

Apache Sentry

Prasad Mujumdar
prasadm@apache.org
prasadm@cloudera.com



Agenda

- Various aspects of data security
- Apache Sentry for authorization
- Key concepts of Apache Sentry
- Sentry features
- Sentry architecture
- Integration with Hadoop ecosystem
- Sentry administration
- Future plans
- Demo
- Questions

Who am I

- Software engineer at Cloudera
- Committer and PPMC member of Apache Sentry
- also for Apache Hive and Apache Flume
- Part of the the original team that started Sentry work

Aspects of security

Perimeter

Authentication

Kerberos, LDAP/AD

Access

Authorization

what user can do
with data

Visibility

Audit, Lineage

data origin, usage

Data

Encryption,

Masking

Data access


Access

Authorization

**what user can do
with data**

- Provide user access to data
- Manage access policies
- Provide role based access

Agenda

- Various aspects of data security
- Apache Sentry for authorization 
- Key concepts of Apache Sentry
- Sentry features
- Sentry architecture
- Integration with Hadoop ecosystem
- Sentry administration
- Future plans
- Demo
- Questions

Apache Sentry (Incubating)

Unified Authorization module for Hadoop

Unlocks Key RBAC Requirements

Secure, fine-grained, role-based authorization

Multi-tenant administration

Enforce a common set of policies across multiple data access path in Hadoop.



Key Capabilities of Sentry

Fine-Grained Authorization

Permissions on object hierarchie. Eg, Database, Table, Columns

Role-Based Authorization

Support for role templetetes to manage authorization for a large set of users and data objects

Multi Tanent Administration

Ability to delegate admin responsibilities for a subset of resources




Project history and status

- Started at Cloudera
- Entered incubation in 2013
- Growing community
 - Committers from Cloudera, IBM, Intel, Oracle, ...
 - Three releases from incubation
- Widely adopted by industry
 - Part of multiple commercial Hadoop distros



Agenda

- Various aspects of data security
- Apache Sentry for authorization
- Key concepts of Apache Sentry ← 
- Sentry features
- Sentry architecture
- Integration with Hadoop ecosystem
- Sentry administration
- Future plans
- Demo
- Questions

Key Concepts in Sentry

- Global concepts
 - User, Group, Role, Privilege
- Authorization Models
 - SQL
 - Server, Database, Table, URI
 - Search Model
 - Collection

Global Concept: User

- Individual person
 - Runs SQL, SOLR queries
- Identified by authentication provider
 - Kerberos, LDAP etc
- Just a string for Sentry
 - Not enforcing existence
 - Sentry is NOT an authentication system



Global Concept: Group

- Set of users
 - Same needs/privileges
- Plugable group mapping
 - Using Hadoop Groups
 - OS, LDAP, Active Directory



Global Concept: Privilege

- Unit of data access
- Tuple
 - Object
 - Action
- Always positive

READ TABLE logs

READ DATABASE prod

WRITE and READ TABLE logs

QUERY COLLECTION logs

UPDATE COLLECTION admin

Global Concept: Role

- Set of privileges
 - Functional template
- Unit of grant

Analyst

Analyst Junior

Warehouse admin

Warehouse user


Project X

Global Concepts: Relations

- Groups have multiple users
- Role have multiple privileges
- Roles are assigned to groups
 - Sentry does not support direct grants to user
- No jumping
 - User to role, group to privilege, ...



Agenda

- Various aspects of data security
- Apache Sentry for authorization
- Key concepts of Apache Sentry
- Sentry features 
- Sentry architecture
- Integration with Hadoop ecosystem
- Sentry administration
- Future plans
- Demo
- Questions

Sentry features – Fine grain authorization

- Privileges at various levels for resource hierarch
 - Eg Database, Table and Column for SQL model
 - Read or Select access on Database implicitly grant access on child tables
- Supports different actions on resources
 - Eg, Select, Insert, Create, Alter in the SQL model
 - Query, Update in Search model
 - ...


Sentry Features – Role based

- Supports Role as collection of permission
 - Template for a functional access rules
 - Eg, Analyst role → Read table sales, Read table customer, Admin of Sandbox
 - Makes auth administration manageable in large and complex deployments
- Allows granting roles to groups
 - A role can be granted to a large set of users in a single operation
 - Easier integration with existing identity management systems like AD
- Onboarding and removing users is lot simpler with roles and groups

