimiters will be used in tables to identify which records belong to which tenant and Enterprise Line of Business.
- Use applications to enforce 100% usage of Tenant IDs and Data Delimiters verified through exceptions, audit & recon

Domain Level OBJECTIVES

Enterprise

Tenant Level

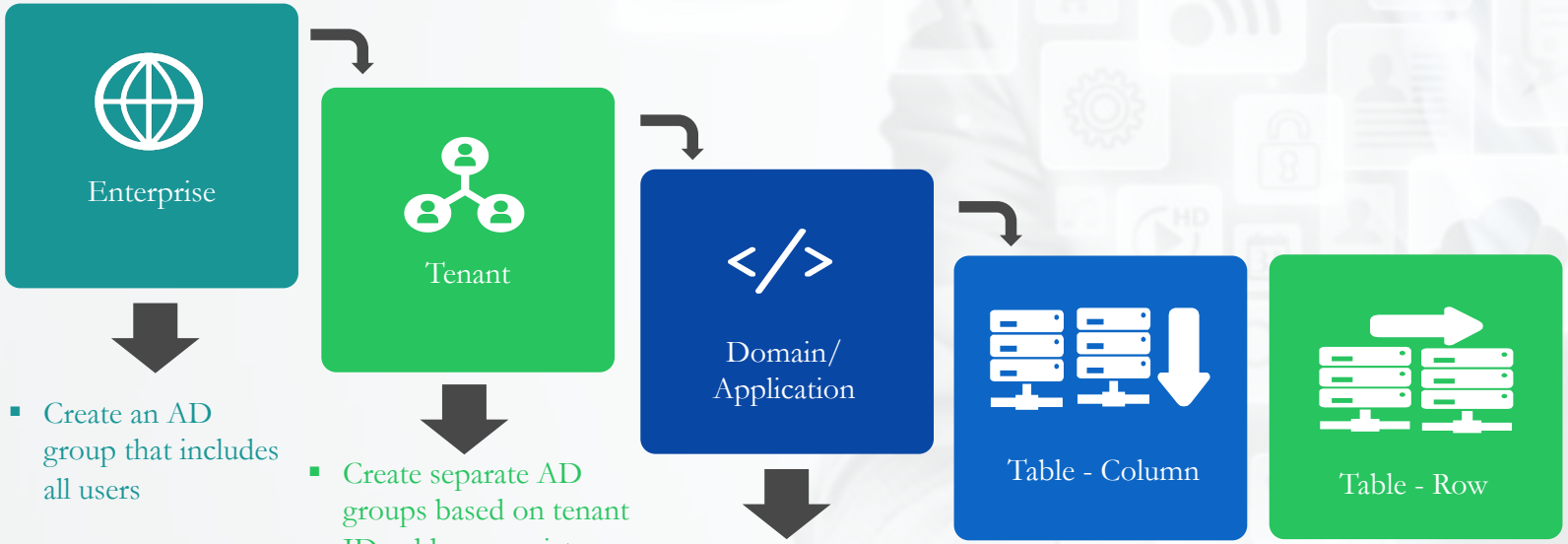
Domain Level Data will…

- Control access and/or capabilities within the tenants they use
- Include application layer that controls what the constituent experiences or what data they may access
  - Also controls *how* the constituent accesses the data

Database Level OBJECTIVES

- ePHI attribute classification and inventory
- User Permissions/Authorizations
- Include audit attributes that answer the following questions for *every* dataset:
  - Who provided the data?
  - What data was collected?
  - When was the data collected?
  - From where is the data collected?
  - Why is the data collected ?
- Data activity monitoring - Who accessed, when accessed, where accessed

Data Handling – Tenant, Domain, Application, Database, Table (Row & Column) Level



- Create an AD group that includes all users
- Resources common across org will be shared across users

- Create separate AD groups based on tenant ID; add appropriate users respectively
- Data gets comingled from different Tenants; Ranger policies control access

