



# Optimizing data lakehouses with Starburst

Subtitle Goes Here

Starburst Academy  
v3.1.1

# Optimizing data lakehouses with Starburst

This course features instructor-led discussions, demonstrations, and hands-on exercises designed to build a working knowledge of the Starburst query engine.

Upon completion, you will gain a more thorough awareness of Starburst architecture, with a specific focus applied to best practices for data lake based schemas, including table formats and partitioning, file formats and sizes, and other optimization techniques.

# Course introduction

**Who, what, where, when & why**

# Course objectives

- Use Starburst as a single point of access for multiple data sources and federate queries across them
- Evaluate and describe how queries are executed within a Starburst cluster
- Use Hive and Iceberg table formats; construct, populate, query, and modify partitioned tables
- Employ file size/format/hierarchy strategies to improve query performance
- Understand the role of the Cost-based optimizer and read query plans to ensure optimizations are occurring as expected and to identify possible issues
- Create role-based access control policies for table operations
- Build a data engineering pipeline with Starburst Galaxy

# What to expect

- Blend of conceptual and practical
- Focused learning objectives
- Instructor demonstrations
- Plenty of hands-on labs
- Interactive discussions
  - Ask questions
  - Validate assumptions
  - Discuss use cases
  - **FEEDBACK IS OXYGEN**

# Course agenda

## 1 - Starburst features

- Overview & architecture
- Web UI
- Connectors & catalogs
- Client tools integration

## 2 - Data lake tables

- Separation of storage & compute
- Schema on read

## 3 - Data lake performance

- Limit data exchanges
- File format options
- Small files problem
- Partitioning & bucketing

## 4 - Table formats

- Moving beyond Hive
- Compare/contrast alternatives
- Exploring Delta Lake

## 5 - Apache Iceberg

- Table format architecture
- Creating tables
- Insert, delete & update

## 6 - Advanced Iceberg

- CDC with merge
- Schema & partition evolution
- Snapshots & compaction

## 7 - Parallel processing

- Divide & conquer
- Beyond single stage queries

## 8 - Cost-based optimizer

- Benefits of statistics
- Query plan analysis

## 9 - Access control

- Configuration options
- RBAC & ABAC

## 10 - Data pipelines

- Definition & differentiation
- Reference architecture

# Introductions

Please feel free to share as much (or as little) as you feel comfortable with

- You! What's your name?
  - Where are you location? Working remote?
  - High-level role (company and/or project)
  - Any hobby/activity you want to share?
- *Identify the email address you want for your Starburst training ID*
- Prior experience with
  - Trino (Presto) and/or Starburst (Enterprise and/or Galaxy)
  - Other open-source big data solutions such as Apache Hive or Spark
- **Any expectations that appear to not be addressed in the agenda**

# Starburst features

Overview

# Lesson objectives

## Starburst features: Overview

1. Understand the negative consequences of data silos.
2. Describe, and understand the benefits of, the Modern Data Lake.
3. Understand the major layers of the Starburst Data Lake Analytics Platform.
4. List industries and companies where Starburst provide impact.
5. Under the two consumption models.

# Is this your data strategy?



# The reality and our approach

**ETL is required...**

**when you already know what data is needed for reporting.**

**Data warehouses are great...**

**for highly structured, repeatable analytics on limited datasets.**

**Data centralization is fine...**

**for simple data environments and slower-moving data orgs.**

**ETL *after* you know what is needed.**

Starburst enables data engineers to *explore* data at the source and *iterate* on their ETL workloads in the data lake.

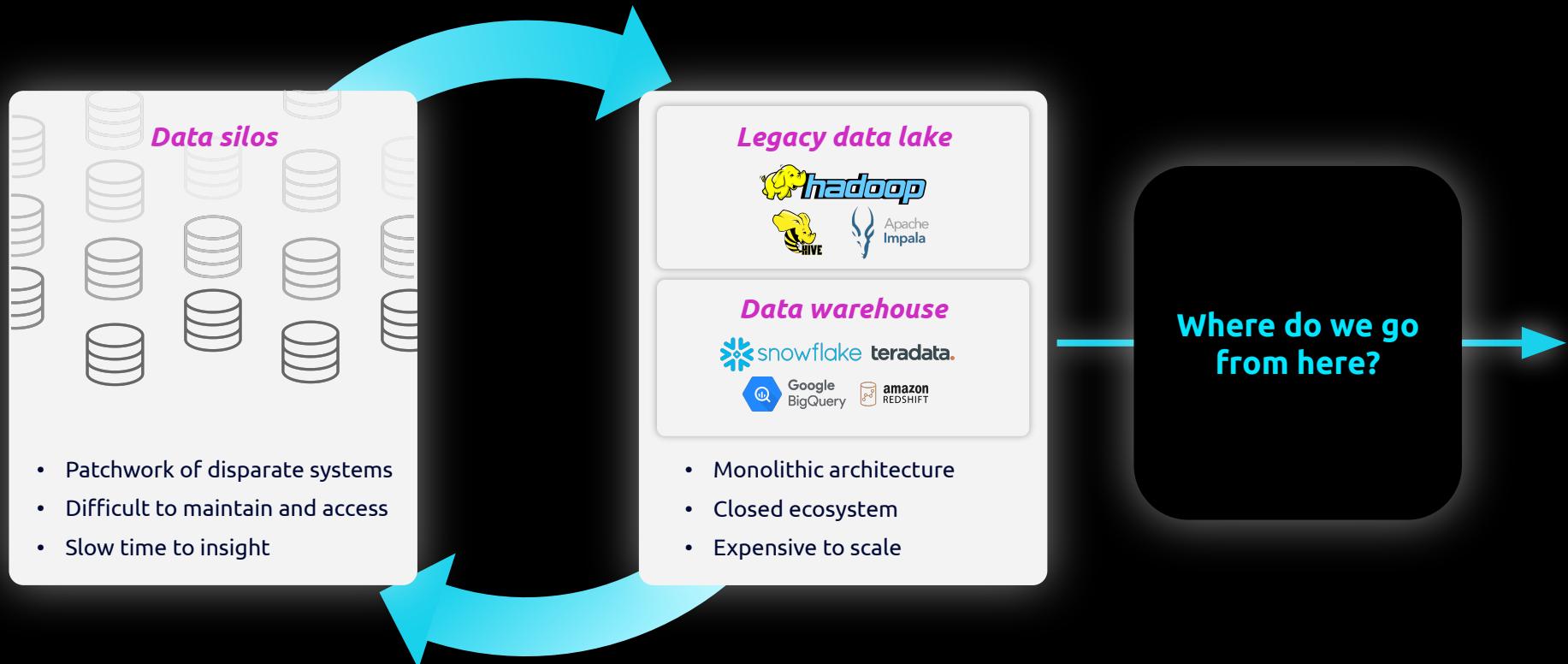
**Don't rely on the data warehouse for *everything*.**