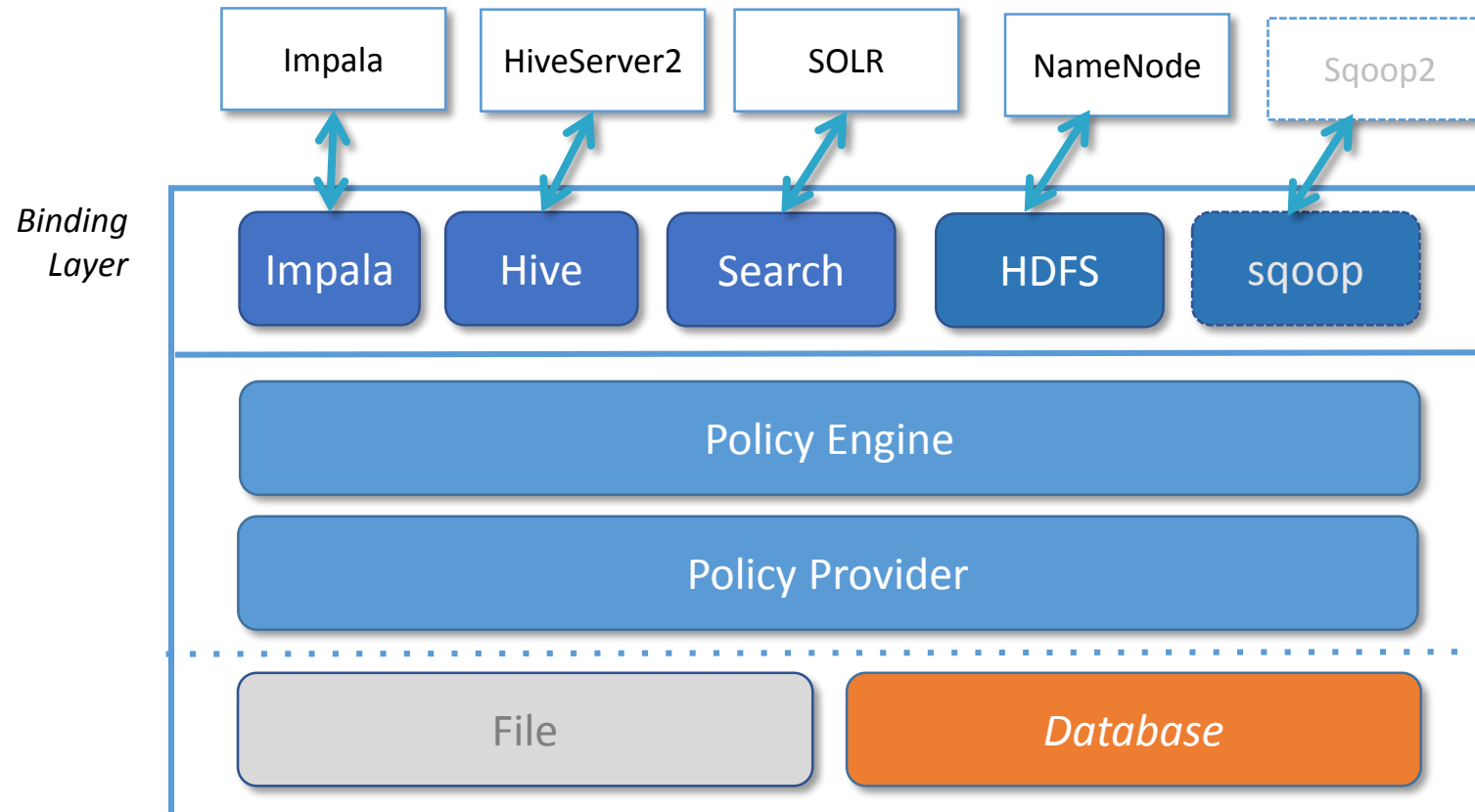
Sentry features – misc

- Multi Tenant administration
 - Ability to delegate admin access for a subset of resources
 - Eg. A user can be an admin of his/her own sandbox database
- Pluggable architecture
 - A new authorization model can be implemented with little code changes
 - Can easily integration with new identity management systems for groups
 - Supports various callbacks for custom monitoring

Agenda

- Various aspects of data security
- Apache Sentry for authorization
- Key concepts of Apache Sentry
- Sentry features
- Sentry architecture 
- Integration with Hadoop ecosystem
- Sentry administration
- Future plans
- Demo
- Questions

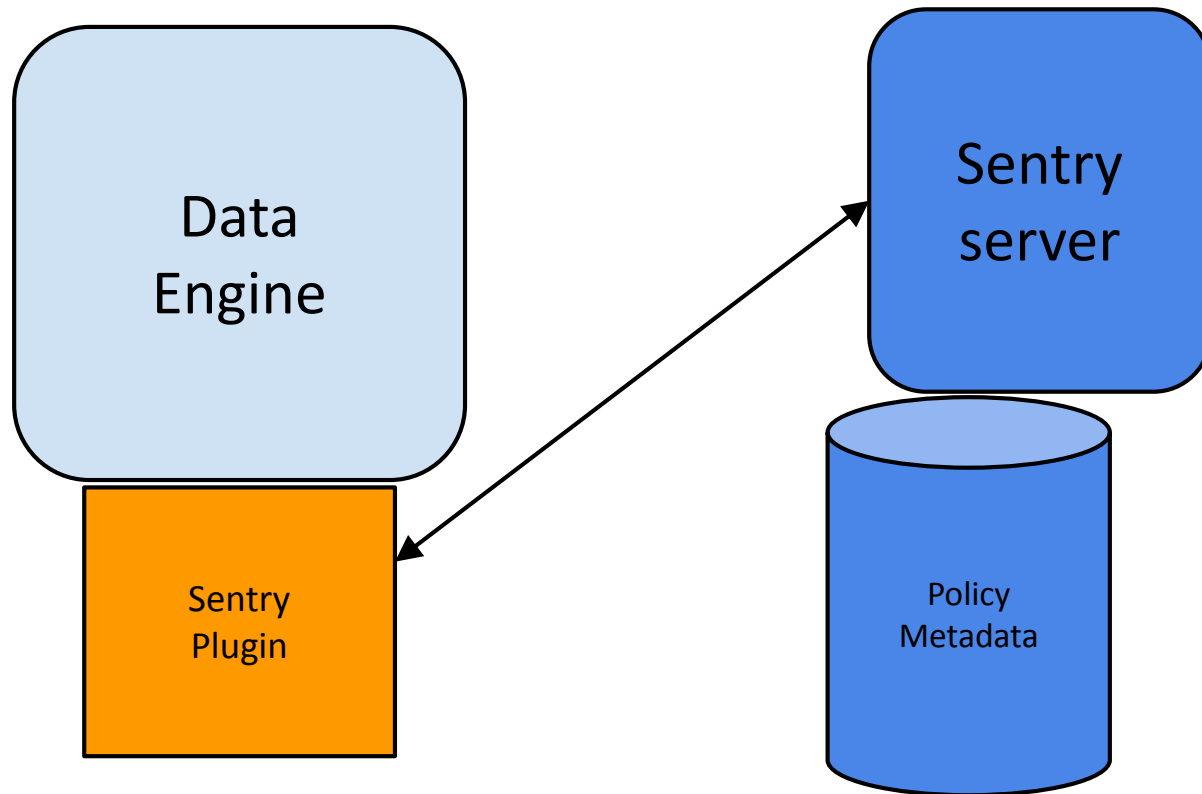
Apache Sentry conceptual overview



Apache Sentry conceptual overview

- Policy Provider
 - Abstraction for loading and manipulating privilege metadata
 - Support for external DB backed storage (default)
 - Also support local or HDFS file storage (deprecated)
- Policy Engine
 - Makes the authorization decision
 - Reads the metadata from policy provider
- Binding
 - Bridging layer between the downstream service and Sentry
 - Handles translating the native access request into Sentry APIs