- Create separate AD groups for read, read/write, & appropriate users from respectively
- There could be multiple applications as part of domain

- Data in tables could be categorized based on roles, such as accessing data based on column or row level.
- Policies are created for Read and Read/write
- Policies are created at Row and Column level
- Policies are created to mask sensitive data

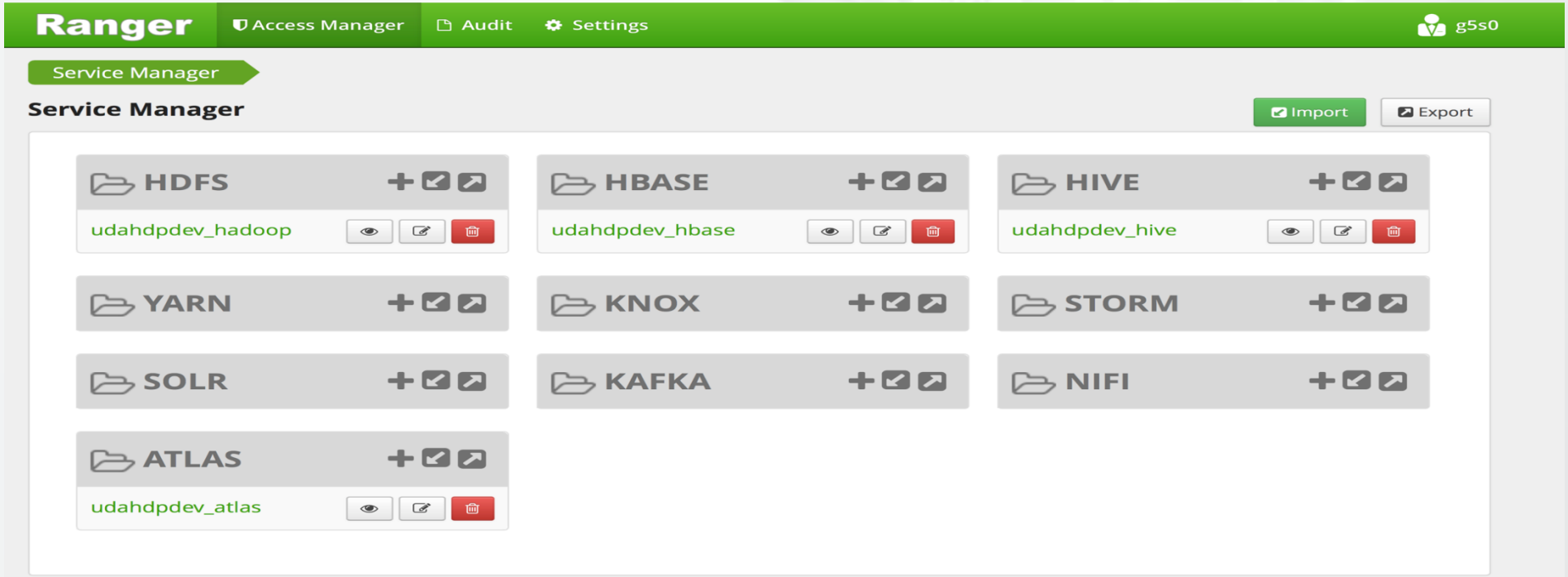
Five Pillars of Security

1. **Administration**
   Central Management & Consistent Security
2. **Authentication**
   Authenticate Users and System
3. **Authorization**
   Provision Access to Data
4. **Data Protection**
   Protect Data at Rest & in Motion
5. **Audit**
   Maintain a record of Data Access