Starburst *frees* data teams to choose the best data architecture for their needs, by federating across all data sources.

**Decentralization is *inevitable* as you grow.**

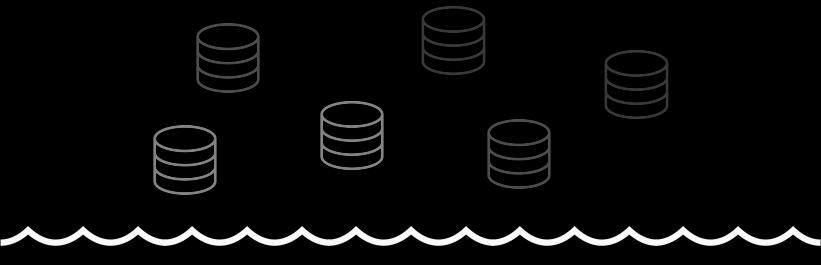
Starburst helps to *future-proof* data access and productivity for your data-driven organization.

# Data silos continue to be a challenge, despite efforts to create a “single source of truth”



# The Modern Data Lake

Global federated access to data sources beyond the lake



MPP query engine

Open table formats



Open file formats



Commodity storage & compute



Object storage



Elastic compute

## Benefits

Single point of access and governance for all data

Advanced warehouse-like capabilities

Vendor agnostic

Scalable, cost effective

# Starburst enhances your data lake additional flexibility, security, and ease of use for faster and broader access to all your data

	<b>Cloud data warehouse</b>	<b>Traditional data lake</b>	<b>Modern data lake powered by Starburst</b>
<b>Enables flexible data architecture</b>	✗ Limited ability to access data outside of warehouse	✗ Complex data management & planning	 Supports open file and table formats
<b>Cost-effective scaling</b>	✗ Vendor lock-in leads to ballooning costs over time	✗ Expensive proprietary software and hardware	 Runs on commodity cloud storage and compute; 50% lower TCO
<b>Fast time to insight</b>	✗ New data must be ingested via ETL process	✗ Special tools & configurations needed to access external data	 Query data at the source when needed; 90% faster time to insight; 10x faster query speed
<b>Global security and compliance</b>	✗ Difficult to manage data outside of warehouse and across regions	✗ Difficult to manage data outside of lake	 Single governance layer for all data
<b>Creating &amp; sharing curated data sets</b>	 Possible with external sharing	✗ Specialized skills and tools needed	 Data Products, Global Search

