Starburst

Federated Semantic Layer

Starburst: www.starburstdata.com

Federated Semantic Layer using Presto

Agenda

1. Presto/Starburst Introduction
2. Where we are at and how did we get here?
3. What are the benefits of semantic layers?
4. Creating an **open**, federated semantic layer using Presto
5. Demo
6. Questions

What is Presto?

Community-driven
open source project

High performance ANSI SQL engine

- New Cost-Based Query Optimizer
- Proven scalability
- High concurrency

Separation of compute and storage

- Scale storage and compute independently
- No ETL or data integration necessary to get to insights
- SQL-on-anything

No vendor lock-in

- No Hadoop distro vendor lock-in
- No storage engine vendor lock-in
- No cloud vendor lock-in

Vision

- Enable seamless access to data anywhere at any time allowing companies to realize the full potential of their most important asset: data.

Who are we?

- Founding team of the largest committers to the open source project Presto since 2015
- Former Teradata, Databricks, Vertica, Hadapt, Netezza, and Ab Initio

Starburst “Stack”

Platforms

On-Premises

Azure

GCP

kubernetes

aws

Enterprise Support

- 24/7 Support
- Long Term Support
- Product Influence
- Hot fixes/patches

Security

- Kerberos
- LDAP
- RBAC
- Encryption
- Query Auditing

Connectivity

- Teradata
- Snowflake
- Oracle
- Cassandra

- ODBC
- JDBC

Cluster Provisioning **Starburst Mission Control** Connector Mangement

Starburst ENTERPRISE PRESTO

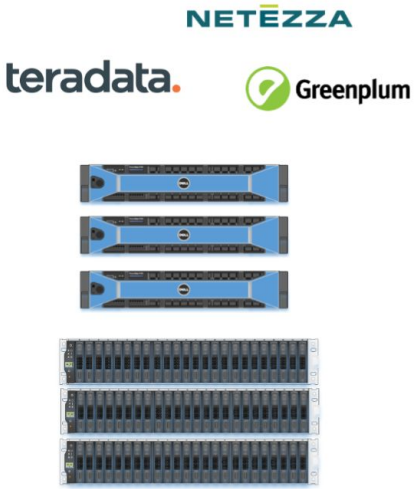
Enterprise Presto Distribution

- Autoscaling/Graceful Shutdown
- High Availability

Open Source Presto

Data Warehouse 1.0

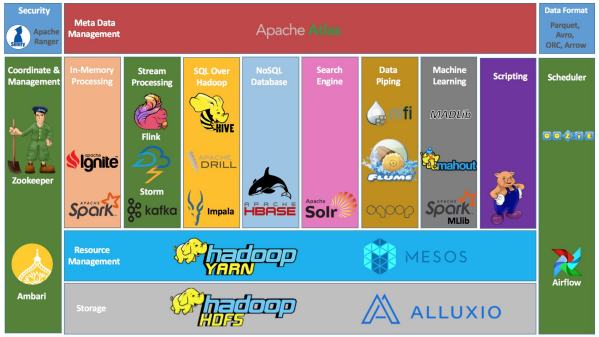
Shift towards MPP



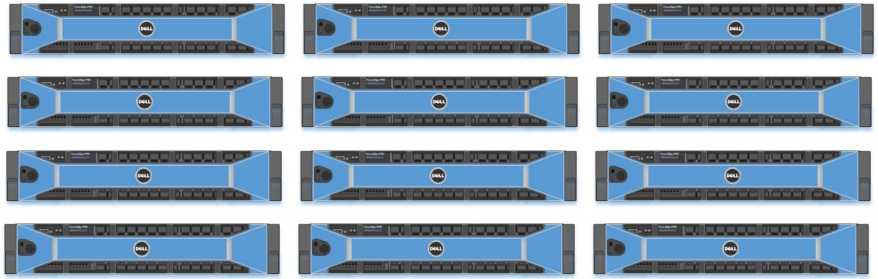
- Limited expansion
- Performance limitations
- Row based acting as a data warehouse