Ranger – Centralized Administration

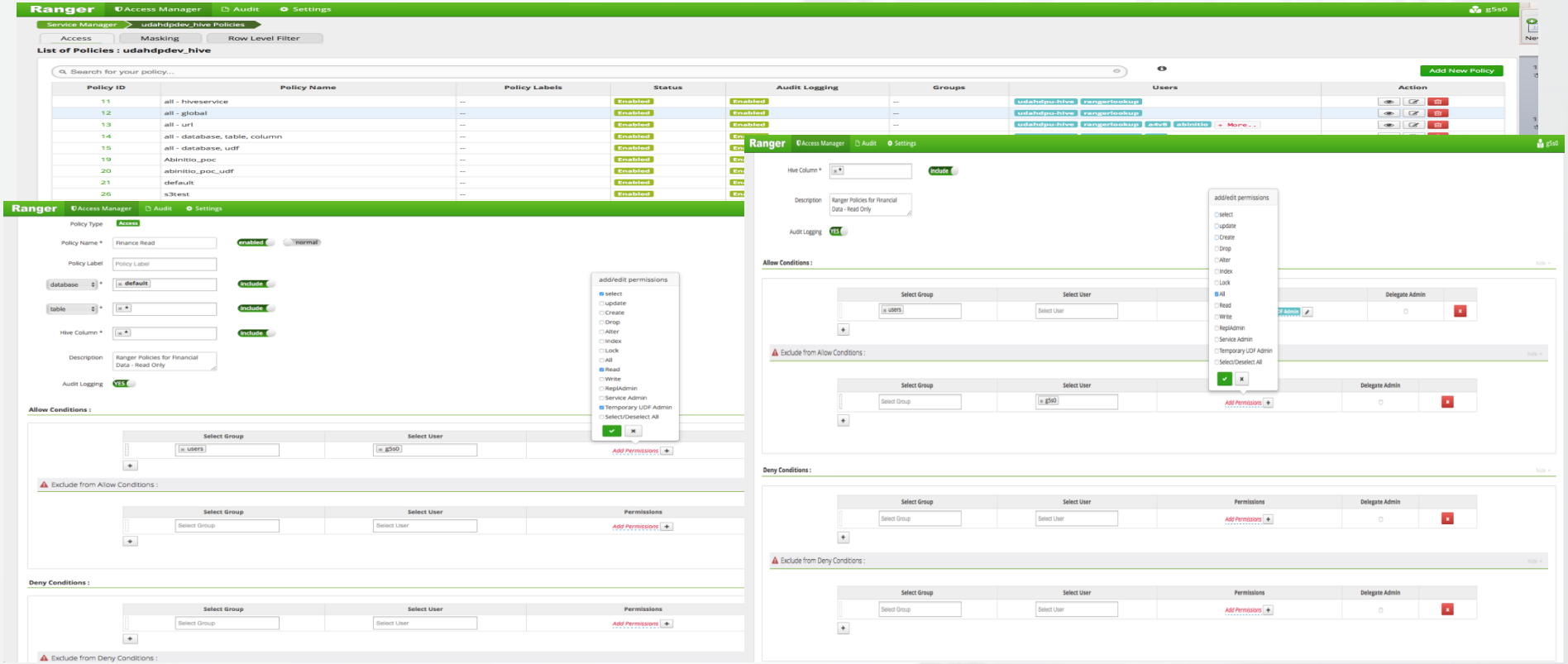
Central Management & Consistent security

Single pane of glass for security administration across multiple Hadoop Components for Creating, implement, Manage and Monitor Security Policies