# Trino is the query engine trusted by industry leaders at PB scale



**25PB** on S3



**1 Exabyte of Data**  
100PB weekly data  
**1200** nodes  
**2.5M** queries/week



**600PB** on S3  
**1000** nodes



**10PB** daily read data  
**250k** queries per day



**300PB** data lake

*But Trino requires **extensive resources** to run successfully...*

**Management:** All manual. No autoscaling

**Security:** No built-in security integrations

**Access Control:** Requires 3rd parties for RBAC

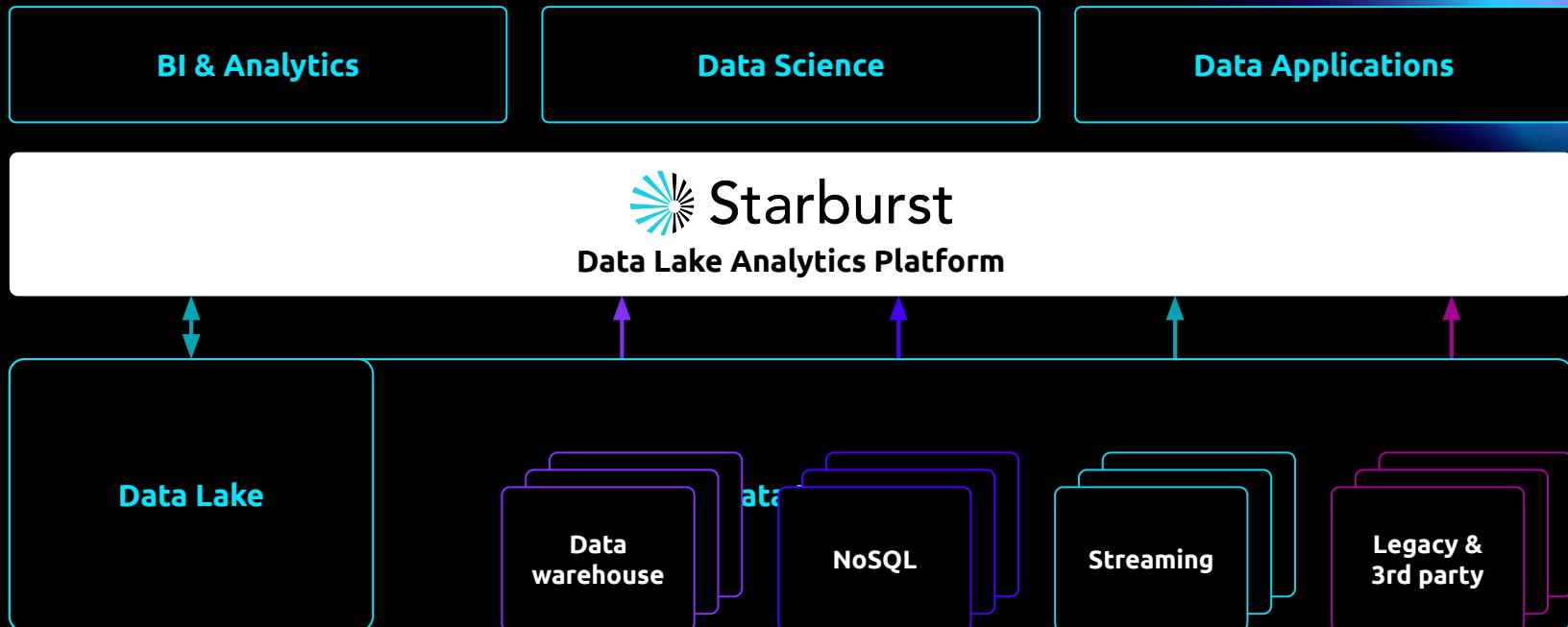
**Support:** No support team, reliant on community responsiveness



**\$\$\$\$**

*(and time!)*

# Starburst activates the data in and around your lake





# Starburst Data Lake Analytics Platform

**Modeling / Semantic Layer**

Data Products

Materialized Views

Catalog

Global Search

Data Profiling

Metastore

**Query Engine**

powered by  
 trino

Warp Speed

Cost-based optimizer

Cluster autoscaling

Dynamic Filtering

Fault-tolerant execution

Accelerated Parquet reader

Runs at PB scale

Query Federation

**Security & Governance**

RBAC & ABAC

SSO & IAM

E2E Encryption

Row & Column Masking

Lineage

Monitoring & Logging

Management APIs

Service Accounts

**Data Access**

Stargate

Deploy anywhere

Data source connectors

**Data sources**



Hybrid, cross-region,  
cross-cloud



On-prem,  
private cloud,  
public cloud



All your data

**3rd party integrations**

Data Science

BI / Visualization

Data catalogs

Security

Access controls

Orchestration

Data ingestion

# Starburst powers innovation across every industry



## Financial Services

- Fraud detection
- Anti-money laundering
- Risk management



## Consumer

- Customer 360
- Marketing analytics
- Supply chain analytics



## Healthcare & Life Sciences

- Patient care optimization
- Regulatory compliance
- Health record analytics



## Technology

- Product analytics
- In-product functionality
- Security / Log analytics



## Telco

- Marketing operations
- Customer care
- Capacity planning

### Trusted by industry leaders



NORDSTROM





# Starburst

## Data Lake Analytics Platform

\$3.3B  
valuation

\$414M  
raised

### Leading investors

andreessen.  
horowitz

Index  
Ventures

COATUE

 salesforce ventures

ALTIMETER

ALKON  
CAPITAL MANAGEMENT

B Capital Group

Original creators of



Deployed at **exabyte scale** at 4 out of 5 FAANG companies; adoption across thousands of companies globally.



Originally created at Facebook to query **300PB Hadoop cluster**; thousands of active users today.

200+  
customers

100%  
YoY growth

85  
NPS

### Industry-recognized Leader

Gartner Market Guide for Analytics Query Accelerators  
GigaOm Radar Report for Data Lakes and Lakehouses  
G2 Enterprise Grid for Big Data Analytics Software

# Flexible consumption models

## Starburst Enterprise

**Deploy anywhere**

Enterprise-grade software backed by professional support and services

**On prem, in the cloud, hybrid, and multi-cloud**



## Starburst Galaxy

**Built for the cloud**

Fully managed cloud data lake analytics built and supported by the creators of Trino

**Available on leading public clouds**



**Available via marketplaces**



**Alibaba Cloud**

Hewlett Packard  
Enterprise



**Red Hat  
OpenShift**

# Lesson summary

## Starburst features: Overview

1. Data can too easily become siloed. This is both expensive and inefficient for the entire organization.
2. Benefits of the Modern Data Lake include a single point of access, governance for all data, advanced warehouse-like capabilities, vendor agnostic, scalable, and cost effective.
3. The major layers of the Starburst Data Lake Analytics Platform are Data Access, Security & Governance, Query Engine, and the Modeling / Semantic Layer.
4. Starburst powers innovation across every industry and Trino is the query engine trusted by industry leaders at PB scale.
5. Flexible consumption models offer you the options of paying for what you configure and/or paying for what you use.