- Non-Elastic
- Expensive
- Closed Ecosystem

Data Lake 1.0

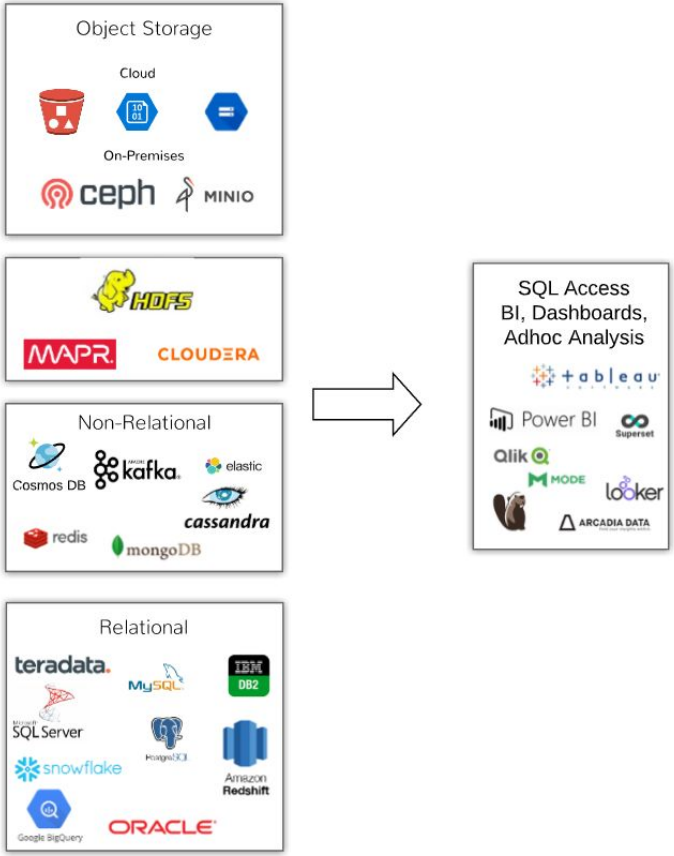


MAPR CLOUDERA HORTONWORKS



- Non-Elastic
- Expensive (commodity was thrown out the window..)
- Lots and lots of moving parts
- Constant upgrades/quick moving projects

Data Warehouse / Lake 3.0



- Low cost object storage the new norm
- Data located in many different systems
- Reluctancy to constantly move data
- Query data “where it lies”
- Open data formats - avoid future lock-in

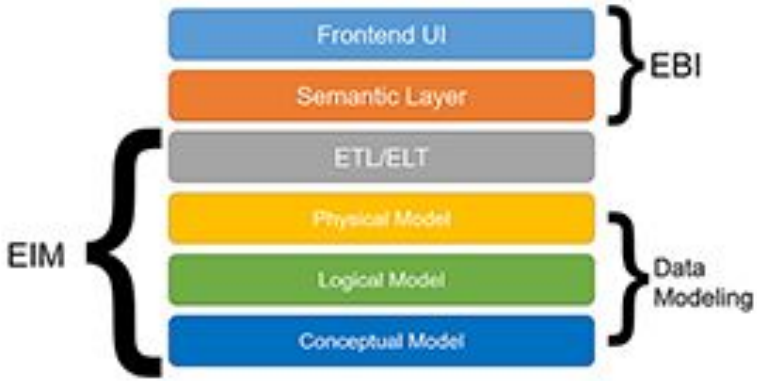
Semantic Layers - What are they?

