Apache Sentry - High Availability

Hao Hao - hao.hao@cloudera.com

Seville, Spain, Nov 14 - 16 2016

cloudera

Sentry

About me

- Software engineers at Cloudera
- Apache Sentry PMC and Committer

Presentation Agenda

- Apache Sentry Overview
  - Introduction
  - Architecture
- Apache Sentry High Availability
  - Motivation: deploying flexible and robust security
  - Challenges to ensure fault tolerance, high availability
  - High level design
- Other Features and Future Work

Sentry Overview

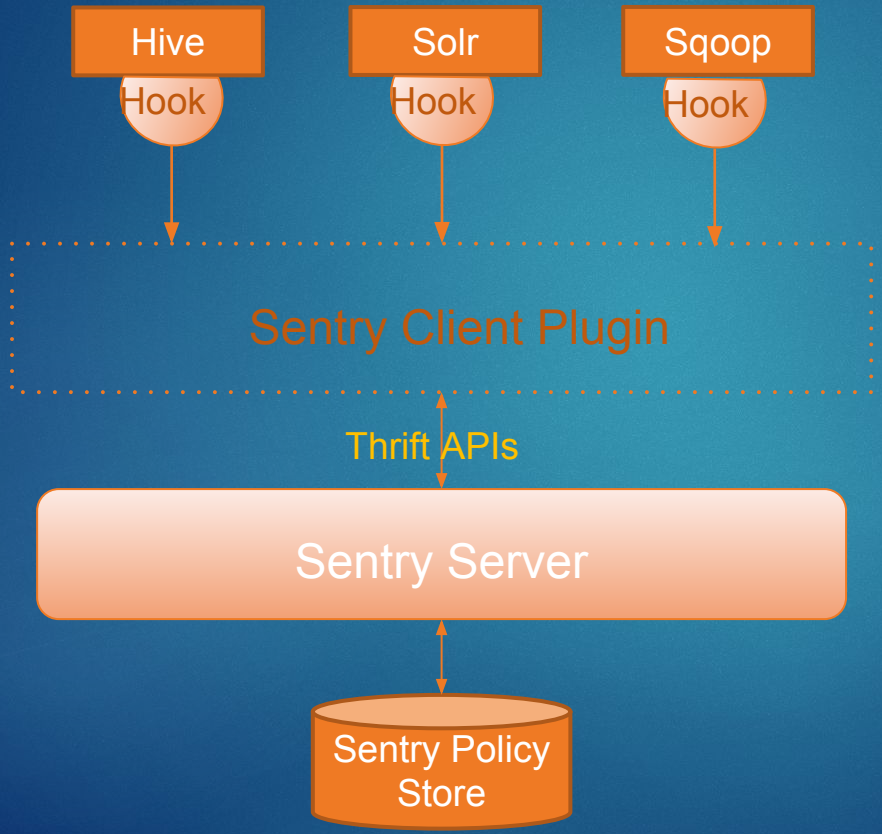
- **Authorization** Service
  - Sentry provides the ability to enforce role-based access control (RBAC) to data and/or metadata for authenticated users in a fine-grained manner.
  - Enterprise grade big data security.
  - Provides unified policy management.
  - Pluggable and highly modular.

- Works out of the box with Apache Hive, Hive metastore/HCataglog, Apache Solr, Apache Kafka, Apache Sqoop and Apache Impala.