# Starburst features

Architecture

# Lesson objectives

Starburst features: Architecture

1. Understand the role that workers and coordinators play in Starburst clusters.
2. Explain how Starburst logical architecture functions in relation to different data sources.
3. Analyze a typical Starburst execution flow in detail.
4. Differentiate Starburst from Trino.

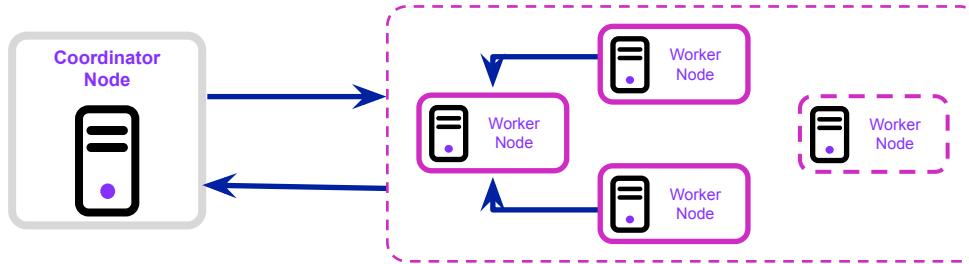
# Server stereotypes

## Coordinator node

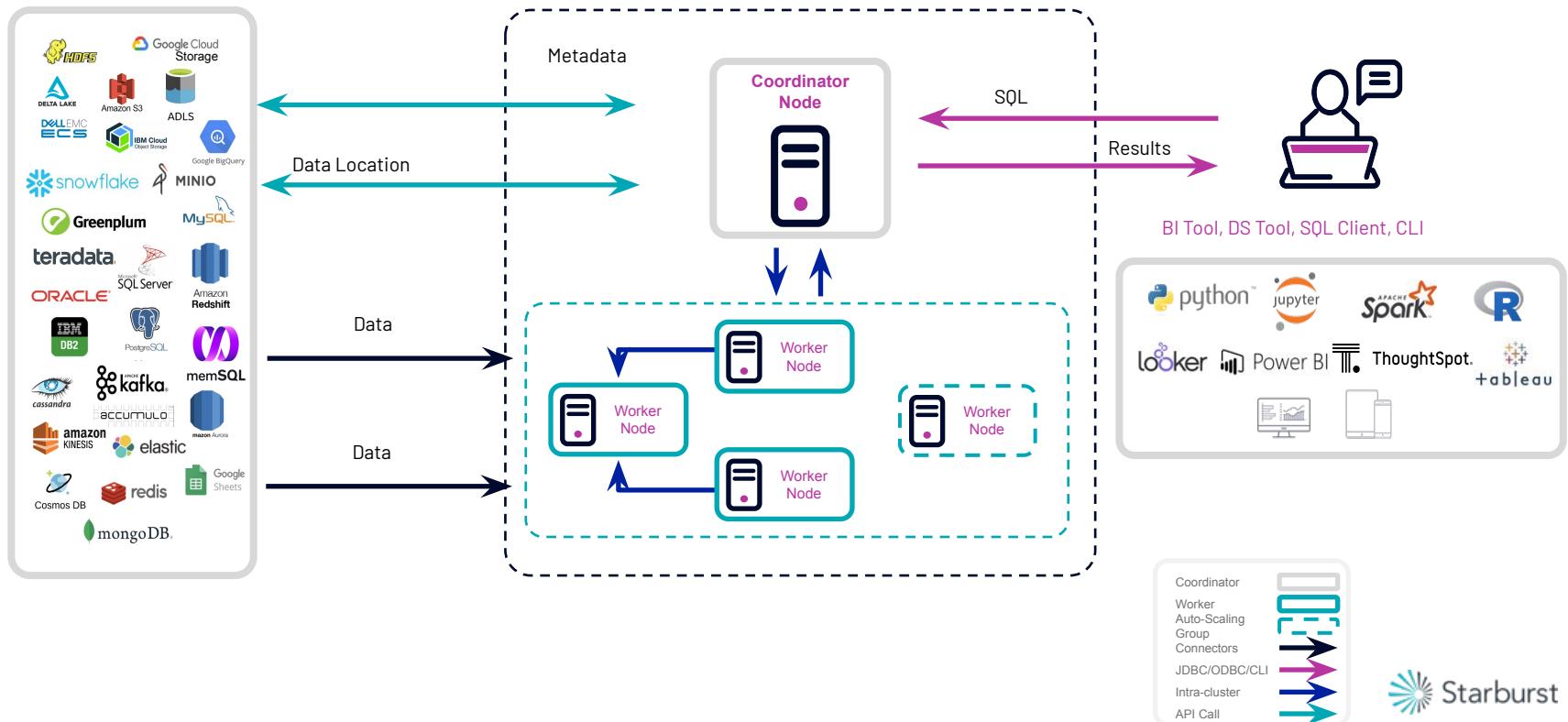
Server that is responsible for parsing statements, planning queries, and managing Trino worker nodes.

