

# Apache Atlas: Governance for your data

**Madhan Neethiraj**  
**Director - Engineering, Atlas PMC**

**Suma Shivaprasad**  
**Staff Engineer, Atlas PMC**



# Disclaimer

- ◆ *This document may contain product features and technology directions that are under development, may be under development in the future or may ultimately not be developed.*
- ◆ *Project capabilities are based on information that is publicly available within the Apache Software Foundation project websites ("Apache"). Progress of the project capabilities can be tracked from inception to release through Apache, however, technical feasibility, market demand, user feedback and the overarching Apache Software Foundation community development process can all effect timing and final delivery.*
- ◆ *This document's description of these features and technology directions does not represent a contractual commitment, promise or obligation from Hortonworks to deliver these features in any generally available product.*
- ◆ *Product features and technology directions are subject to change, and must not be included in contracts, purchase orders, or sales agreements of any kind.*
- ◆ *Since this document contains an outline of general product development plans, customers should not rely upon it when making purchasing decisions.*

# Agenda

- Introduction
- Apache Atlas
  - Overview
  - Data Provenance/Lineage
  - Classification
  - Metadata Catalog Search
  - Architecture
- Demo
- Roadmap
- Q & A

# Apache Atlas: Introduction

*Provides metadata-driven core foundational governance services for Hadoop and enterprise data ecosystem*

## Data Lineage/Provenance

- Captures data lineage across components

## Data Classification

- Supports classification of data assets using tags – PII, PHI, PCI, EXPIRES\_ON, CLAIMS, LIFE\_INSURANCE

## Metadata Catalog Search

- Free text search on metadata
- Advanced search using DSL

## Integrations

- OOB real-time metadata and lineage ingestion with Hive, Sqoop, Storm/Kafka
- APIs for custom metadata ingestion
- Apache Ranger integration for classification based security

## Metadata Repository

- Flexible metamodel to capture technical, business, operational metadata
- Out-of-box models for Hive, Storm, Sqoop, HDFS, Kafka, HBase
- APIs to register custom models

# Background: DGI Community becomes Apache Atlas



Global Financial  
Company



Dec  
2014  
*DGI group  
Kickoff*

May  
2015  
*Apache  
Atlas  
Incubation*

Aug  
2016  
*HDP 2.5/  
Apache 0.7  
Foundation  
Release*

Apr  
2017  
*HDP 2.6/  
Apache 0.8  
Release*



Community Connection

294  
Questions

536  
Answers

\* DGI: Data Governance Initiative

- #Committees – 35
- Code contributors from
  - Hortonworks, IBM, Aetna, Merck, Target

## Apache 0.8/HDP 2.6

- Simplified Search UI
- Simplified APIs
- Classification-based security for HDFS, Kafka, HBase
- Knox SSO
- Performance/scalability improvements

## Apache 0.7.1/HDP 2.5.3

- High availability support
- LDAP Authentication/Authorization
- Classification based security for Hive
- UI Redesign



# Apache Atlas : Lineage

## Lineage

- Where does this data originate from (source/provenance)?
- **Upstream path:** Path through all data assets and processes leading up to current data asset