Sentry Service Architecture

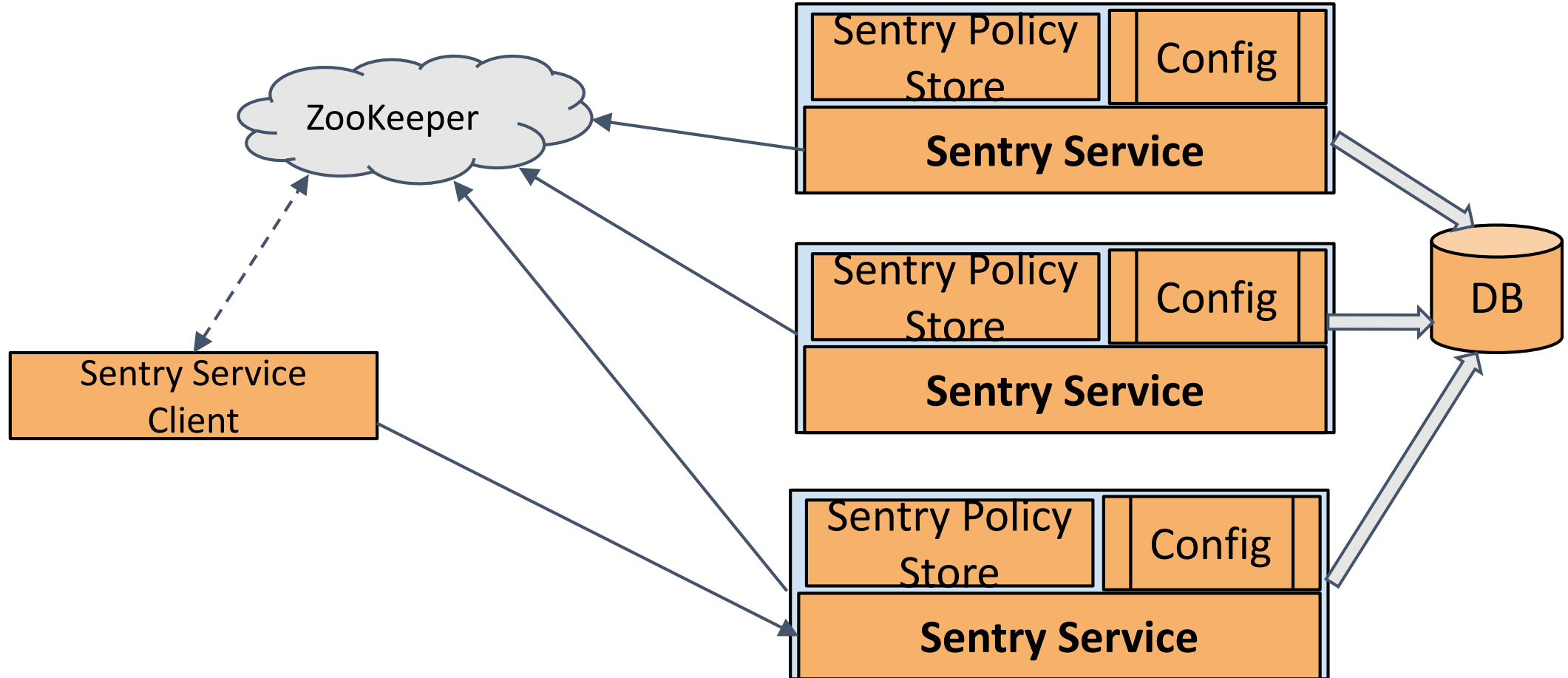


- Data Engine, eg Hive
- Sentry plugin
- Sentry RPC server
- Policy metadata store

Sentry Service

- RPC Service to manage metadata
 - Apache Thrift RCP implementation
 - Java client
 - Secured with kerberos
- API to retrieve and manipulate policies
- Metadata stored in external backend DB
 - Supports Derby, MySQL, Postgres, Oracle and DB2

Sentry Service HA



Sentry Service HA

- Active/Active HA
- Each service registers with ZK
- Client first retrieves service address for ZK
- User Apache Curator framework

File based privileged metadata

- Policy information can be stored in local or HDFS files
- Deprecated in newer releases in favor of DB based policies
- ini format property file

```
# group to role mapping
[groups]
manager = analyst_role, junior_analyst_role
analyst = analyst_role
admin = admin_role

# role to privilege mapping
[roles]
analyst_role = server=server1->db=analyst1, \
server=server1->db=jranalyst1->table=*->action=select, \
server=server1->db=default->table=tab2
```


Sentry Client Plugin

- Client side piece of Sentry
 - Integrates via the authorization interfaces
- Responsible for authorization decision
 - Receives requested resources and user from caller
 - Retrieves relevant privileges from Sentry service
 - Evaluates the request