cloudera

Sentry

Sentry Architecture

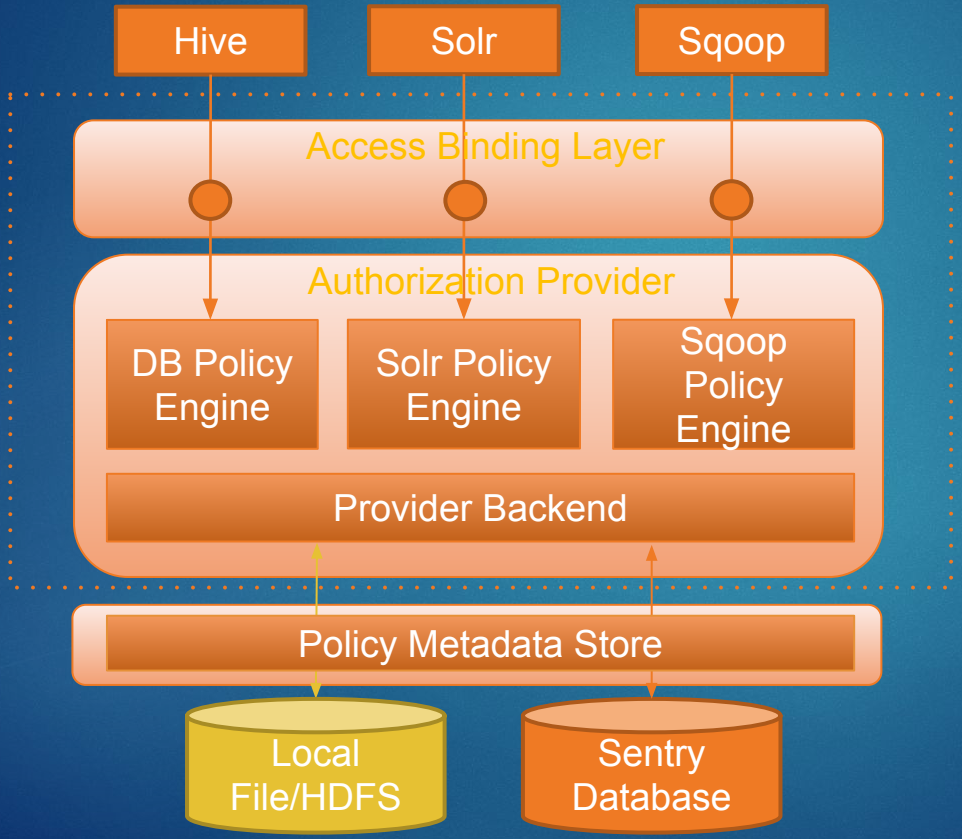


- Server Client Model
- Thrift client APIs
  - Get privileges
  - Grant/Revoke role
  - Grant/Revoke privileges
  - List roles