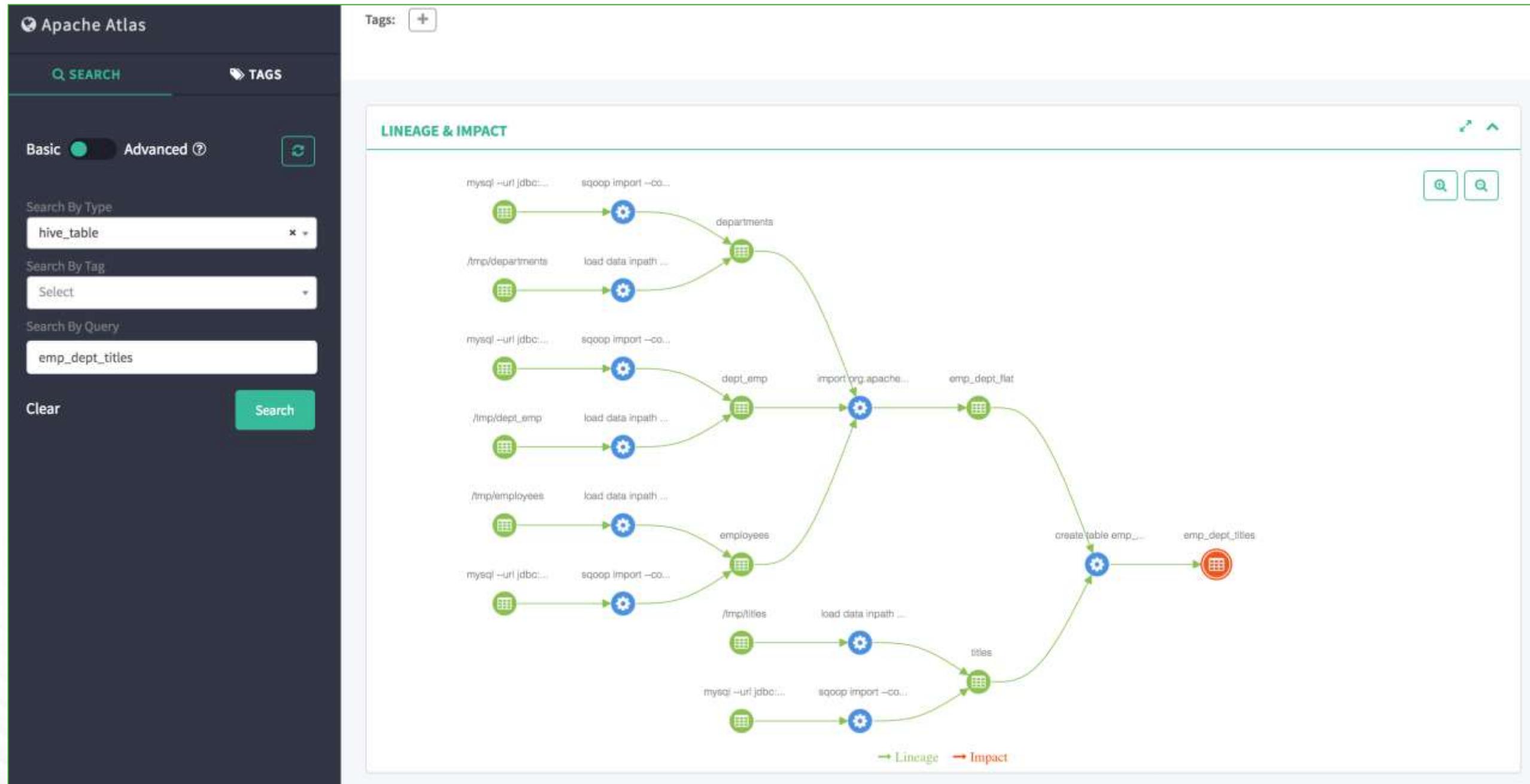
## Impact

- How is this data being used ?
- What other data assets (derivative/dependent) does this impact?
- **Downstream path:** Path through all data assets and processes leading out of current data asset