"A semantic layer is a business representation of corporate data that helps end users access data autonomously using common business terms. A semantic layer maps complex data into familiar business terms such as product, customer, or revenue to offer a unified, consolidated view of data across the organization." - Wikipedia

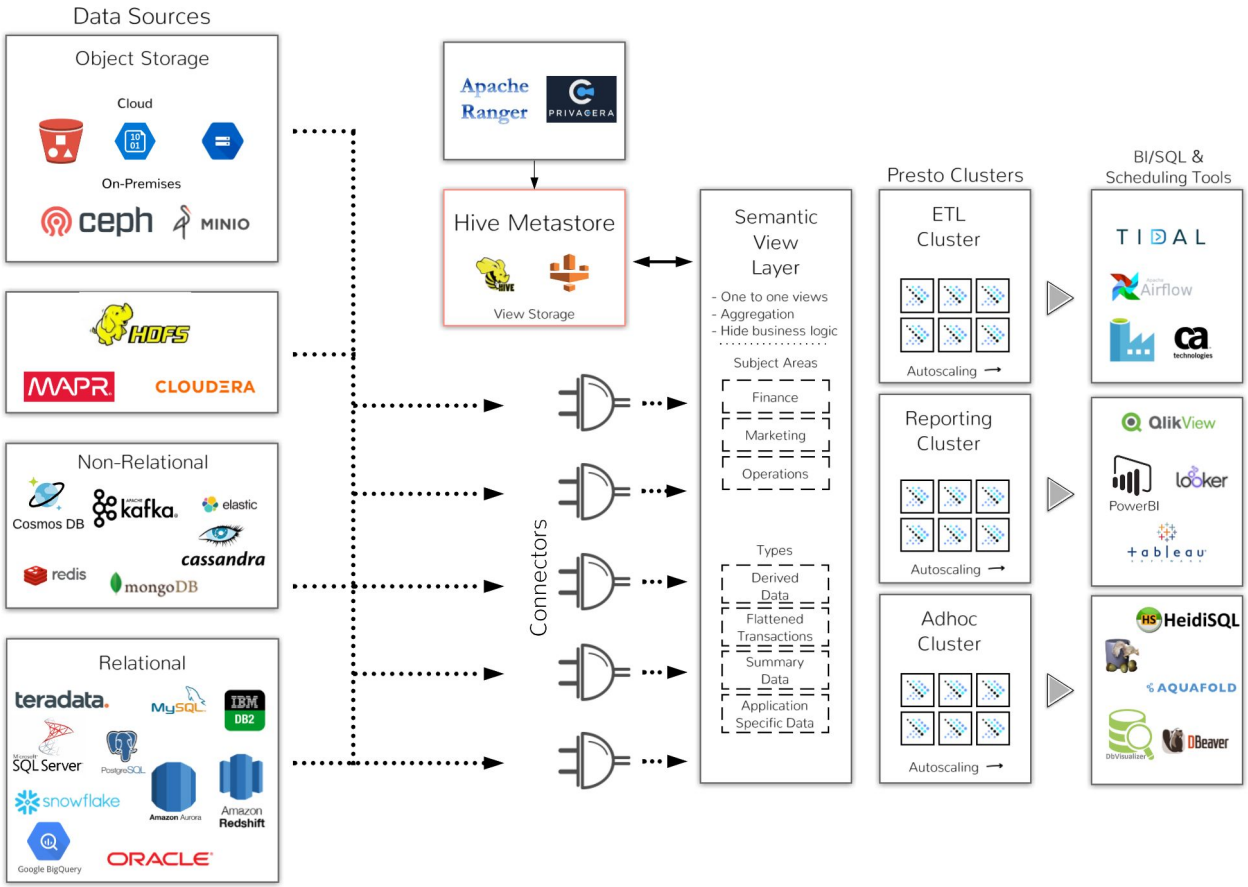
- Hide business logic
- Unified access layer
- Control access/security
- Self service
- Connect from any tool
- Version Control
- Performance
- Low latency