## Worker nodes

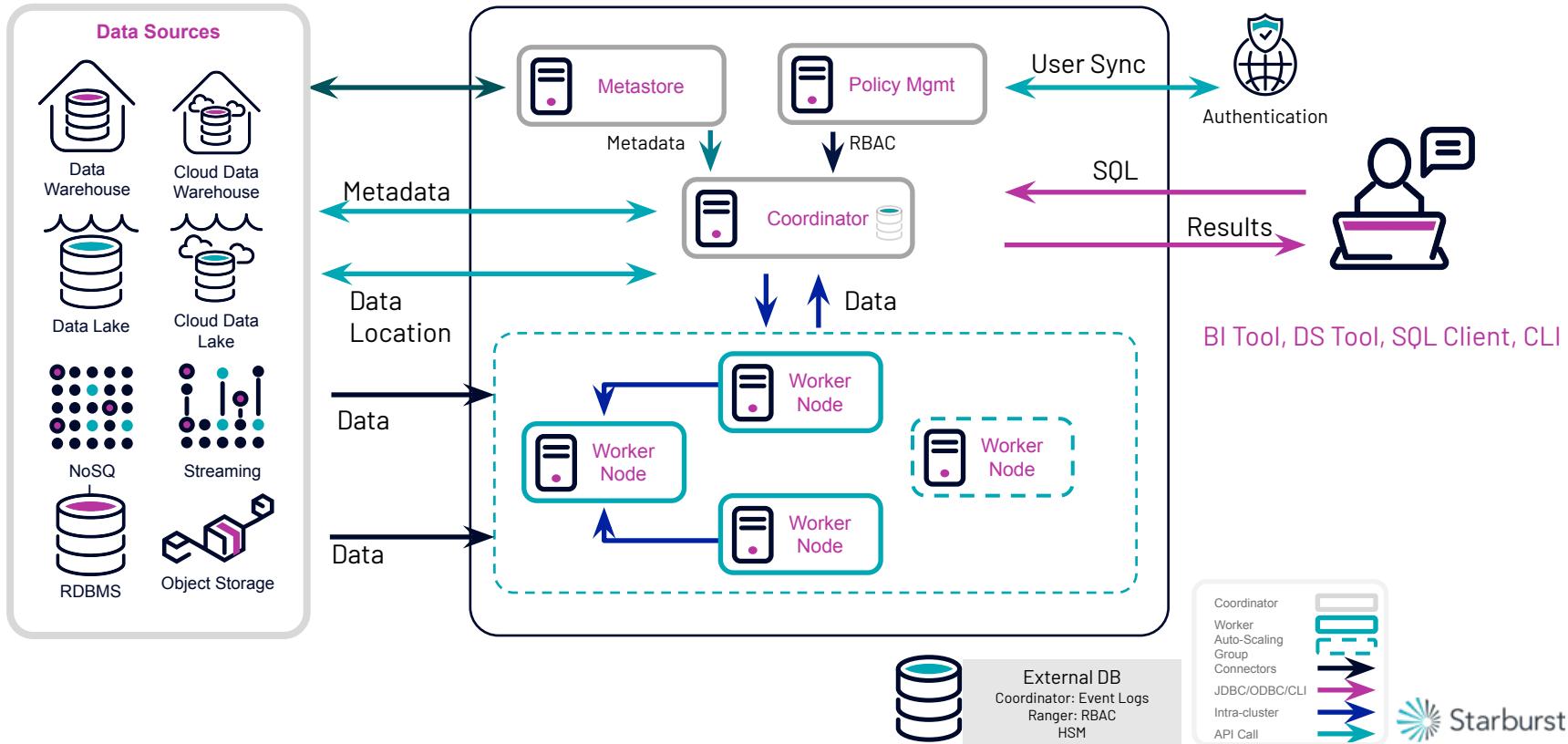
Server which is responsible for executing tasks and processing data. Worker nodes fetch data from connectors and exchange intermediate data with each other.