cloudera

Sentry

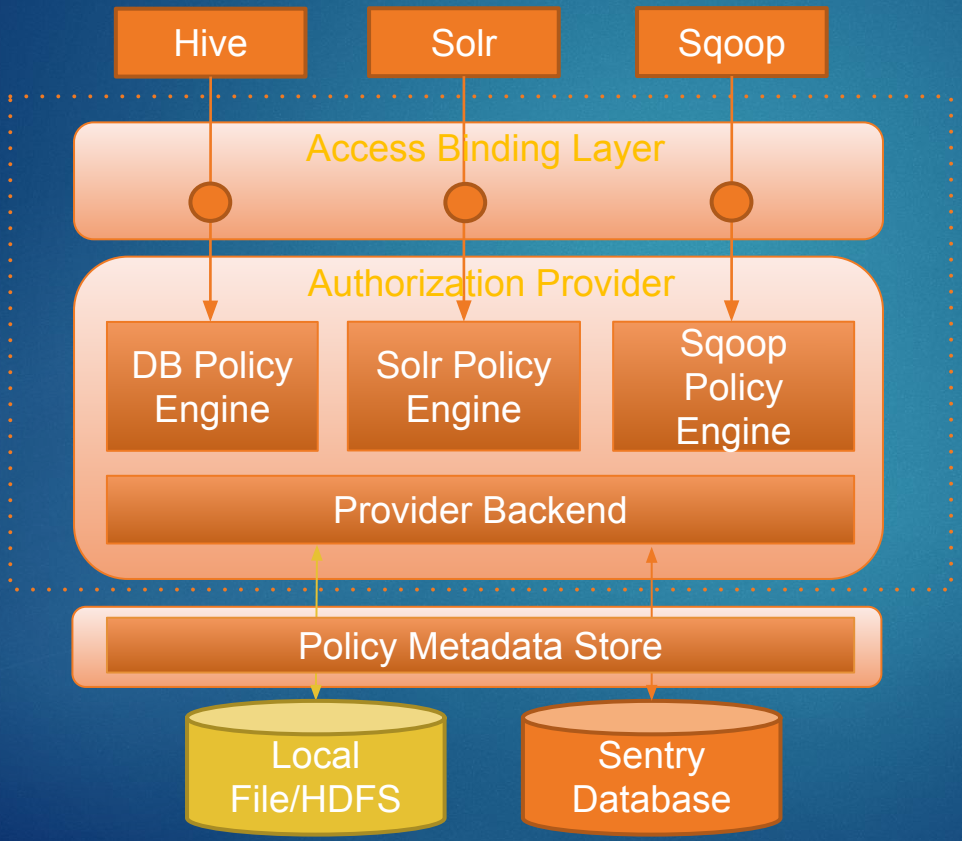
Sentry Architecture



cloudera

Sentry

Sentry Architecture

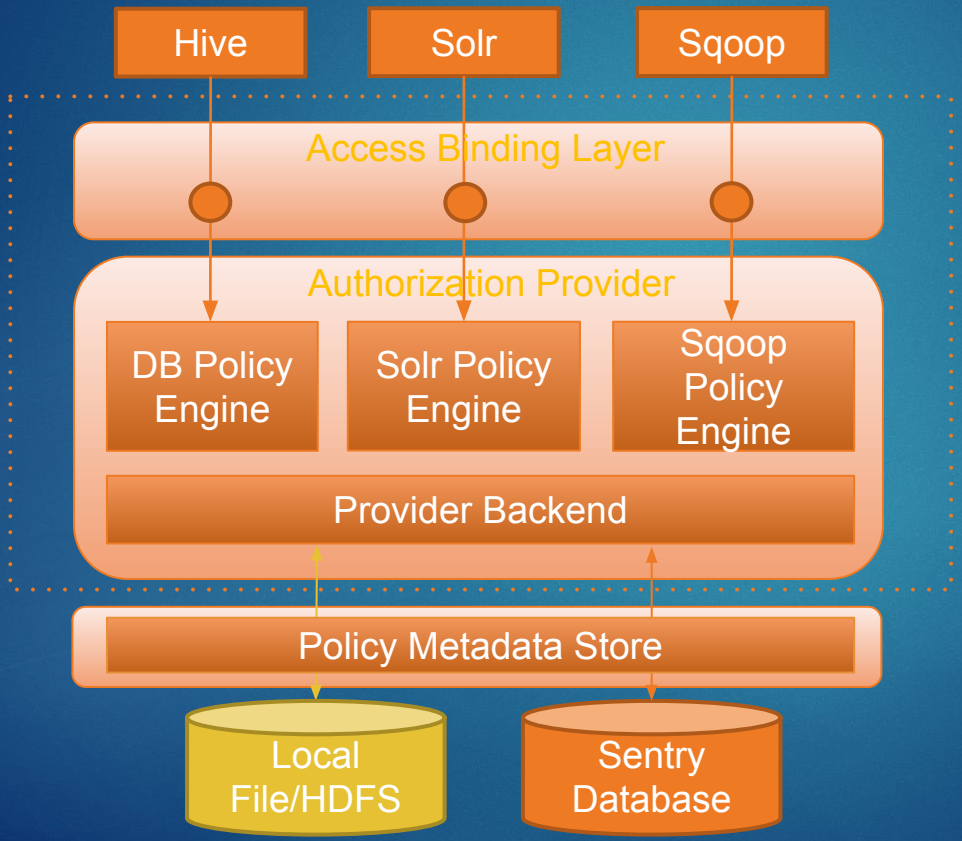


- **Binding Layer**: takes the authorization requests in the native format of requestors and converts that into an authz request based on the **authorization data model** that can be handled by Sentry authorization provider.