Semantic Layers - Before

Kimball Dimensional DW

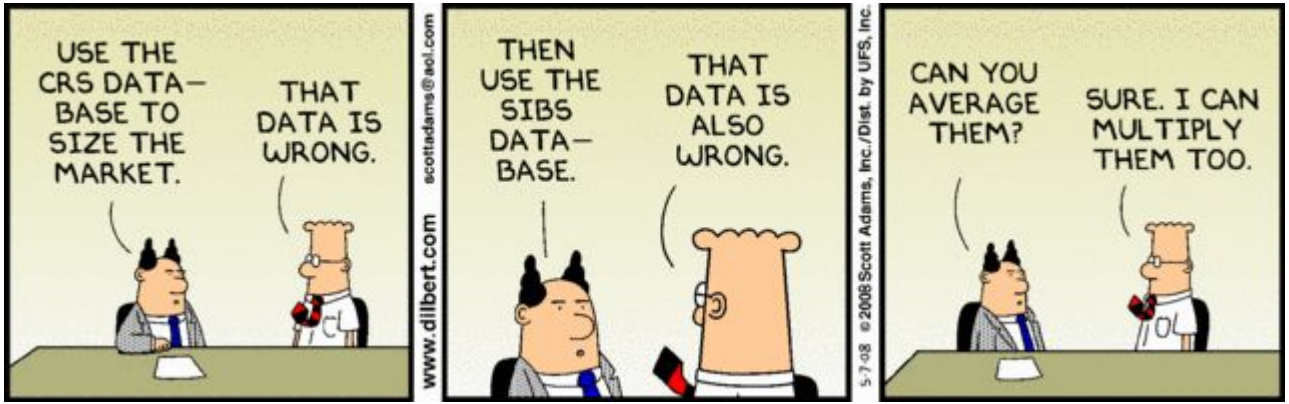


A Federated Semantic Layer - powered by Presto



- Access data in real-time - where it lies
- Connect the tool of your choice
- Different clusters for different functions (chargeback)
- Build business views over a variety of sources
- Additional access control over all sources

FSL - What they are not



- Data still needs to be clean and free of errors
- GIGO - Garbage In / Garbage Out

Demo

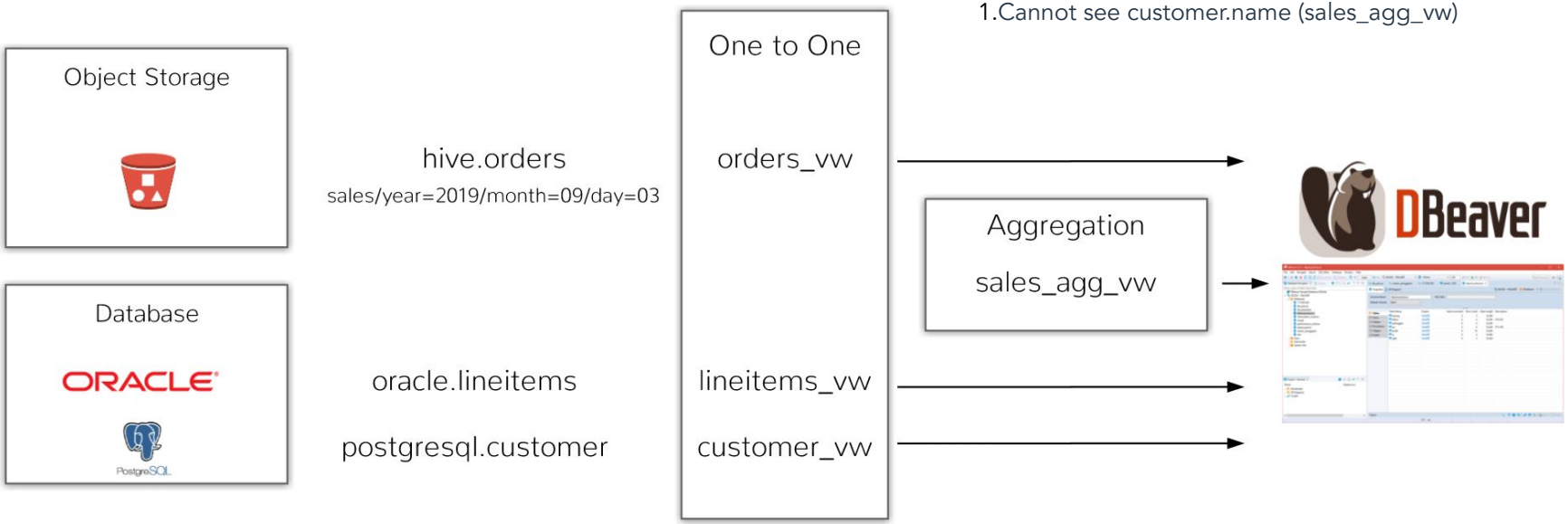
Sources: S3, Oracle and PostgreSQL
One to One Views: orders_vw, lineitems_vw & customer_vw
Agg View: sales_agg_vw