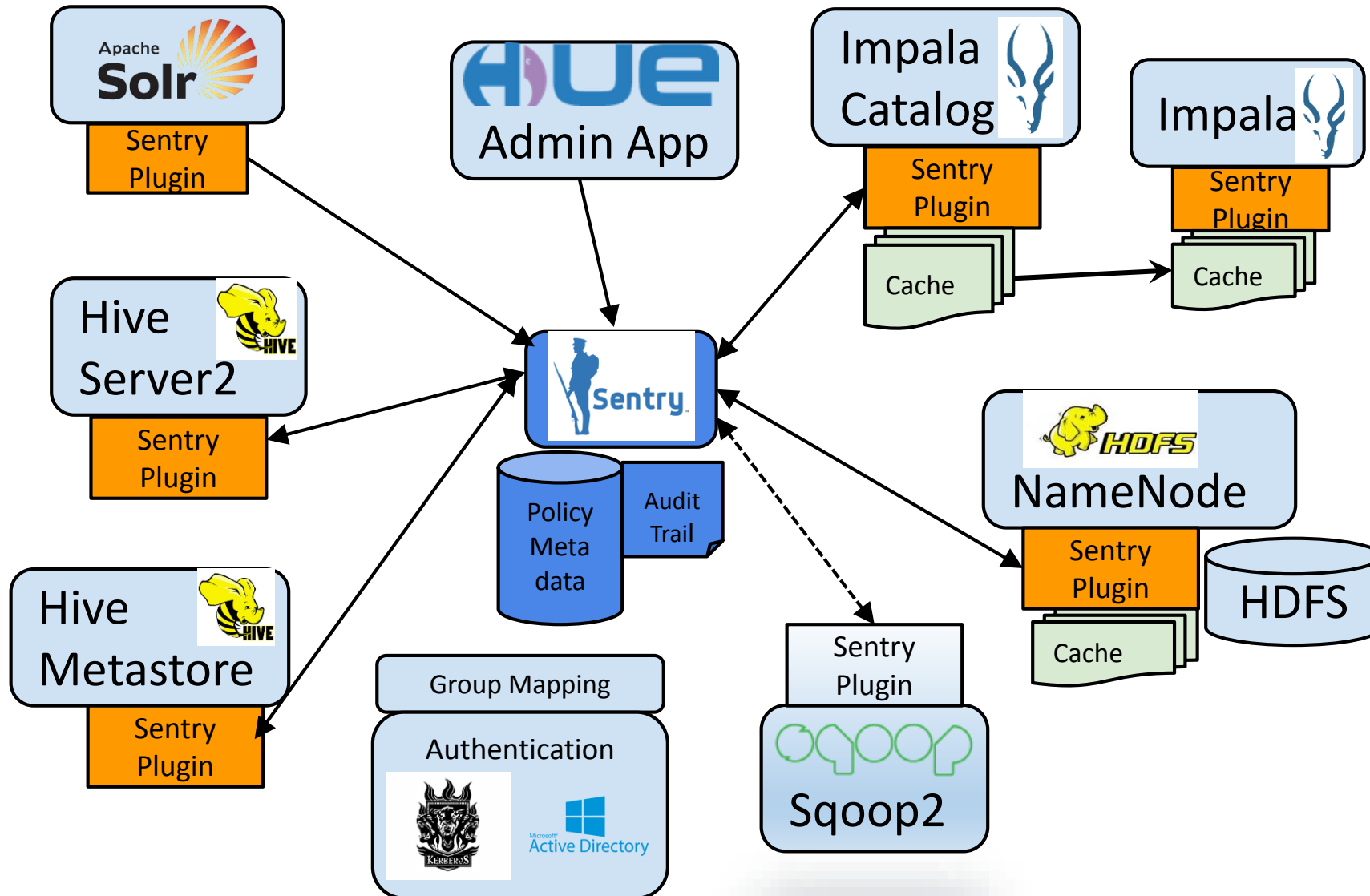
Auditing

- Sentry service generates audit trails
- Policy changes are audited
 - Eg granting privileges, create/drop roles
- Audit JSON format audit log
 - Easier for processing by audit reporting tools
- Client side auditing handled by client's auth auditing mechanism
 - Eg Hive and Impala
 - Sentry support client callbacks which can be used for customization

Agenda

- Various aspects of data security
- Apache Sentry for authorization
- Key concepts of Apache Sentry
- Sentry features
- Sentry architecture
- Integration with Hadoop ecosystem 
- Sentry administration
- Future plans
- Demo
- Questions