cloudera

Sentry

Sentry Architecture

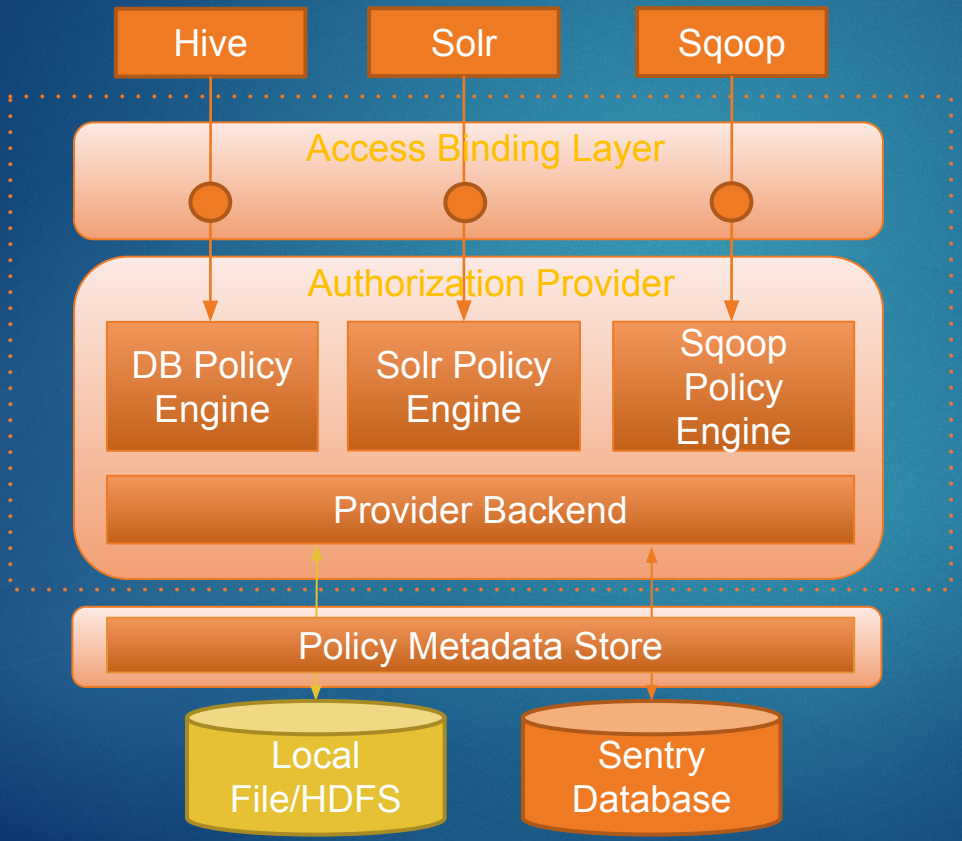


- **Authorization provider**: an abstraction for making the authorization decision for the authz request from binding layer. Currently, supplies a RBAC authorization model implementation.

cloudera

Sentry

Sentry Architecture

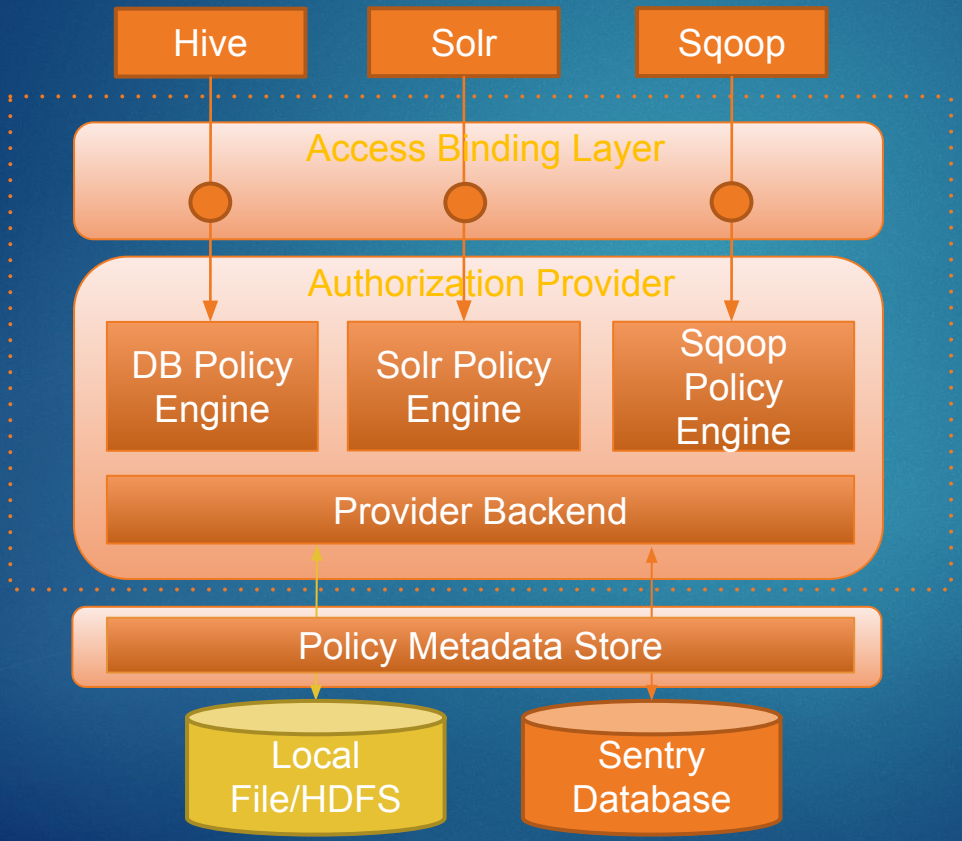


- **Policy Engine**: gets the requested privileges from the binding layer and the required privileges from the provider layer. It looks at the requested and required privileges and makes the decision whether the action should be allowed.