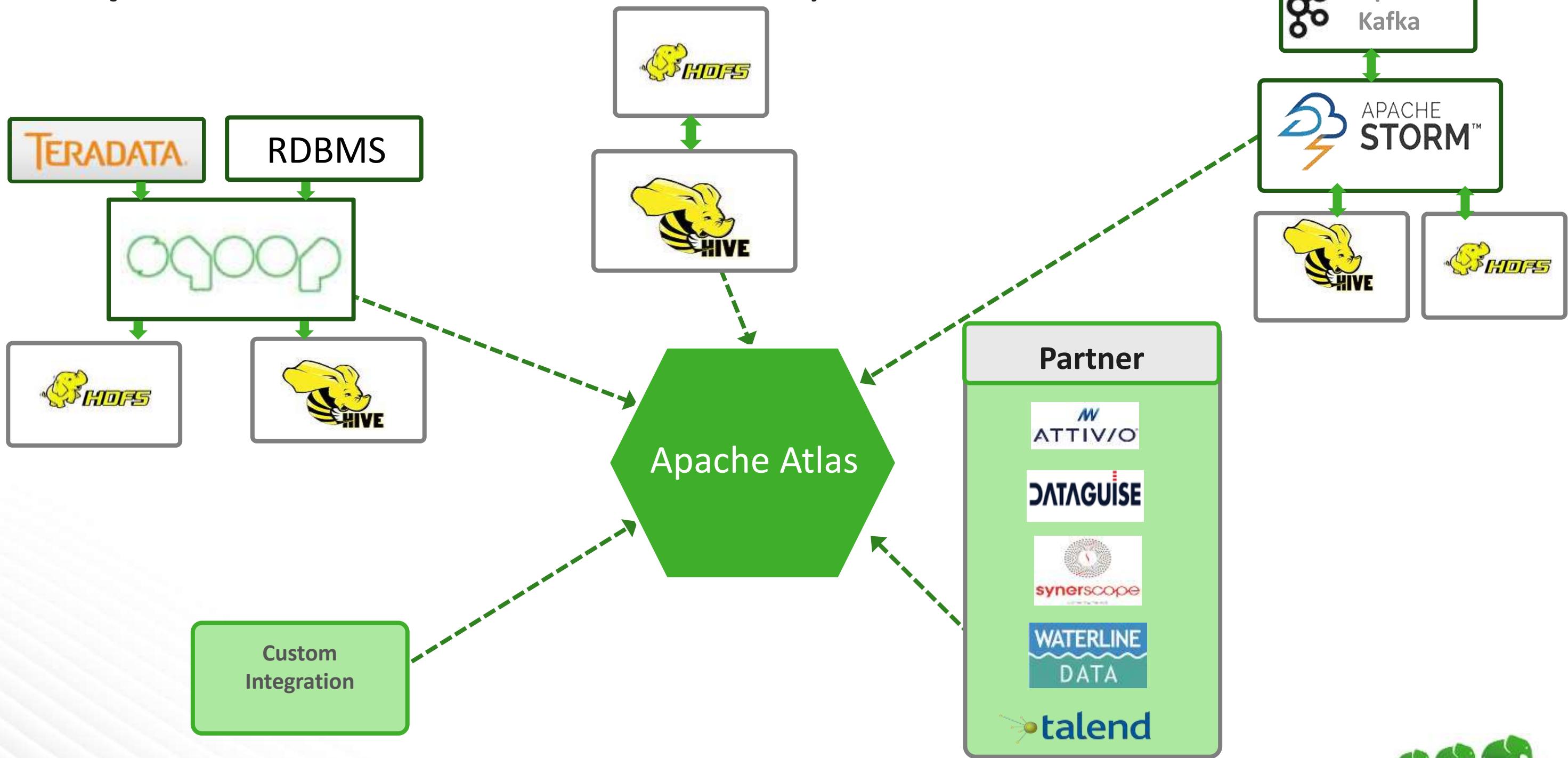
## Used for forensics

- Impact analysis
- Auditing and Compliance

# Apache Atlas: Lineage and Impact



# Apache Atlas: Connectors and Ecosystem



# Apache Atlas: Classification

- Categorize and curate data assets for easier discovery
- Associate context with data assets – Governance, Security, Business, ...

The screenshot shows the Apache Atlas web interface. On the left, a sidebar lists various tags: AUTO\_INSURANCE, BANKING, CONSUMER\_BANKING, DATA\_QUALITY, EXPIRES\_ON (highlighted in green), FINANCE\_PII, HEALTH\_INSURANCE, INSURANCE, LIFE\_INSURANCE (highlighted in green), PII, REFERENCE\_DATA, SMALL\_BUSINESS\_BANKING, TradingDataset, VENDOR\_PII, and WEALTH\_MANAGEMENT. The main panel shows results for the tag 'EXPIRES\_ON'. It displays three tables named 'tax\_2015', 'tax\_2010', and 'tax\_2009', all of type 'hive\_table' owned by 'hive'. Each table has a 'Tags' column containing 'EXPIRES\_ON'. Three green arrows point from the sidebar tags 'EXPIRES\_ON' and 'LIFE\_INSURANCE' to the main panel, each pointing to a specific table. A green callout bubble labeled 'GOVERNANCE' points to the 'EXPIRES\_ON' tag in the sidebar. Another green callout bubble labeled 'SECURITY' points to the 'EXPIRES\_ON' tag in the main panel. A third green callout bubble labeled 'BUSINESS' points to the 'LIFE\_INSURANCE' tag in the sidebar.