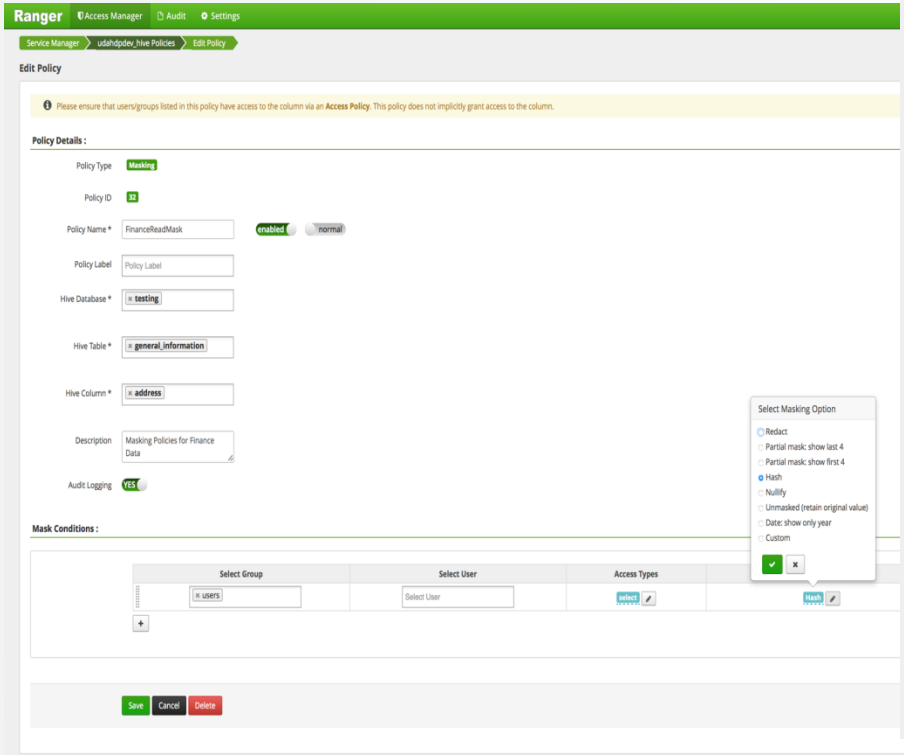


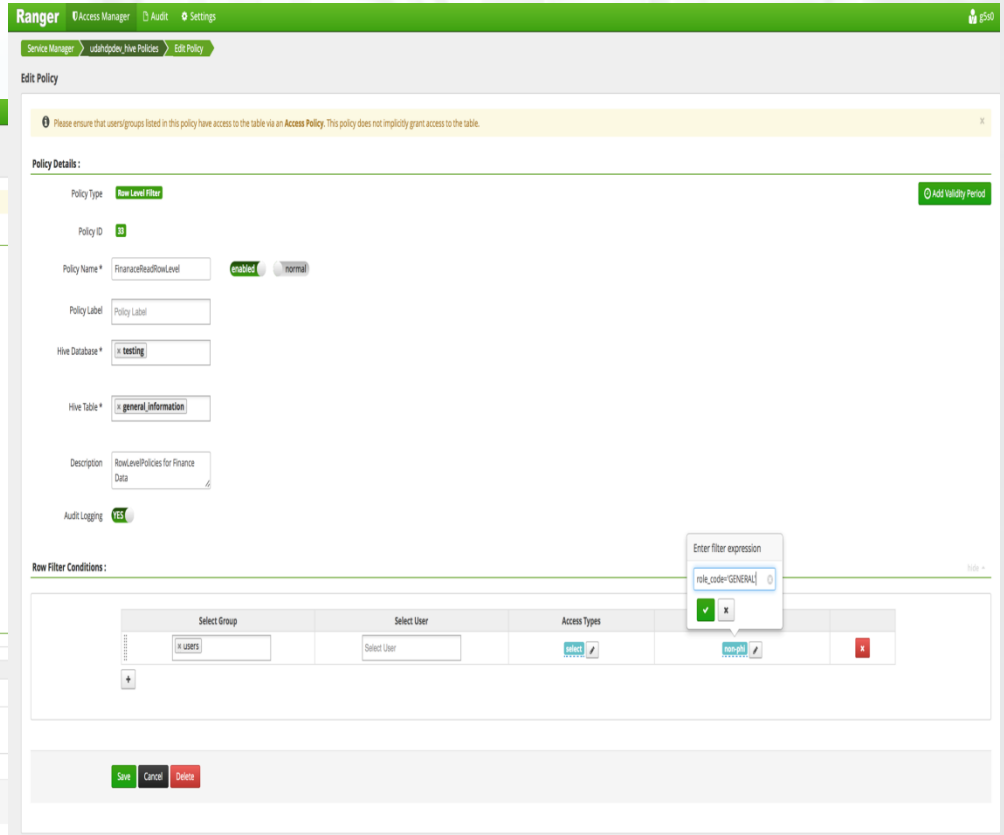
Ranger – Authorization Policies

Consistent authorization policy structure