cloudera

Sentry

Sentry Architecture

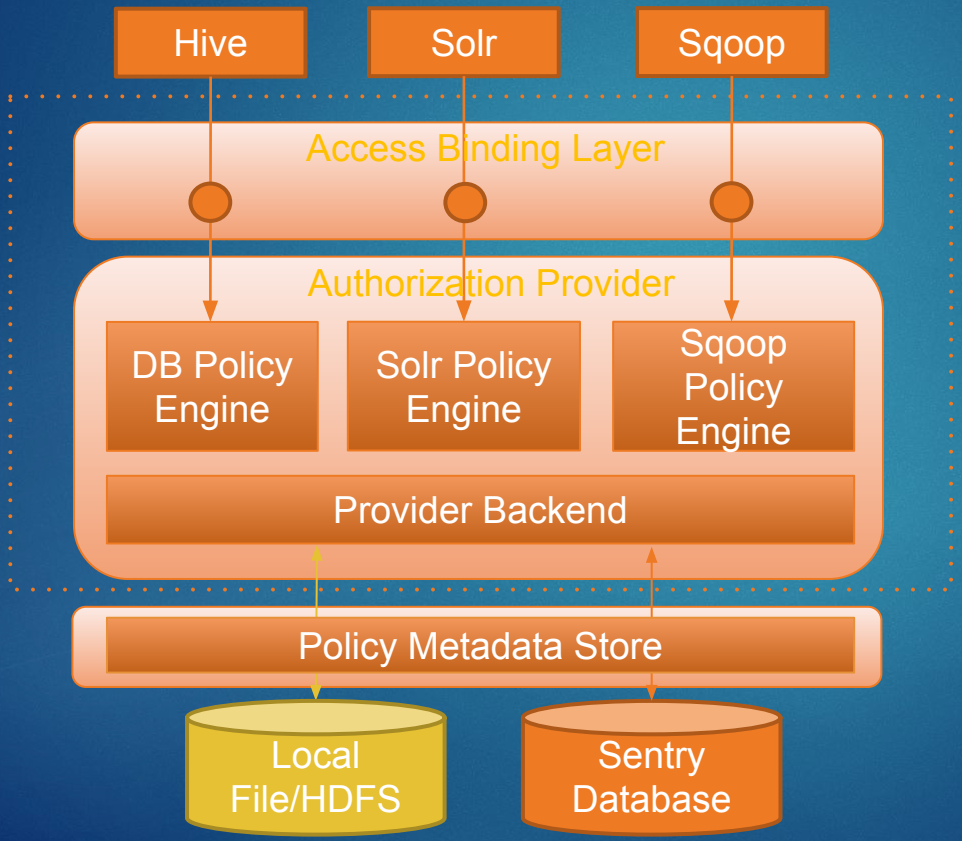


- **Policy Backend**: making the authorization metadata available for the policy engine. It allows the metadata to be pulled out of the underlying repository independent of the way that metadata is stored.

cloudera

Sentry

Sentry Architecture



- **Sentry policy store and Sentry Service**: persist the role to privilege and group to role mappings in an RDBMS and provide programmatic APIs to create, query, update and delete it. This enables various Sentry clients to retrieve and modify the privileges concurrently and securely.