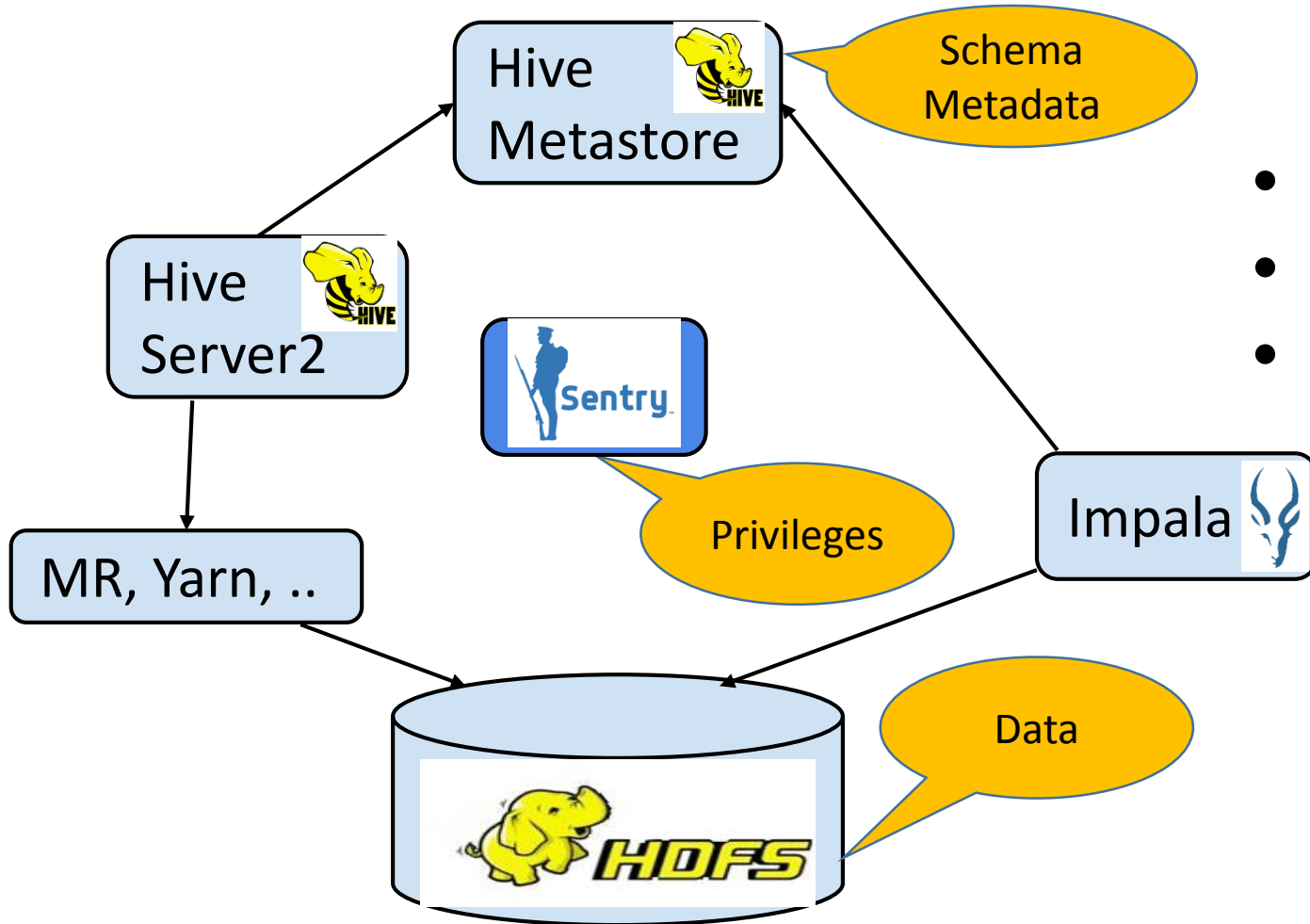
Integration with Hadoop Ecosystem



Unified authorization for Hadoop ecosystem

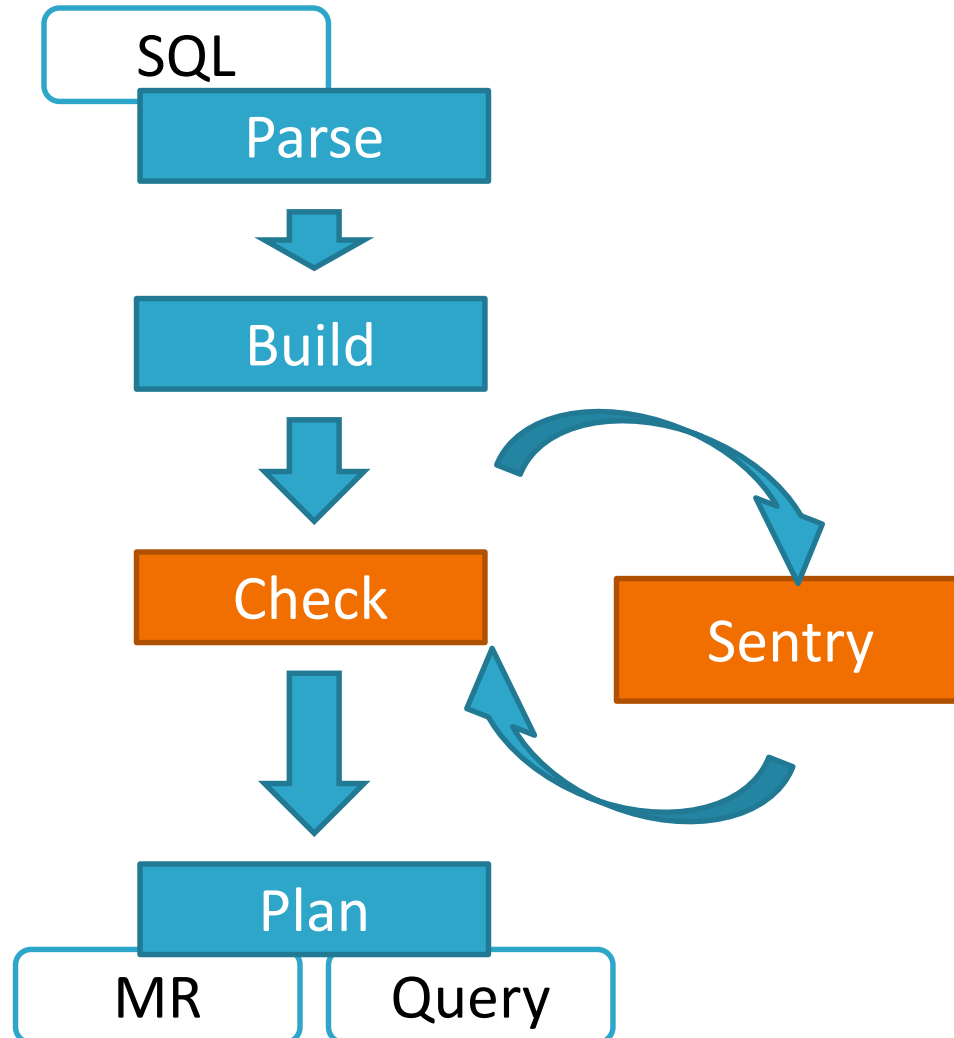
- Single source of truth
- Other projects don't have to implement their own auth
- Same set of roles and group available across tools
 - Makes the authorization administration lot simpler
- Same privileges enforced irrespective of the access path

SQL on Hadoop



- *Data on HDFS, owned by Hive*
- *Metadata in Hive Metastore*
- *Auth policies in Sentry*