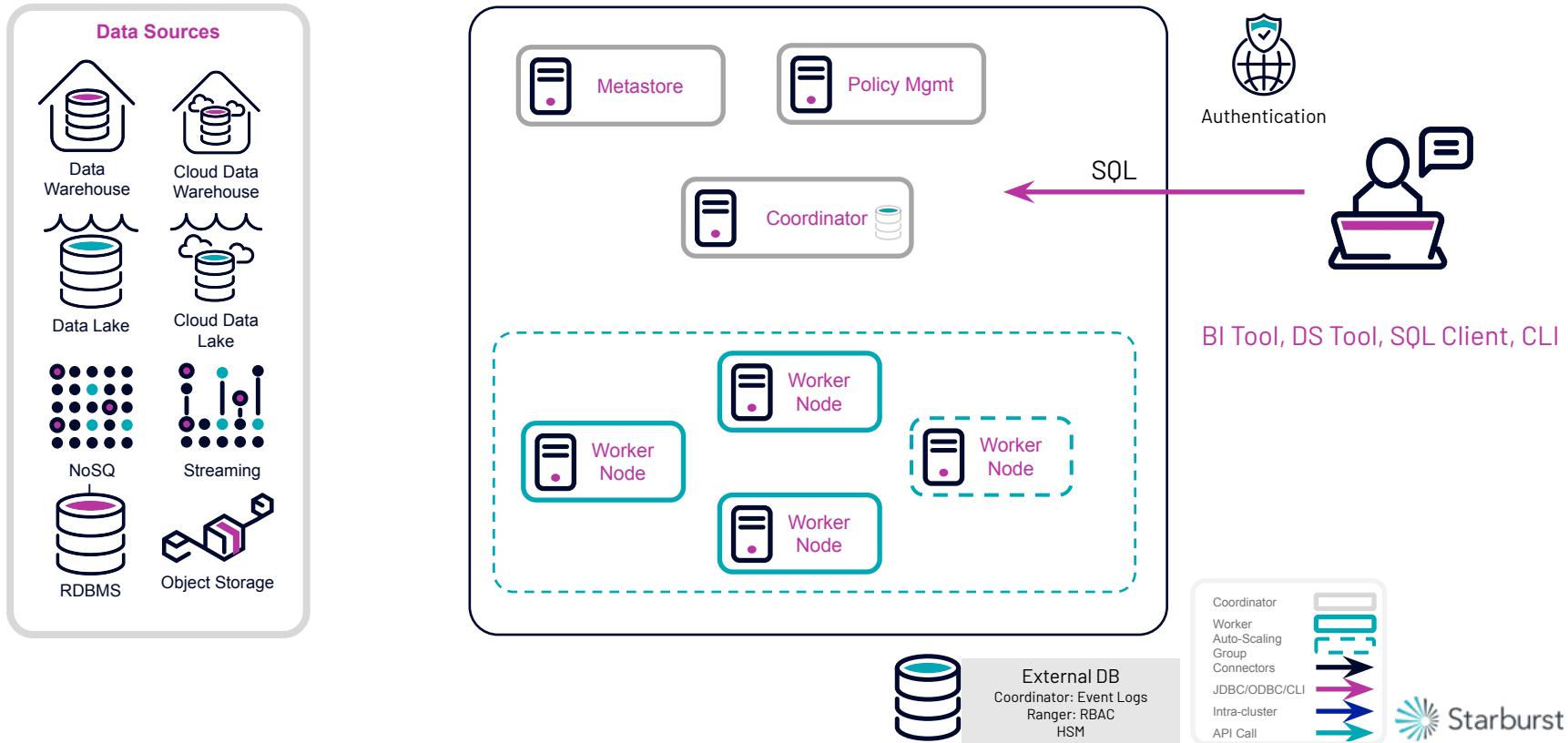
# Data sources & tools integrations



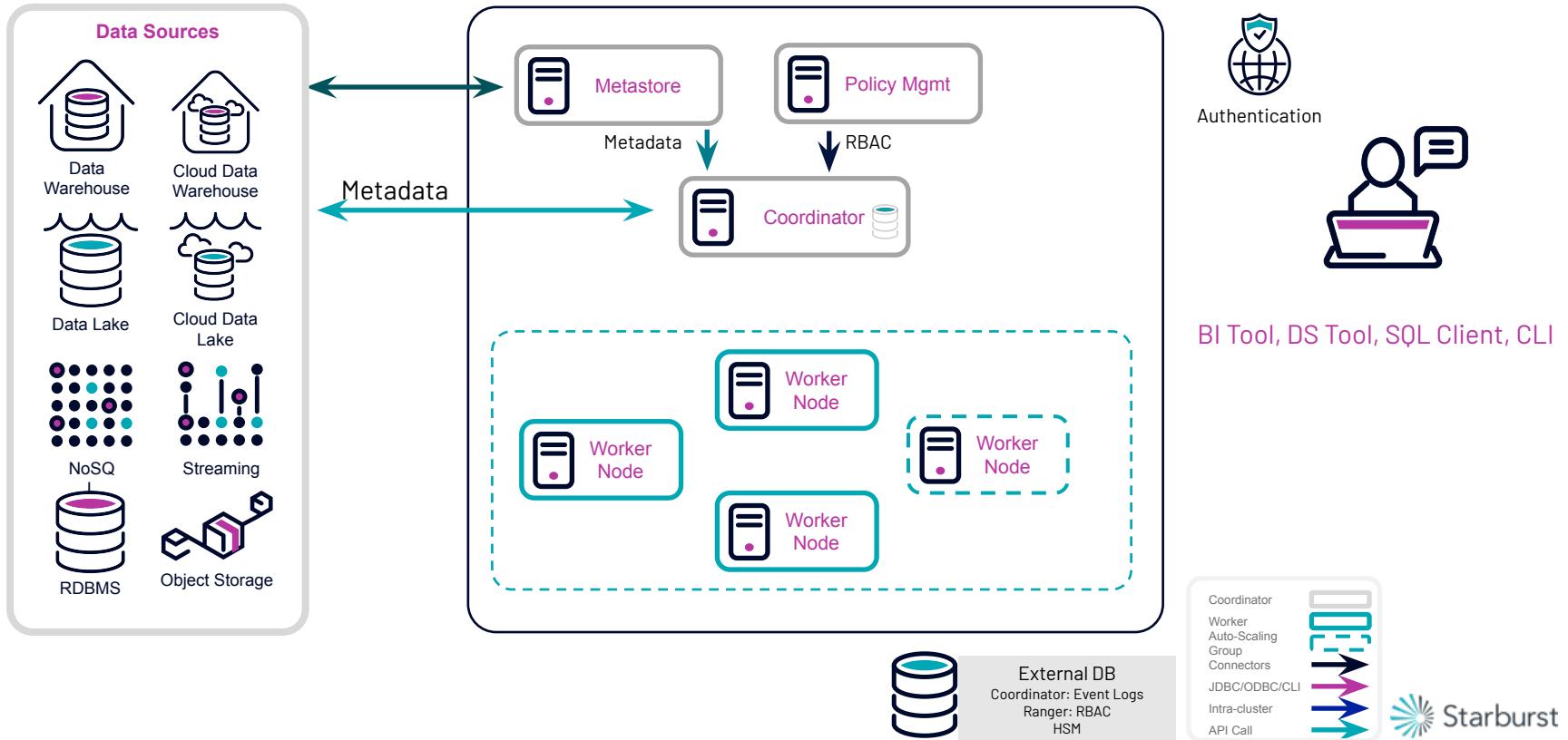