Name	Description	Type	Owner	Tags
tax_2015		hive_table	hive	EXPIRES_ON
tax_2010		hive_table	hive	EXPIRES_ON
tax_2009		hive_table	hive	EXPIRES_ON

# Apache Atlas Classification : usecase – REFERENCE\_DATA

## Security policy enforcement for denying updates on immutable data assets

- **REFERENCE\_DATA** classification associated with immutable hive\_table **eu\_countries**
- Apache Ranger policies block updates on the table for all users except admins

**REFERENCE\_DATA**

REFERENCE\_DATA

Attributes: ADD Attribute +

Results for **REFERENCE\_DATA**

Showing 1 - 1

Previous Next

Name	Description	Type	Owner	Tags
eu_countries	hive_table	hive		REFERENCE_DATA X +

**Policy Details :**

Policy Type: Access

Policy ID: 40

Policy Name \*: access: REFERENCE\_DATA

enabled

TAG \*: REFERENCE\_DATA

Audit Logging: YES

Description: Access policy for data classified as REFERENCE\_DATA

**Components Permissions**

Component	Permissions
hdfs	<input type="checkbox"/> Read <input checked="" type="checkbox"/> Write <input type="checkbox"/> Execute
hive	<input type="checkbox"/> select <input checked="" type="checkbox"/> update <input type="checkbox"/> Create <input checked="" type="checkbox"/> Drop <input checked="" type="checkbox"/> Alter <input type="checkbox"/> Index <input type="checkbox"/> Lock <input type="checkbox"/> All

**Allow Conditions :**

**Deny Conditions :**

Select Group	Select User	Policy Conditions	Component Permissions
public	Select User	Add Conditions	HDFS HIVE



# Apache Atlas Classification : usecase – access expiry

## Data expiration

- **EXPIRES\_ON** classification with attribute expiry\_date
- tax\_2009 table tagged with EXPIRES\_ON(expiry\_date=2016/12/31)
- tax\_2010 table tagged with EXPIRES\_ON(expiry\_date=2017/12/31)
- Apache Ranger policies use the attribute to block access after expiry date

**tax\_2009 (hive\_table)**

Tags: **EXPIRES\_ON**

**LINEAGE & IMPACT**

**DETAILS**

Properties Tags Audits Schema

Showing 1 - 1

Tags	Attributes
EXPIRES_ON	expiry_date:2016/12/31

**tax\_2010 (hive\_table)**

Tags: **EXPIRES\_ON**

**LINEAGE & IMPACT**

**DETAILS**

Properties Tags Audits Schema

Showing 1 - 1

Tags	Attributes
EXPIRES_ON	expiry_date:2017/12/31

**Policy Details :**

Policy Type **Access**

Policy ID **4**

Policy Name \* **access: EXPIRES\_ON**  **enabled**

TAG \* **# EXPIRES\_ON**

Audit Logging **YES**

Description **Policy for data with EXPIRES\_ON tag**

**Allow Conditions :**

**Deny Conditions :**

Select Group **# public** Select User **Select User** Policy Conditions **accessed-after-expiry : yes**

# Apache Atlas Classification: usecase – attribute based authorization

## Data quality

- Deny access to **analysts** group based on data quality threshold

The screenshot shows the Apache Atlas interface. On the left, there's a sidebar with search and tag filters. The main area displays a table named "claim\_savings (hive\_table)". Below the table, under "LINEAGE & IMPACT", is a "DETAILS" section. In the "Tags" tab of the details section, there is a single tag entry: "DATA\_QUALITY".

The screenshot shows the "Policy Details" page for creating a new access policy. The policy type is set to "Access". The policy ID is "5". The policy name is "access: DATA\_QUALITY" and it is marked as "enabled". The tag associated with this policy is "\* DATA\_QUALITY". The audit logging setting is "YES". The description states: "Prevent analyst from accessing data with low data-quality score". Under "Allow Conditions", there is a "Select Group" field containing "analyst". Under "Policy Conditions", there is a field with the expression "expression : tagAttr.score < 0.5".