across Hadoop components



Ranger – Row-filter, Column-masking



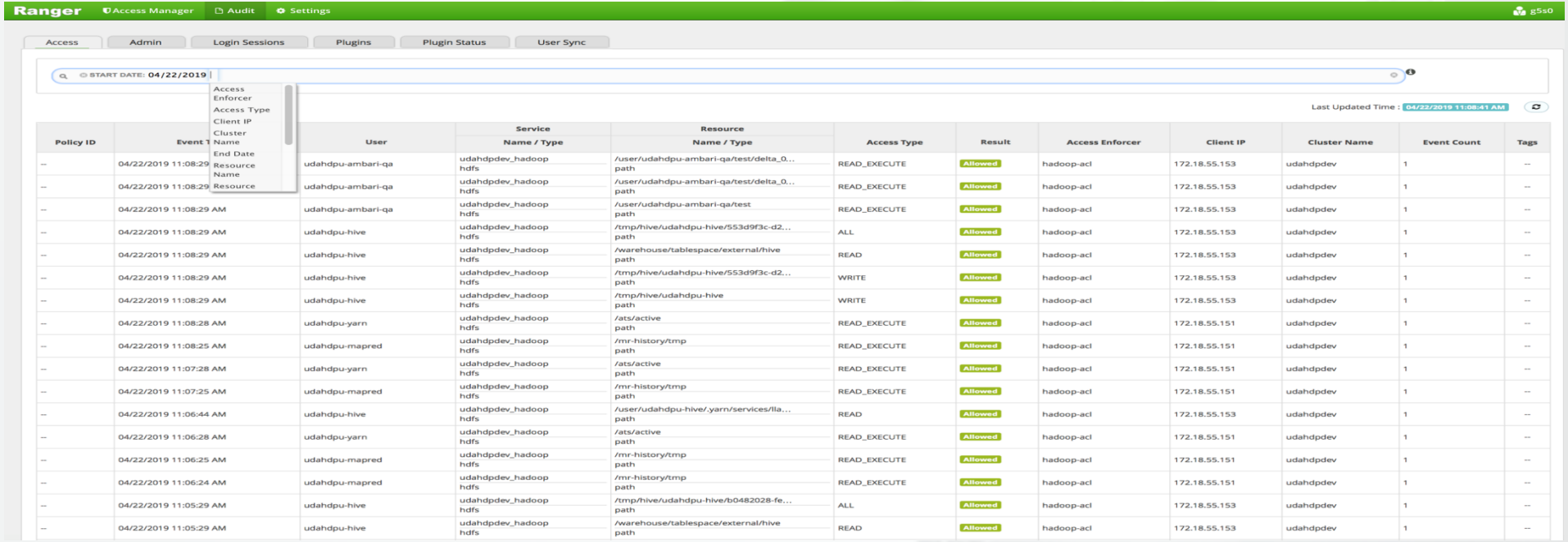


Ranger – Access Audit Logs

Apache Ranger generates