# Starburst logical architecture



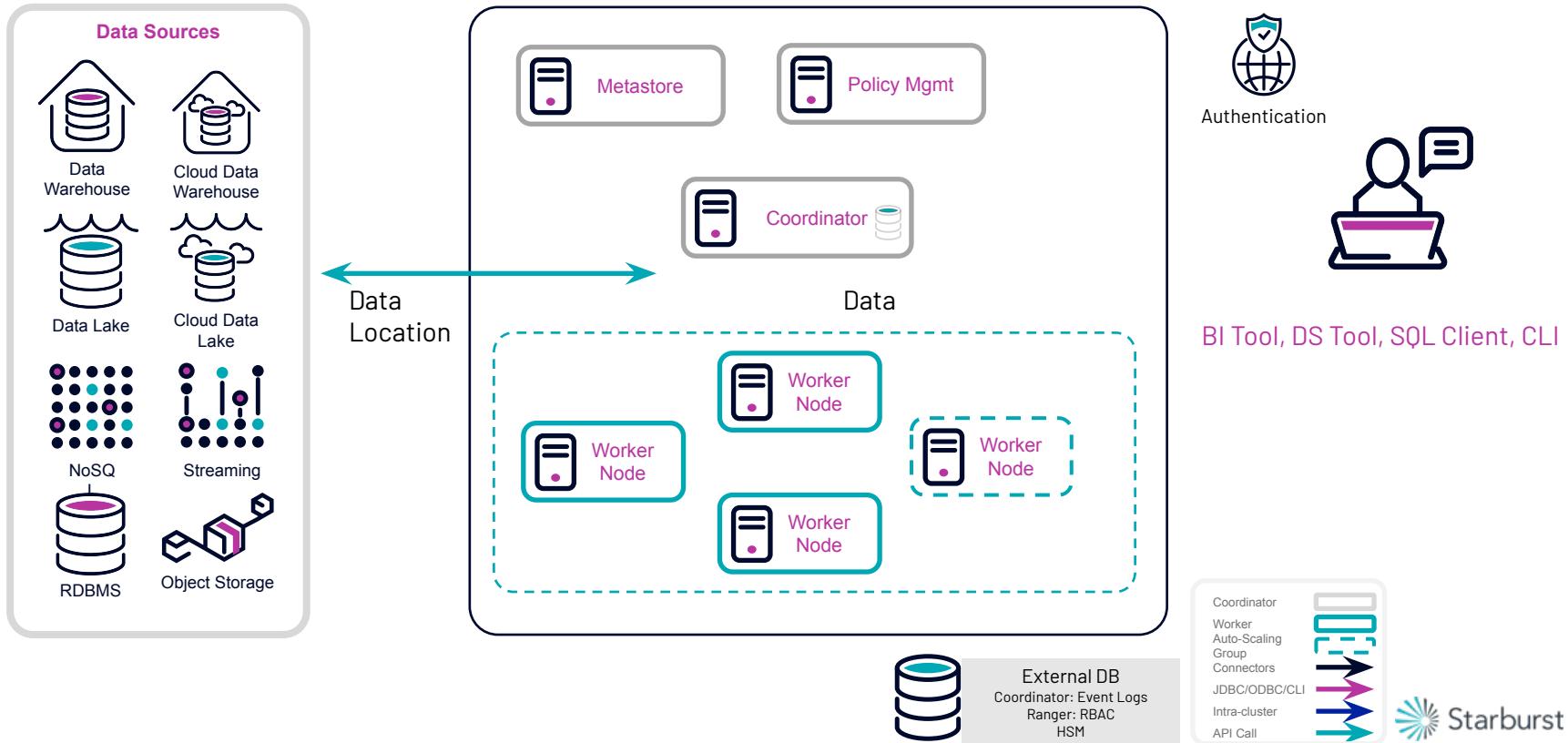
# Query step: Statement submitted



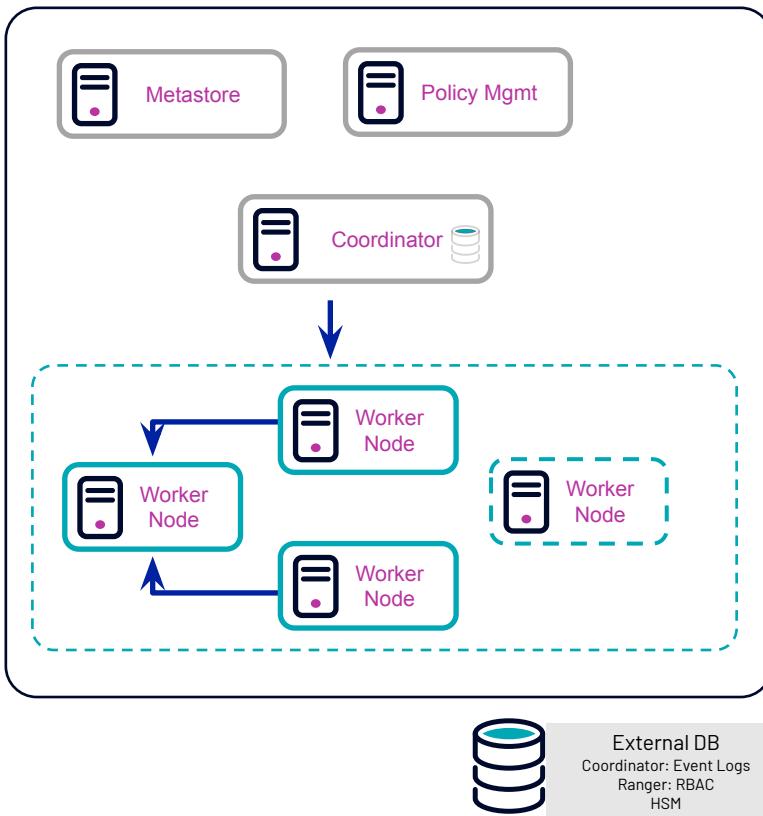