# Apache Atlas Classification: usecase – cross component

## Classification based security on cross-component data assets

The screenshot displays two main components of the Hortonworks Data Platform interface:

**Left Panel (Classification View):**

- Search Bar:** Includes a search input field and a "Create Tag" button.
- Tags Sidebar:** A list of available tags including: AUTO\_INSURANCE, BANKING, CONSUMER\_BANKING, DATA\_QUALITY, EXPIRES\_ON, FINANCE\_PII, HEALTH\_INSURANCE, INSURANCE, LIFE\_INSURANCE, PII, REFERENCE\_DATA, SMALL\_BUSINESS\_BANKING, TradingDataset (highlighted in green), VENDOR\_PII, and WEALTH\_MANAGEMENT.
- TradingDataset View:** A list of datasets under the "TradingDataset" tag. The results are as follows:

Name	Description	Type	Owner	Tags
/feeds/mutualfunds	Mutual Funds	hdfs_path	brokers	TradingDataset
futures_trade		hive_table	hive	TradingDataset
/feeds/forex	Forex	hdfs_path	brokers	TradingDataset
derivatives_trade		hive_table	hive	TradingDataset
/feeds/commodities	Commodity Data	hdfs_path	brokers	TradingDataset

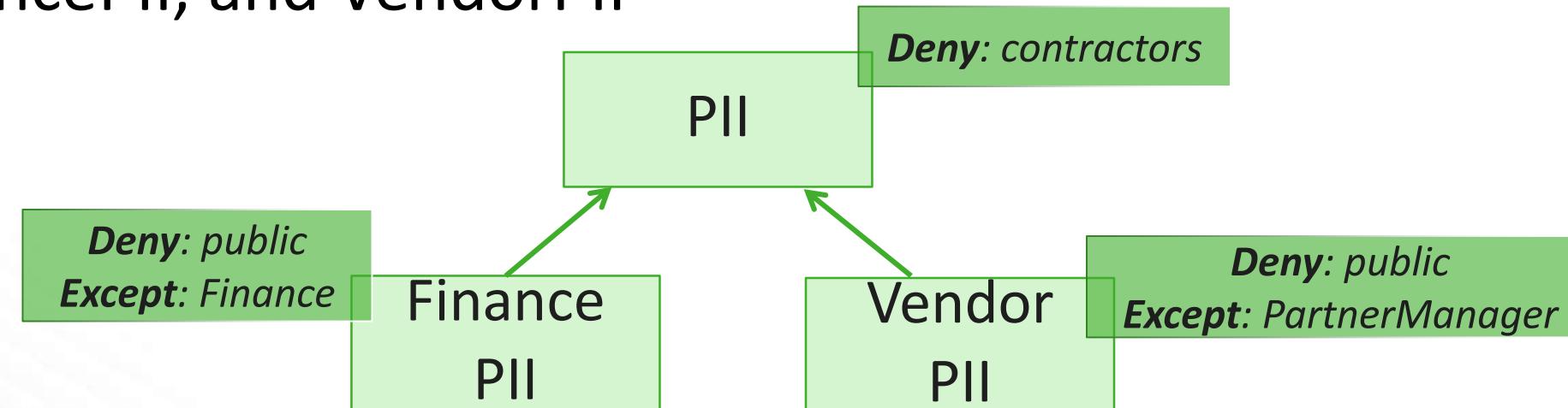
**Right Panel (Policy Details):**

- Policy Details:** Shows a policy named "access: TradingDataset" with ID 41, enabled status, and a TAG of "TradingDataset". Audit Logging is set to YES. Description: "Access policy for data classified as TradingDataset".
- Allow Conditions:** A table for selecting groups and users with a condition for "brokers".
- Exclude from Allow Conditions:** An empty section.
- Deny Conditions:** A table for selecting groups and users with a condition for "public".