detailed logs of access to protected resources

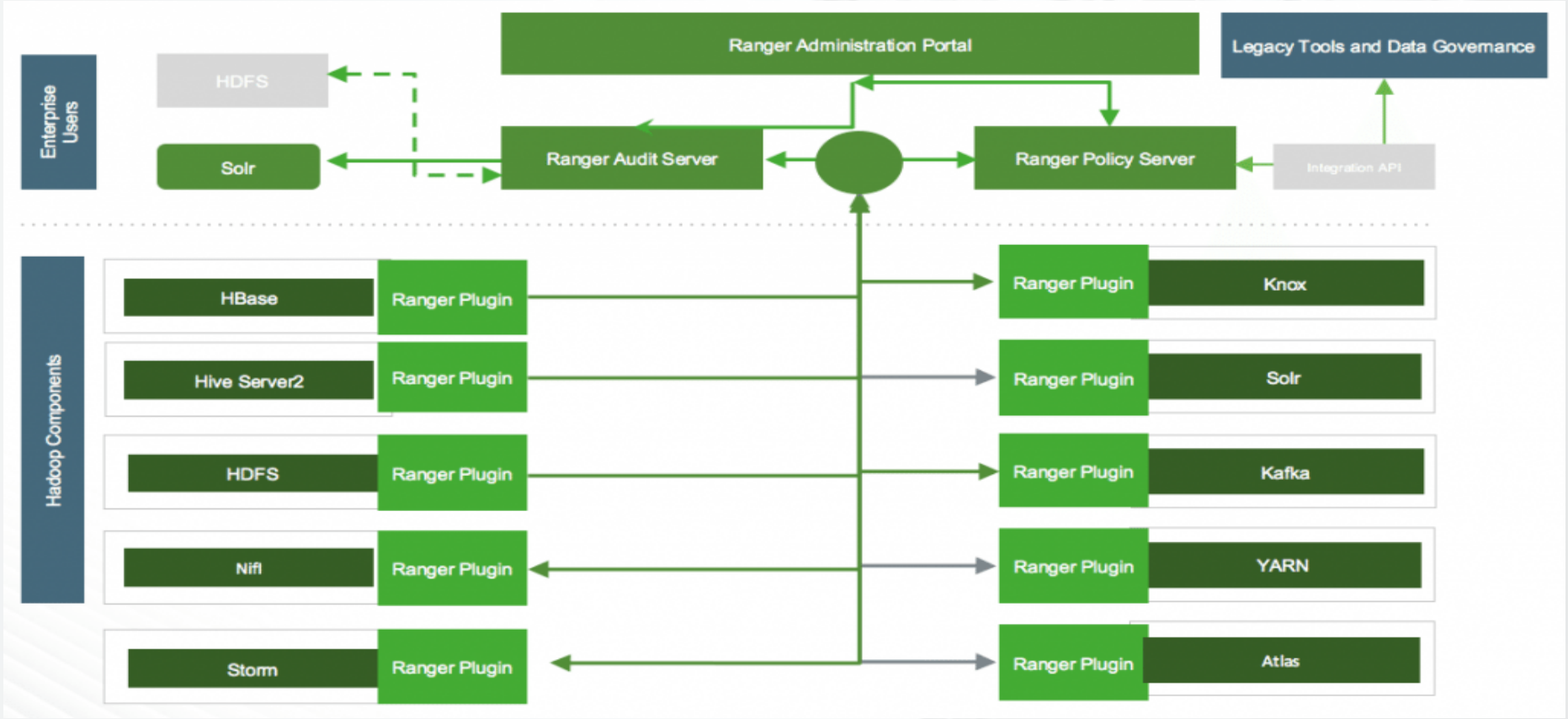
Audit logs to multiple destinations like HDFS, Solr and Log4j appender