Sentry with Apache Hive

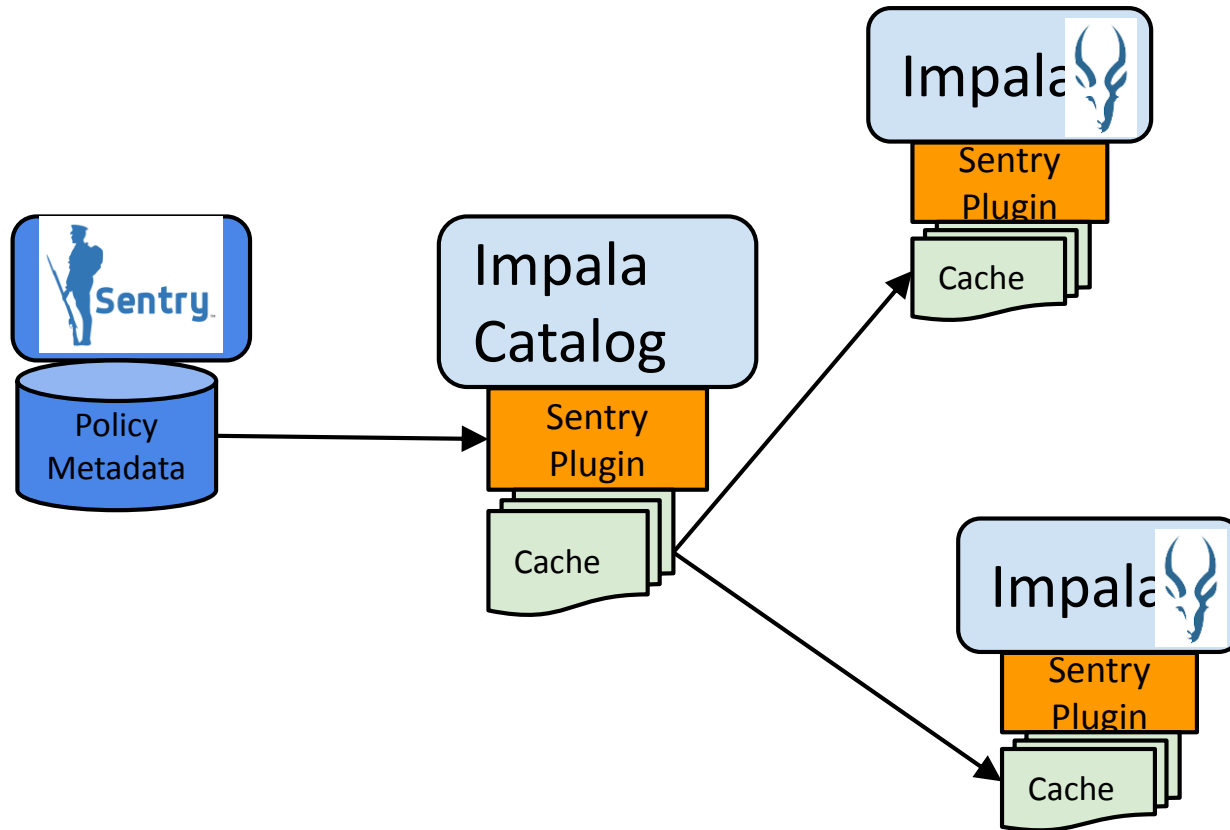


Validates access to SQL entities before executing the query.

Sentry with Apache Hive

- Requires HiveServer2, not supported with thick hive client
- SQL model with fine grained authorization
 - DB objects - Database, Table and Column
 - DB Actions - SELECT, INSERT, CREATE, ALTER, ..
- Support managed and external tables
 - Special handling of external path specification via URI level privilege
- Authorization administration via SQL
 - grant, revoke, create/drop role etc.

Sentry with Impala



Validates access to SQL entities before executing the query using the cached privileges.

Sentry with Impala

- Uses same SQL model as Hive
- Fine grained authorization
 - DB objects - Database, Table, Column
 - DB Actions - SELECT, INSERT, CREATE, ALTER, ..
- Support managed and external tables
 - Special handling of external path specification via URI level privilege
- Impala engine caches privilege metadata for faster access

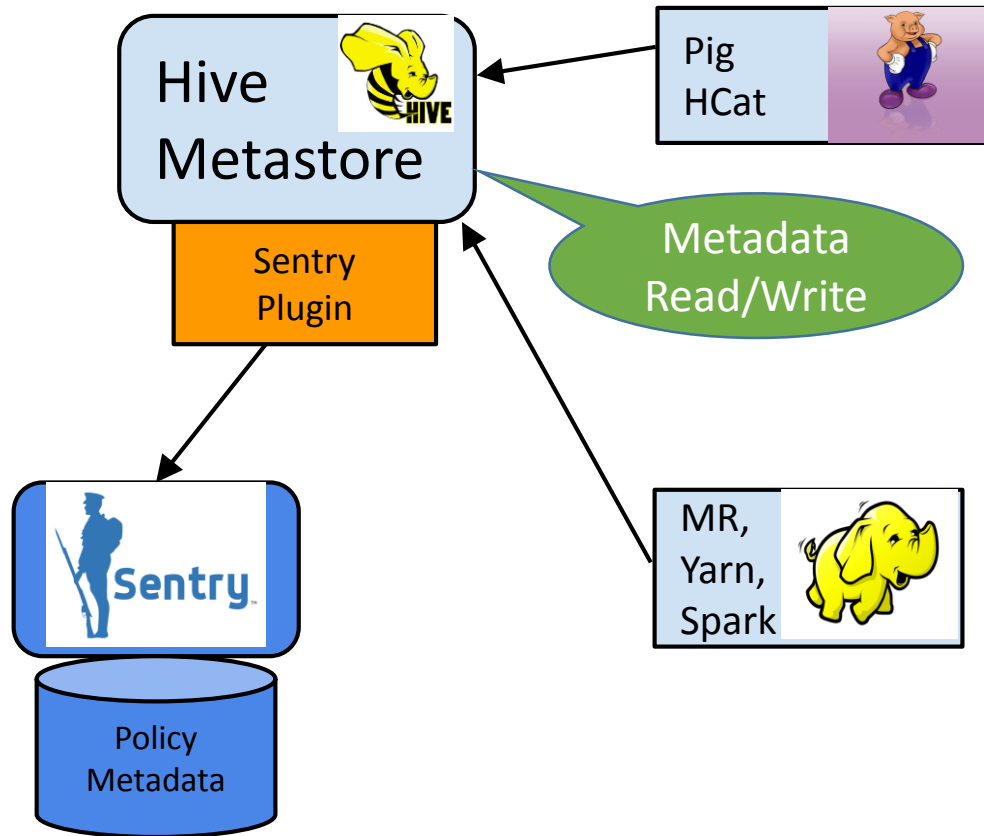
View level privileges for SQL authorization

- Views are essentially queries defined on one or more tables
 - Eg `CREATE VIEW v1 AS SELECT tab1.col1, tab2.col2 FROM tab1, tab2 ...`
- Privileges on views are independent of the base tables
- This enables row/cell level privileges
- Requires data files to be owned and access by Hive user

URI level privilege

- Hive SQL supports file URI leading to security loopholes
 - Alternate storage path for tables
 - Create table
 - Alter table
 - External table
 - One can specify the path of a different table and bypass authorization
 - *ALTER TABLE sandbox.sales SET LOCATION '/user/hive/warehouse/production/sales'*
- Hive UDFs using jars with untrusted/unauthorized static code
- URI resource privilege to can be used to prevent this
 - A file URI can only be used if you have explicit grant to use it