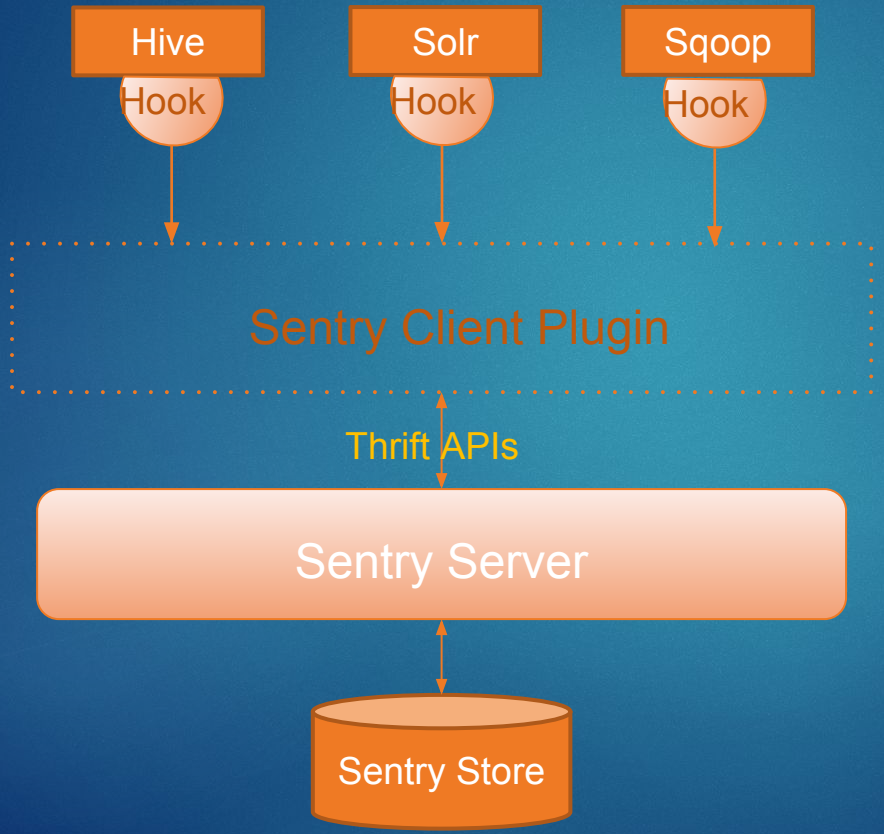
cloudera

Sentry

HDFS Sync Feature

- Single source of truth
  - User “Alice” has INSERT privileges on db1.tb1
  - ACLs for “Alice” => WRITE_EXECUTE on /user/warehouse/db1.db/tb1
  - Multiple compute frameworks can rely on same policies

Sentry Architecture



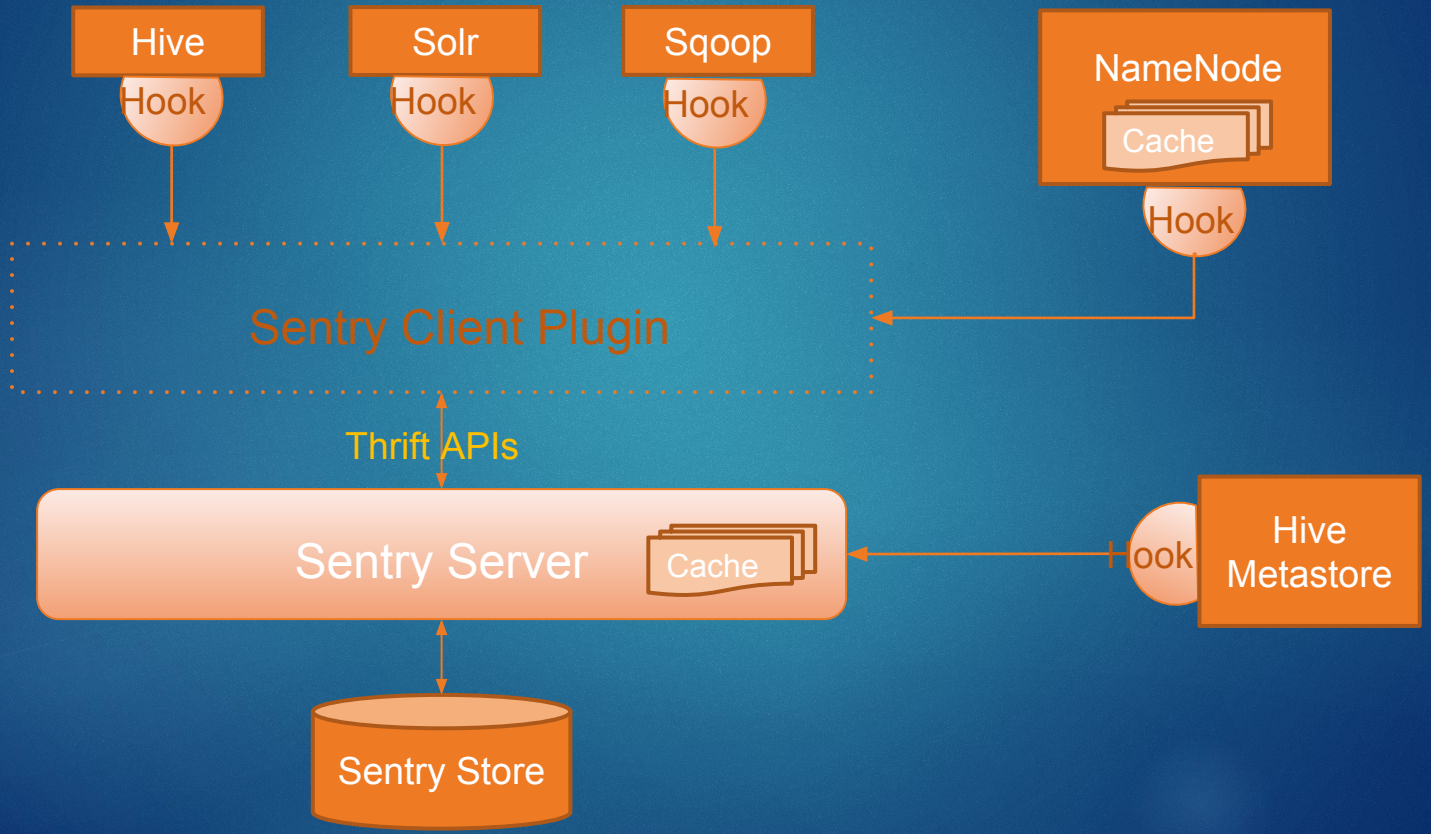
cloudera

Sentry

Sentry Server State

- Sentry sever is stateless
- Sentry permissions are kept in Sentry policy DB:
  - Roles to privilege mappings
  - Groups to roles mappings