Interactive view of audit logs in Admin console



Policy ID	Event T	User	Service Name / Type	Resource Name / Type	Access Type	Result	Access Enforcer	Client IP	Cluster Name	Event Count	Tags
--	04/22/2019 11:08:29	udahdpu-ambari-qa	udahdpdev_hadoop hdfs	/user/udahdpu-ambari-qa/test/delta_0... path	READ_EXECUTE	Allowed	hadoop-acl	172.18.55.153	udahdpdev	1	--
--	04/22/2019 11:08:29	udahdpu-ambari-qa	udahdpdev_hadoop hdfs	/user/udahdpu-ambari-qa/test/delta_0... path	READ_EXECUTE	Allowed	hadoop-acl	172.18.55.153	udahdpdev	1	--
--	04/22/2019 11:08:29 AM	udahdpu-ambari-qa	udahdpdev_hadoop hdfs	/user/udahdpu-ambari-qa/test path	READ_EXECUTE	Allowed	hadoop-acl	172.18.55.153	udahdpdev	1	--
--	04/22/2019 11:08:29 AM	udahdpu-hive	udahdpdev_hadoop hdfs	/tmp/hive/udahdpu-hive/553d9f3c-d2... path	ALL	Allowed	hadoop-acl	172.18.55.153	udahdpdev	1	--
--	04/22/2019 11:08:29 AM	udahdpu-hive	udahdpdev_hadoop hdfs	/warehouse/tablespace/external/hive path	READ	Allowed	hadoop-acl	172.18.55.153	udahdpdev	1	--
--	04/22/2019 11:08:29 AM	udahdpu-hive	udahdpdev_hadoop hdfs	/tmp/hive/udahdpu-hive/553d9f3c-d2... path	WRITE	Allowed	hadoop-acl	172.18.55.153	udahdpdev	1	--
--	04/22/2019 11:08:29 AM	udahdpu-hive	udahdpdev_hadoop hdfs	/tmp/hive/udahdpu-hive path	WRITE	Allowed	hadoop-acl	172.18.55.153	udahdpdev	1	--
--	04/22/2019 11:08:28 AM	udahdpu-yarn	udahdpdev_hadoop hdfs	/ats/active path	READ_EXECUTE	Allowed	hadoop-acl	172.18.55.151	udahdpdev	1	--
--	04/22/2019 11:08:25 AM	udahdpu-mapred	udahdpdev_hadoop hdfs	/mr-history/tmp path	READ_EXECUTE	Allowed	hadoop-acl	172.18.55.151	udahdpdev	1	--
--	04/22/2019 11:07:28 AM	udahdpu-yarn	udahdpdev_hadoop hdfs	/ats/active path	READ_EXECUTE	Allowed	hadoop-acl	172.18.55.151	udahdpdev	1	--
--	04/22/2019 11:07:25 AM	udahdpu-mapred	udahdpdev_hadoop hdfs	/mr-history/tmp path	READ_EXECUTE	Allowed	hadoop-acl	172.18.55.151	udahdpdev	1	--
--	04/22/2019 11:06:44 AM	udahdpu-hive	udahdpdev_hadoop hdfs	/user/udahdpu-hive/.yarn/services/lla... path	READ	Allowed	hadoop-acl	172.18.55.153	udahdpdev	1	--
--	04/22/2019 11:06:28 AM	udahdpu-yarn	udahdpdev_hadoop hdfs	/ats/active path	READ_EXECUTE	Allowed	hadoop-acl	172.18.55.151	udahdpdev	1	--
--	04/22/2019 11:06:25 AM	udahdpu-mapred	udahdpdev_hadoop hdfs	/mr-history/tmp path	READ_EXECUTE	Allowed	hadoop-acl	172.18.55.151	udahdpdev	1	--
--	04/22/2019 11:06:24 AM	udahdpu-mapred	udahdpdev_hadoop hdfs	/mr-history/tmp path	READ_EXECUTE	Allowed	hadoop-acl	172.18.55.151	udahdpdev	1	--
--	04/22/2019 11:05:29 AM	udahdpu-hive	udahdpdev_hadoop hdfs	/tmp/hive/udahdpu-hive/b0482028-fe... path	ALL	Allowed	hadoop-acl	172.18.55.153	udahdpdev	1	--
--	04/22/2019 11:05:29 AM	udahdpu-hive	udahdpdev_hadoop hdfs	/warehouse/tablespace/external/hive path	READ	Allowed	hadoop-acl	172.18.55.153	udahdpdev	1	--

Ranger – Architecture



Questions