Sentry with Metastore

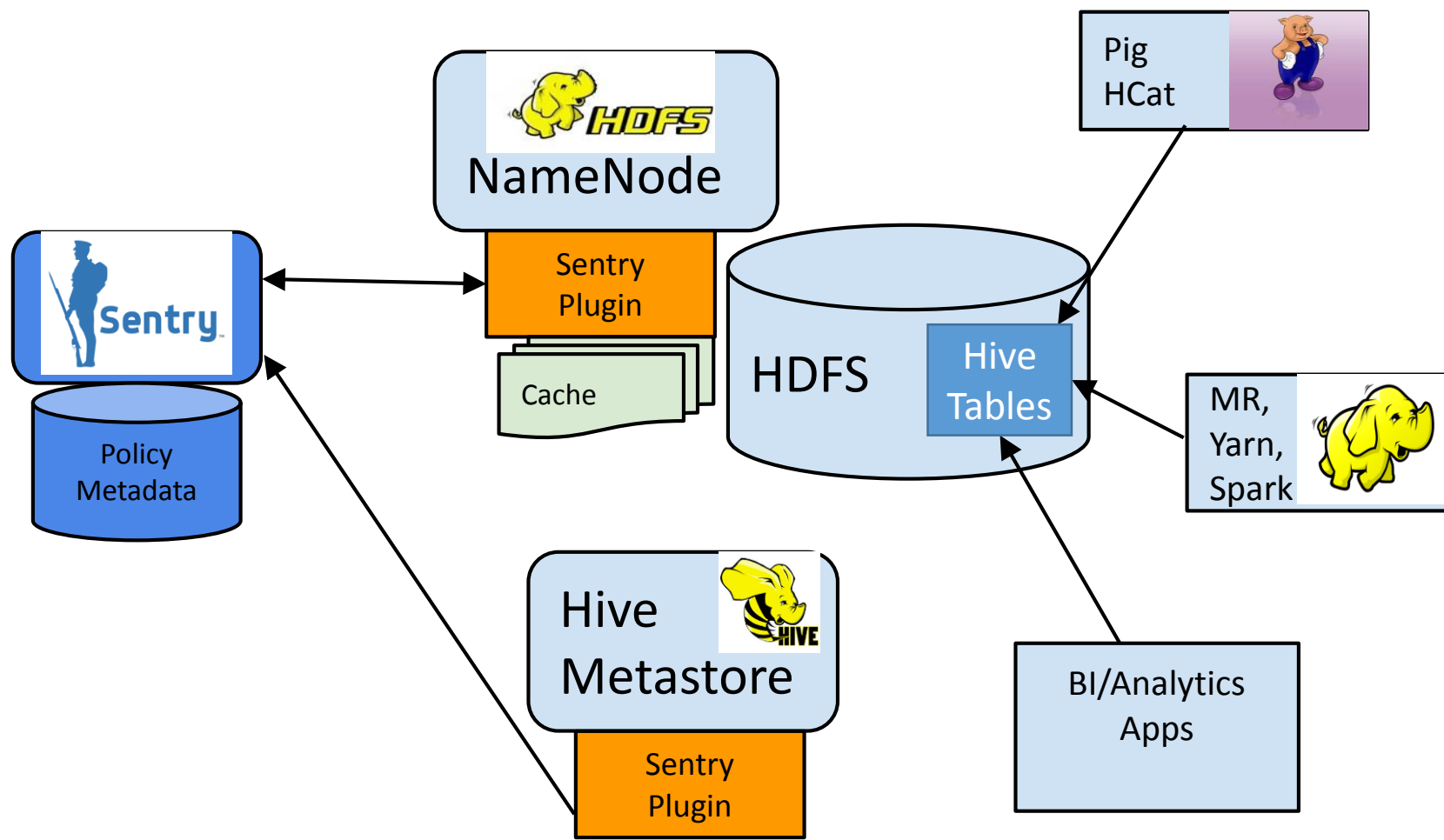


Metastore RPC clients can read/write metadata directly. Sentry enforces the same privileges on metadata

Sentry with Metastore

- Enforces the same policies for metadata access
- Prevents unauthorized schema changes
- Hides metadata from unauthorized users
- Works for all Metastore RPC clients
 - Apache Pig with Hcatalog
 - Hadoop jobs
 - Third party applications