Rules:

Admin
1.Full access

Marketing
1.Cannot view customer.acctbal (customer_vw)
2.Sales_agg_vw.address is masked (sales_agg_vw)
3.Cannot view nationkey = 7 (customer_vw)

Finance
1.Cannot see customer.name (sales_agg_vw)



Demo

Steps:

1. Create one to one views
2. Create aggregation view
3. Select from views
4. Implement rules in Ranger

Create One-to-One Views

customer_vw view:

```
create view hive.sales.customer_vw as
select
*
from
postgresql.public.customer;
```

orders_vw view:

```
create view hive.sales.orders_vw as
select
*
from
hive.sales.orders;
```

lineitem_vw view:

```
create view
hive.sales.lineitem_vw as
select
*
from
oracle.presto.lineitem;
```

Create Aggregate View

sales_agg_vw view:

```
create view hive.sales.sales_agg_vw as
select
        c.custkey,
        c.name,
        c.address,
        c.nationkey,
        round(sum(o.totalprice)) as total_sales
from
        hive.sales.orders o,
        postgresql.public.customer c
where
        o.custkey = c.custkey
group by
        c.custkey,
        c.name,
        c.address,
        c.nationkey;
```

Ranger Rules

Rule Name	Group	
all - admin - hiveservice	-	Allow admin to see all
CustomerVW-Marketing	Marketing	Cannot view customer_vw.acctbal
MaskAddressSalesAGGVW	Marketing	sales_agg_vw.address is masked
NoViewNationKey7	Marketing	Cannot view nationkey = 7
Sales_AGG_VW-Finance	Finance	Cannot see customer.name

Questions?

Helper Notes

```
drop view hive.sales.lineitem_vw;
drop view hive.sales.customer_vw;
drop view hive.sales.orders_vw;
drop view hive.sales.sales_agg_vw view;
```

Cannot view customer.acctbal (customer_vw) Ranger Rule: CustomerVW-Marketing
-- Try to select from all of the columns in the customer_vw. acctbal won't be allowed
select * from hive.sales.customer_vw limit 5;
--SQL Error [100050] [HY000]: [Starburst][Presto](100050) Query failed: Access Denied: Cannot select from columns [nationkey, mktsegment, address, phone, custkey, name, comment, acctbal] in table or view sales.customer_vw.
-- Select the columns that this user has access to
select custkey,name,address,nationkey,phone,mktsegment,comment from hive.sales.customer_vw limit 10;

Sales_agg_vw.address is masked – Ranger Rule: 2-MaskAddressSalesAGGVW
-- Mask the address column form the sales_agg_vw view
select * from hive.sales.sales_agg_vw limit 10;

Cannot view nationkey = 7 – Ranger Rule: 3-NoViewNationKey7
-- Try to select nationkey = 7 from customer_vw
select custkey,nationkey from hive.sales.customer_vw where nationkey = 7;
-- Show other nationkeys return with no filtering
select custkey,nationkey from hive.sales.customer_vw;

Cannot see customer.name (sales_agg_vw) – Ranger Rule: Sales_AGG_VW-Finance
-- Try to select all of the columns from the sales_agg_vw:
select * from hive.sales.sales_agg_vw limit 10;
-- Leave out the name column and see if the query exectues
select nationkey,address,custkey,total_sales from hive.sales.sales_agg_vw limit 10;