# Apache Atlas Classification: usecase - hierarchy

## Security policy enforcement based on classification hierarchy

- Data assets classified as **PII** will be denied for all contractors
- Data assets classified as **FinancePII** will be denied for anyone not in Finance group
- Data assets classified as **VendorPII** will be denied for anyone not in PartnerManager group
- ..hence contractors will be denied access to data assets classified as PII, FinancePII, and VendorPII



# Metadata Catalog Search : Free Text

The screenshot shows the Apache Atlas search interface. On the left, there's a sidebar with 'SEARCH' and 'TAGS' buttons, and tabs for 'Basic' and 'Advanced'. Below these are sections for 'Search By Type', 'Search By Tag', and 'Search By Query'. The 'Search By Type' section contains a dropdown with 'hive\_table' selected. The 'Search By Tag' section contains a dropdown with 'PII' selected. The 'Search By Query' section contains a text input with 'emp\*' and a 'Search' button. In the center, the search results are displayed under the heading 'Results for **hive\_table & PII & emp\***'. It says, 'If you do not find the entity in search result below then you can [create new entity](#)'. A table shows one result: Name: employees, Description: null, Type: hive\_table, Owner: hive, Tags: PII. At the bottom, it says 'Showing 1-1'. Three callout boxes provide additional information: one for 'Filter by Data Asset type', one for 'Filter by Classification', and one for the search text input.

Results for **hive\_table & PII & emp\***  
If you do not find the entity in search result below then you can [create new entity](#)

Showing 1-1

Name	Description	Type	Owner	Tags
employees		hive_table	hive	PII

*Filter by Data Asset type*

*Filter by Classification*

*Search text*  
Wildcards: *emp\**, *\*dept\**  
Logical expressions: *emp\* AND \*dept\**

*Search for a **hive\_table** classified as 'PII' and name starting with 'emp'*

# Metadata Catalog Search : Advanced

The screenshot shows the Apache Atlas Metadata Catalog Search interface. On the left, there's a sidebar with tabs for 'Basic' and 'Advanced'. The 'Advanced' tab is selected, indicated by a green dot. Below it are sections for 'Search By Type' (containing 'hive\_table') and 'Search By Query' (containing 'where name='employees' and owner='hive''). A 'Search' button is at the bottom of this sidebar. On the right, the main content area has a header 'DSL search with SQL like syntax' with a sub-instruction 'Select columns from impressions table in raw database'. It displays a query: `hive_column where table.name='impressions' and table.db.name = 'raw'`. Below this, it shows results for `hive_table where name='employees' and owner='hive'`, with a note about creating new entities. A green callout box labeled 'Filter by Data asset type' points to the 'Data asset type' column in the results table. The results table has columns: Name, Description, Type, Owner, and Tags. One row is shown: Name is 'employees', Description is empty, Type is 'hive\_table', Owner is 'hive', and Tags include 'PII' with a delete icon and a plus icon. A green callout box labeled 'DSL query string' points to the search query in the sidebar. Another green callout box labeled 'Showing 1-1' points to the table header.

Apache Atlas

SEARCH TAGS

Basic Advanced ?

Search By Type

hive\_table

Search By Query

where name='employees' and owner='hive'

Clear

Search

DSL search with SQL like syntax

Select columns from *impressions* table in *raw* database

`hive_column where table.name='impressions' and table.db.name = 'raw'`

Results for `hive_table where name='employees' and owner='hive'`

If you do not find the entity in search result below then you can [create new entity](#)