Sentry HDFS ACL syncn

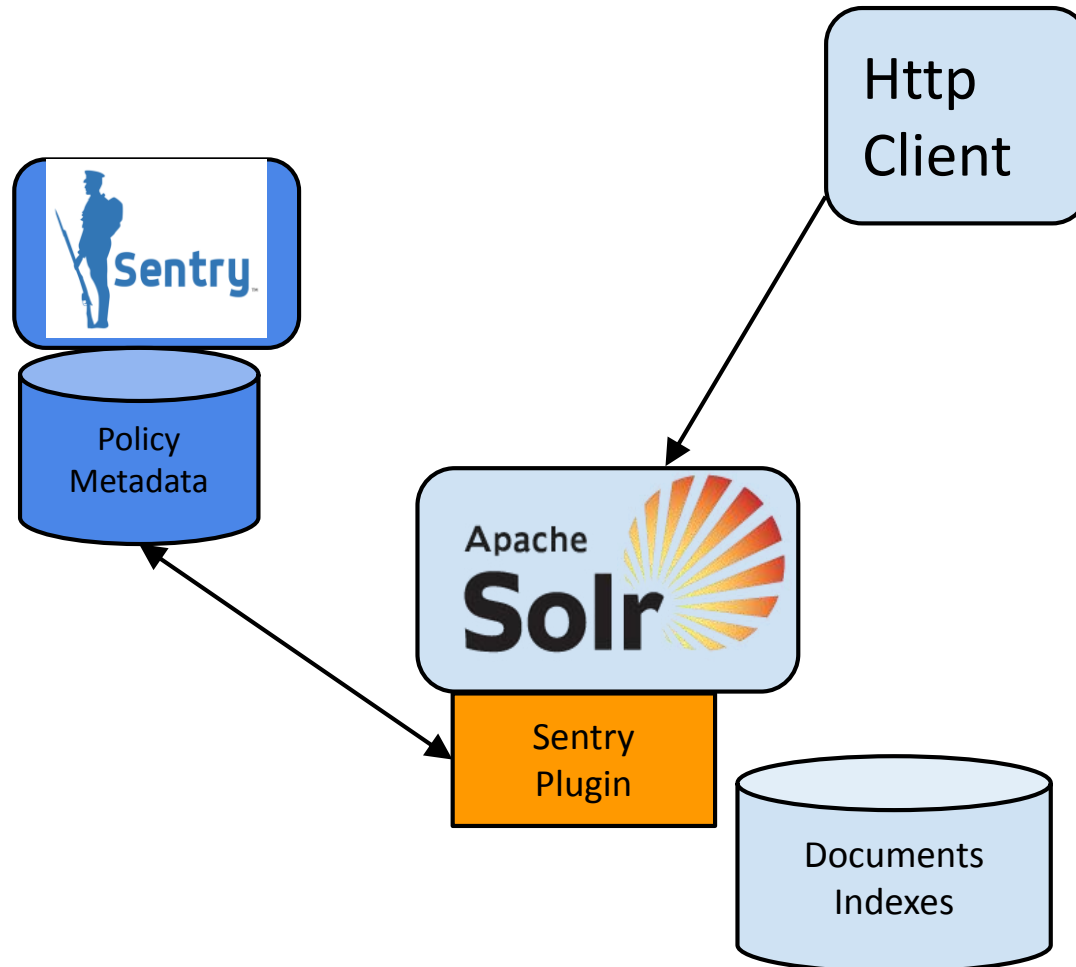


HDFS applies Sentry privileges as ACLs for files/directories that are part of Hive data to enable non-SQL clients.

HDFS ACL syncn for non-sql clients

- Apply Sentry privileges as HDFS ACLs
 - Requires HDFS extended ACLs enabled
- Namenode maintains a cache of privileges
- Currently supported for Hive data only
- Enables same granularity of access to files for non-sql clients
- *Hadoop side changes is recently committed only available in trunk*

Sentry with Apache Solr



Validate access to Solr collection and documents.


Sentry with Apache Solr

- Fine grained authorization
 - Collection
 - Documents
 - Index
- Support query and update access on the resources

Sentry with Apache Sqoop

- Authorization of various sqoop resources
 - connector, link, jobs
- Fine grained authorization of actions
 - Create, Enable, Start/Stop, List etc.
- Under development
 - SENTRY-612 being reviewed

Agenda

- Various aspects of data security
- Apache Sentry for authorization
- Key concepts of Apache Sentry
- Sentry features
- Sentry architecture
- Integration with Hadoop ecosystem
- Sentry administration 
- Future plans
- Demo
- Questions

Sentry Administration

- Privileges managed natively by downstream app
 - Auth SQL statements
 - Application APIs
- Hue UI
 - Sentry App for policy administration
- Pluggable groups mapping
 - By default same as Hadoop (OS or LDAP/AD)

Sentry App in Hue

The screenshot displays the Hue web interface for configuring Sentry permissions. The top navigation bar includes 'HUE', 'Query Editors', 'Data Browsers', 'Workflows', 'Search', 'Security', 'File Browser', 'Job Browser', and 'hive'. The main header shows 'Hadoop Security' with sub-tabs for 'Sentry Tables' and 'File ACLs'. On the left, a sidebar contains 'PRIVILEGES' (with 'Browse' and 'Roles' options) and 'GROUPS' (with a search input). The main content area is titled 'Database and Table privileges' and features a search box containing 'production'. Below the search box is a tree view of databases under 'server1', with 'production' selected. The 'production' database contains tables: 'customer_info', 'customers', 'products', and 'transaction'. To the right, the 'Privileges' section shows a role named 'retail_manager_role' with a timestamp '10 minutes ago DATABASE' and the privilege configuration 'server=server1 → db=production → action=ALL'. A plus sign indicates that more privileges can be added.

Sentry App in Hue


The screenshot displays the Hue interface for the Sentry App. The top navigation bar includes 'HUE', 'Query Editors', 'Data Browsers', 'Workflows', 'Search', 'Security', 'File Browser', 'Job Browser', and 'hive'. The main navigation area shows 'Hadoop Security' with sub-tabs for 'Sentry Tables' and 'File ACLs'. A left sidebar contains 'PRIVILEGES' (Browse, Roles), 'GROUPS' (ANALYST), and a search bar. The main content area is titled 'Roles' and features a search box and an 'Expand' button. Two roles are listed: 'hc_analyst_role' and 'retail_analyst_role', both assigned to the 'analyst' group. Each role has a list of permissions, such as 'SERVER' and 'TABLE' permissions on various servers, databases, and tables.

<input type="checkbox"/>	Name	Groups
<input type="checkbox"/>	hc_analyst_role	analyst
	18 minutes ago SERVER server=server1 → file://demo/data/patient_data → action=ALL	
	18 minutes ago TABLE server=server1 → db=healthcare → table=customer_info → action=SELECT	
	18 minutes ago TABLE server=server1 → db=healthcare → table=patient_data → action=SELECT	
	+	
<input type="checkbox"/>	retail_analyst_role	analyst
	18 minutes ago TABLE server=server1 → db=production → table=customer_info → action=SELECT	
	18 minutes ago TABLE server=server1 → db=production → table=products → action=SELECT	
	18 minutes ago TABLE server=server1 → db=production → table=products → action=INSERT	
	18 minutes ago TABLE	

Setting up Sentry in Hadoop cluster

- Should have strong authentication like Kerberos or LDAP
- Setup sentry service
 - Setup metadata DB
 - Configure and run the service
- Setup data services to use Sentry
 - Configure auth plugins
 - Setup sentry client configuration to use sentry service
- Create roles and privileges
 - Hue UI app is super useful

Agenda

- Various aspects of data security
- Apache Sentry for authorization
- Key concepts of Apache Sentry
- Sentry features
- Sentry architecture
- Integration with Hadoop ecosystem
- Sentry administration
- Future plans 
- Demo
- Questions

Future plans

- Integration with more Hadoop ecosystem components
 - Hbase
 - Kafka, Flume ..
- Attribute based access control
- Row/cell level authorization for HDFS

References

- Project page
 - <https://sentry.incubator.apache.org>
- Wiki
 - <https://cwiki.apache.org/confluence/display/SENTRY/Home>
- Source
 - git clone <http://git-wip-us.apache.org/repos/asf/incubator-sentry.git>
- Downloads
 - <https://sentry.incubator.apache.org/general/downloads.html>
- Jira
 - <https://issues.apache.org/jira/browse/Sentry>
- How to contribute
 - <https://cwiki.apache.org/confluence/display/SENTRY/How+to+Contribute>
- Mailing list
 - dev@sentry.incubator.apache.org

Demo Time

Thank You!

Contributions are welcome !