Sentry Architecture with HDFS Sync



cloudera

Sentry

Sentry Server State with HDFS Sync

- Sentry server become stateful
- In-memory cache:
  - Replication/snapshot of Apache Hive metastore (HMS) data:
    - <Hive Obj, HDFS path>
  - HMS delta change:
    - (Id, Object, requests to add or remove path)

High Availability Requirement

- Deploy Apache Sentry without introducing any **single points of failure**
- Apache Sentry needs to work with **Apache Hive metastore high availability** solution
- Be able to use enable Apache Sentry in conjunction with **Apache HDFS high availability** as well

High Availability Assumption

- Backend database supports HA and is configured to use it.
- Apache Sentry clients are permitted to cache permission data for some reasonable amount of time and use outdated permissions during this time.

High Availability Goal

- Availability
  - Sentry service must be available when one of the sentry servers stops working. It should also be possible to restart a Sentry daemon while the cluster is active.

High Availability Goal

- Consistency: support a **reasonable timeline consistency** model.
  - Any write operation should bring the database from one valid state to another. For example, a request to create a table followed by a request to change table permissions, followed by table scan, all within a single session, should work as expected.
  - Read requests should return a consistent view of the permissions. For clients accessing the Server across multiple sessions, the goal is to provide eventual consistency.

High Availability Challenges

- Ad-hoc synchronization between Apache Hive Metastore and Apache Sentry
  - pitfalls related to different daemons receiving events in different orders.
- Sentry server is stateful
  - pitfalls related to synchronizing in memory state between multiple active daemons

High Availability Design

- Make use of **notification log** API introduced by Hive HA: for clients to receive changes to the Hive metastore.
  - Hive notification log uses HMS database as the source of information and the notification log provides a global stream of modifications made by all Hive MetaStore (HMS) clients.
  - Notifications log are identified by a unique epoch number. All notifications are stored in a SQL table.

High Availability Design

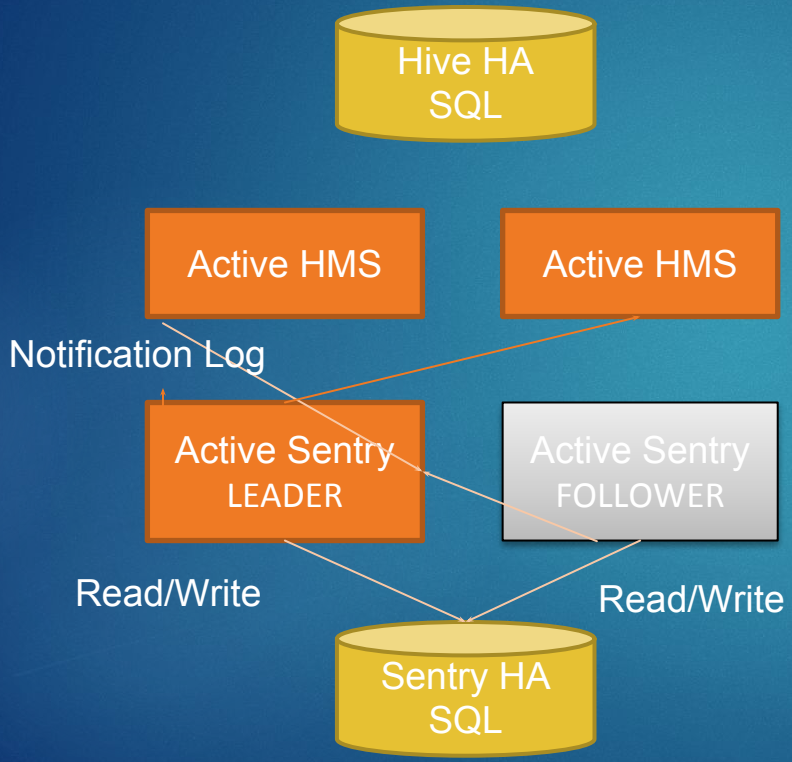
- Converting Sentry to a stateless service where all the actual **state is stored in the backend HA database**.
  - The removal of in-memory state allows for **active/active** High Availability configuration with preserved consistency.
  - Store all state in sentry store:
    - Snapshot of Sentry permissions
    - Snapshot of HMS data
    - Update log for HMS delta change
    - Update log for Sentry permission delta change