Showing 1-1

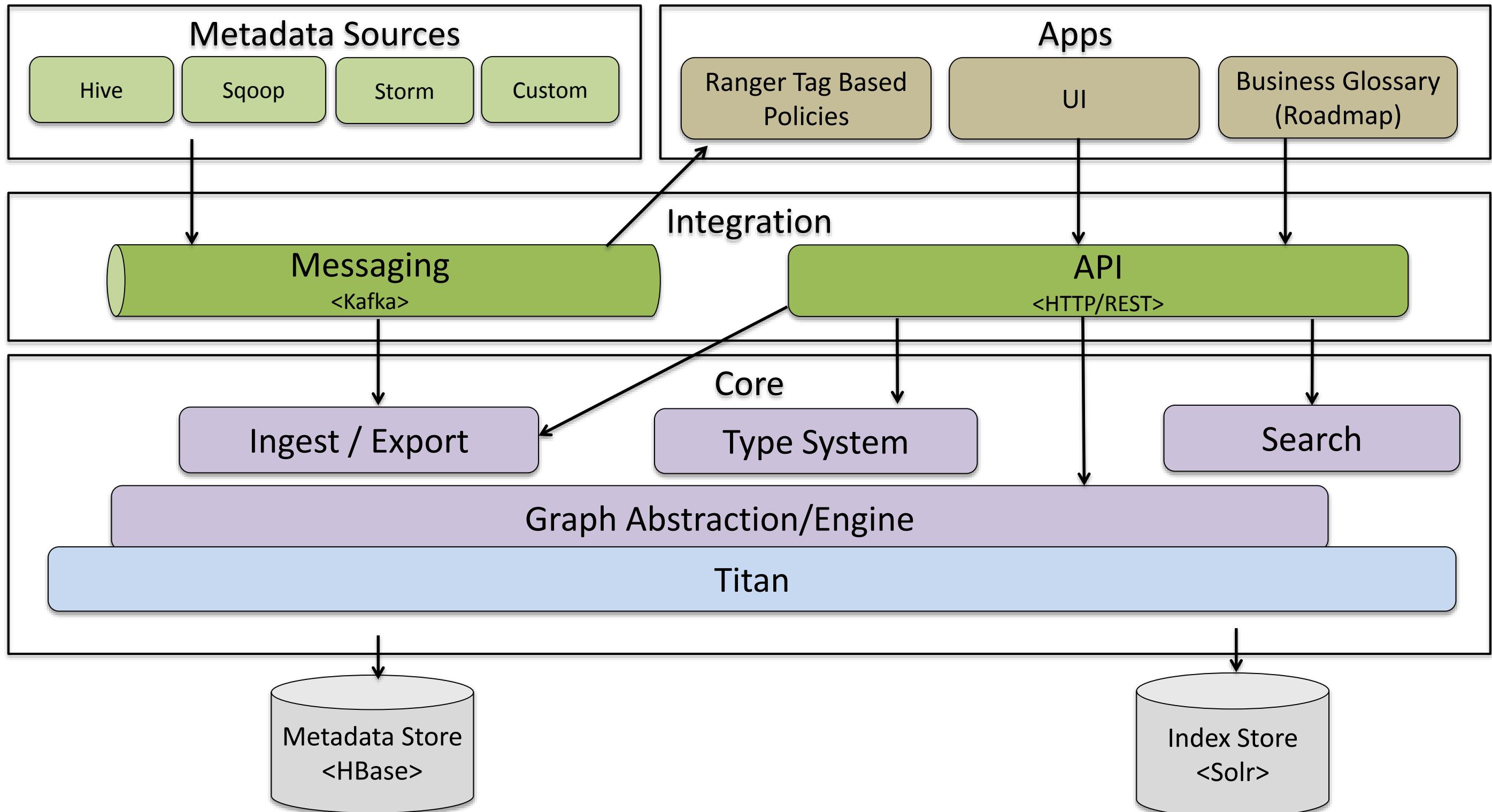
Filter by Data asset type

	Name	Description	Type	Owner	Tags
	employees		hive_table	hive	PII

DSL query string

Search for a `hive_table` named '*employees*' and owner '*hive*'

# Apache Atlas: Architecture



# Apache Atlas : Roadmap

## Business Taxonomy/Glossary

- Vocabulary for business users
- Categorized and curated by data stewards
- Association with physical assets like database, tables for effective data classification
- Discover data assets more naturally with glossary terms

## Connectors

- Spark, NiFi, HBase

## Hive Column-Level Lineage

## Search/Filtering on Lineage

## Export/Import of Atlas lineage and metadata

# Questions

# References

- Apache Atlas
  - <http://atlas.apache.org>
  - <http://hortonworks.com/apache/atlas>
- Apache Atlas community
  - <https://community.hortonworks.com/spaces/64/governance-lifecycle-track.html?topics=Atlas&type=question>
- Apache Ranger
  - <http://ranger.apache.org>
  - <http://hortonworks.com/apache/ranger>
  - <https://cwiki.apache.org/confluence/display/RANGER/Tag+Based+Policies>