High Availability Design

cloudera

Sentry

High Availability Design

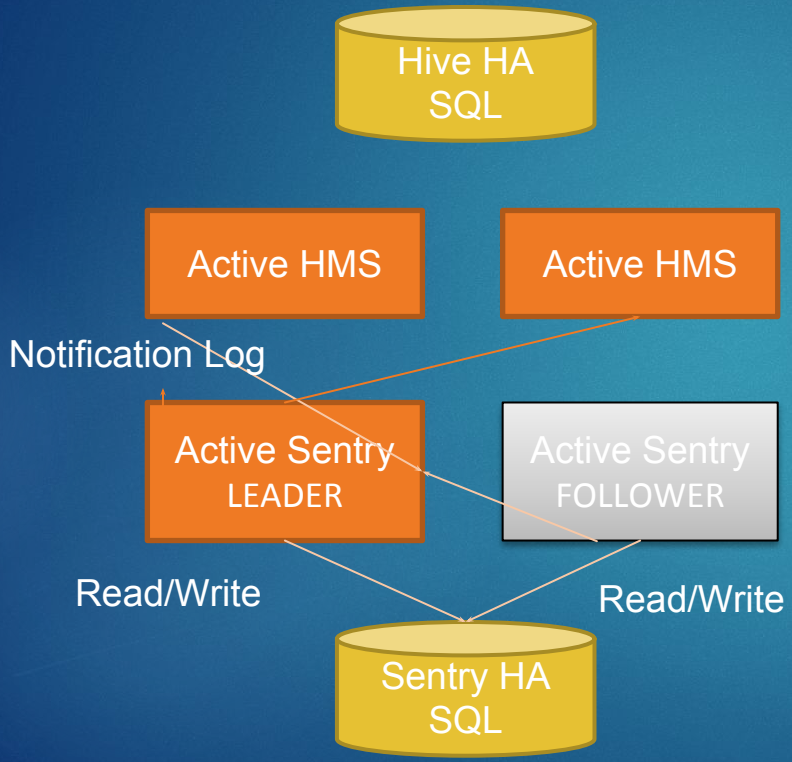


- **Synchronizing with HMS update s**. Dedicate a single node to the processing of HMS notification log. It also provide a way to associate Sentry permission state with specific HMS notification ID change by referring Hive object name.

cloudera

Sentry

High Availability Design



- **Leader election** is performed using ZooKeeper.
- The first daemon to claim the node becomes the leader. This is currently implemented using Curator framework.

cloudera

Sentry

High Availability Design

- Interaction with Hive Metastore.
  - Initially during the transition to the new model, need to synchronize HMS point-in-time snapshot with the Sentry state.

  - Assumption: Need stop any write activity on the cluster
    - Read current HMS notification ID_initial
    - Read HMS metadata state
    - Read current notification ID_new
    - If ID_initial != ID_new then discard the current state and retry.

High Availability Design

- Notification logs are processed asynchronously
- Write barrier
  - an RPC call for use by HMS Sentry plugin. The call wait_until(ID_n) should block until Sentry processes ID_n. This call should be used whenever HMS updates state.

Sentry Releases

- Successfully graduated from the Incubator in March of 2016 and now is a Top-Level Apache project
- Sentry 1.5.1 released.

Sentry Releases

- Sentry 1.6.0 released on Feb 24, 2016
  - Add capability to export/import to dump or load Sentry metadata
  - Integrate Sqoop with Sentry by using generic authorization model

- About to release 1.7.0
  - Integrate Hive v2 into Sentry
  - Integrate Kafka with Sentry by using generic authorization model
  - Integrate Solr with Sentry by using generic authorization model

Other Features

- Sentry generic authorization model
  - Motivation: easy integration with Apache data engines, even third-party data applications
  - Successfully integrated with Apache Solr, Kafka and Sqoop2

Other Features

- Sentry column level privilege: Fine-grained authorization for SQL engine
  - Motivation: No need to create views for the purpose of column level authorization
  - Support Hive/Impala

How to contribute?

- Jump in on discussions
- File Bugs
- Review Code
- Provide Patches
- Reference: https://cwiki.apache.org/confluence/display/SENTRY/How+to+Contribute

cloudera

Sentry

Reference

- Apache Sentry: https://cwiki.apache.org/confluence/display/SENTRY/Home
- Apache Sentry High Availability Design: https://issues.apache.org/jira/secure/attachment/12835271/SENTRY-872_design-v2.1.1.pdf

cloudera

Sentry

Questions?