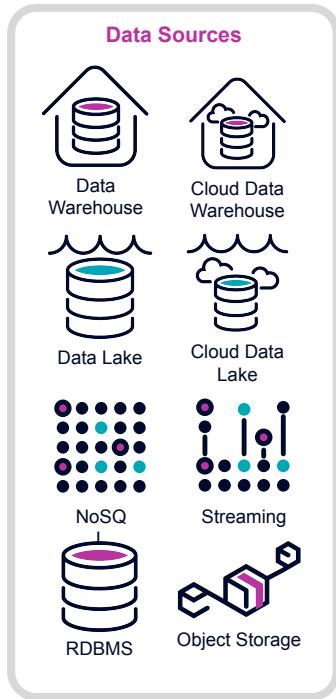
# Query step: Query validated



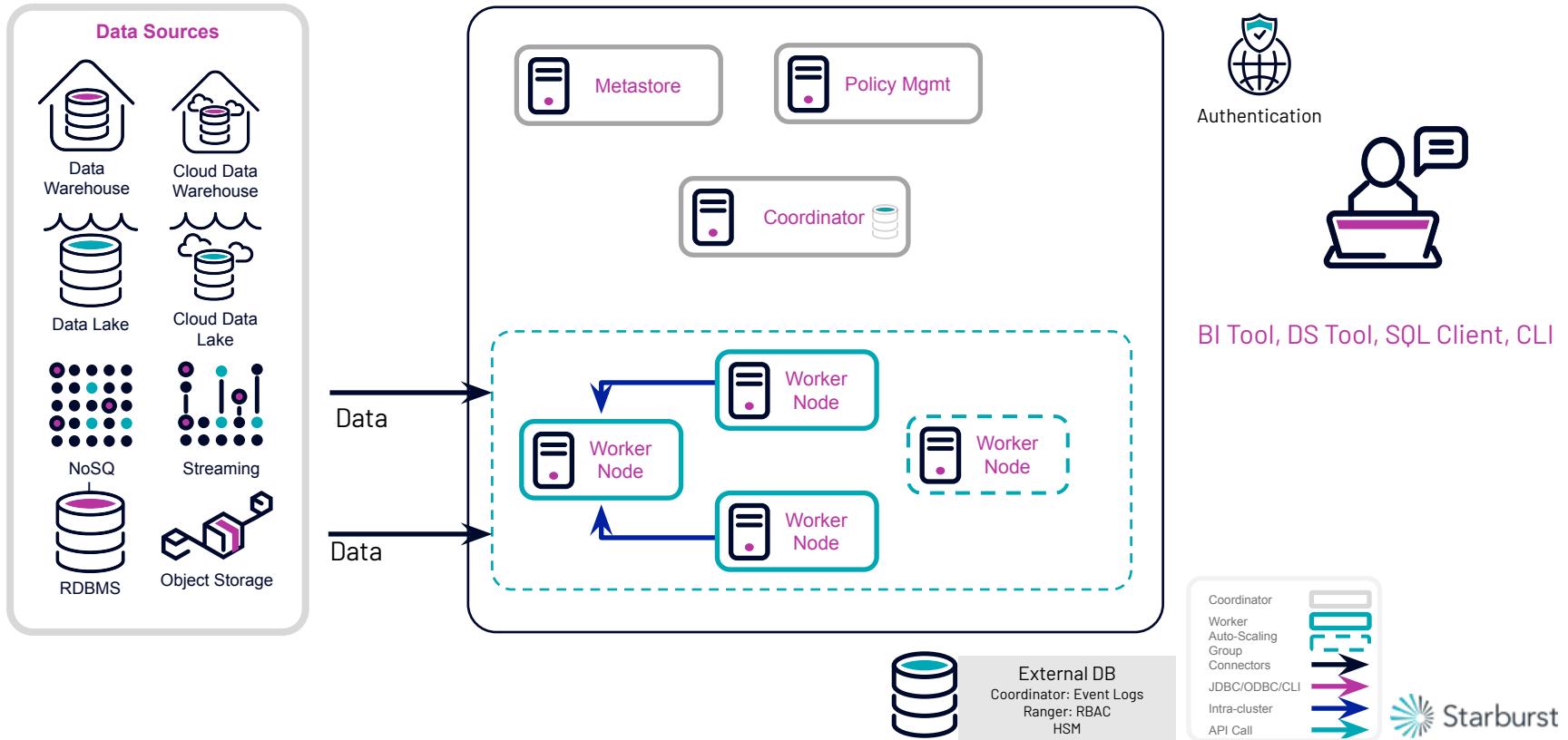
# Query step: Schedule created



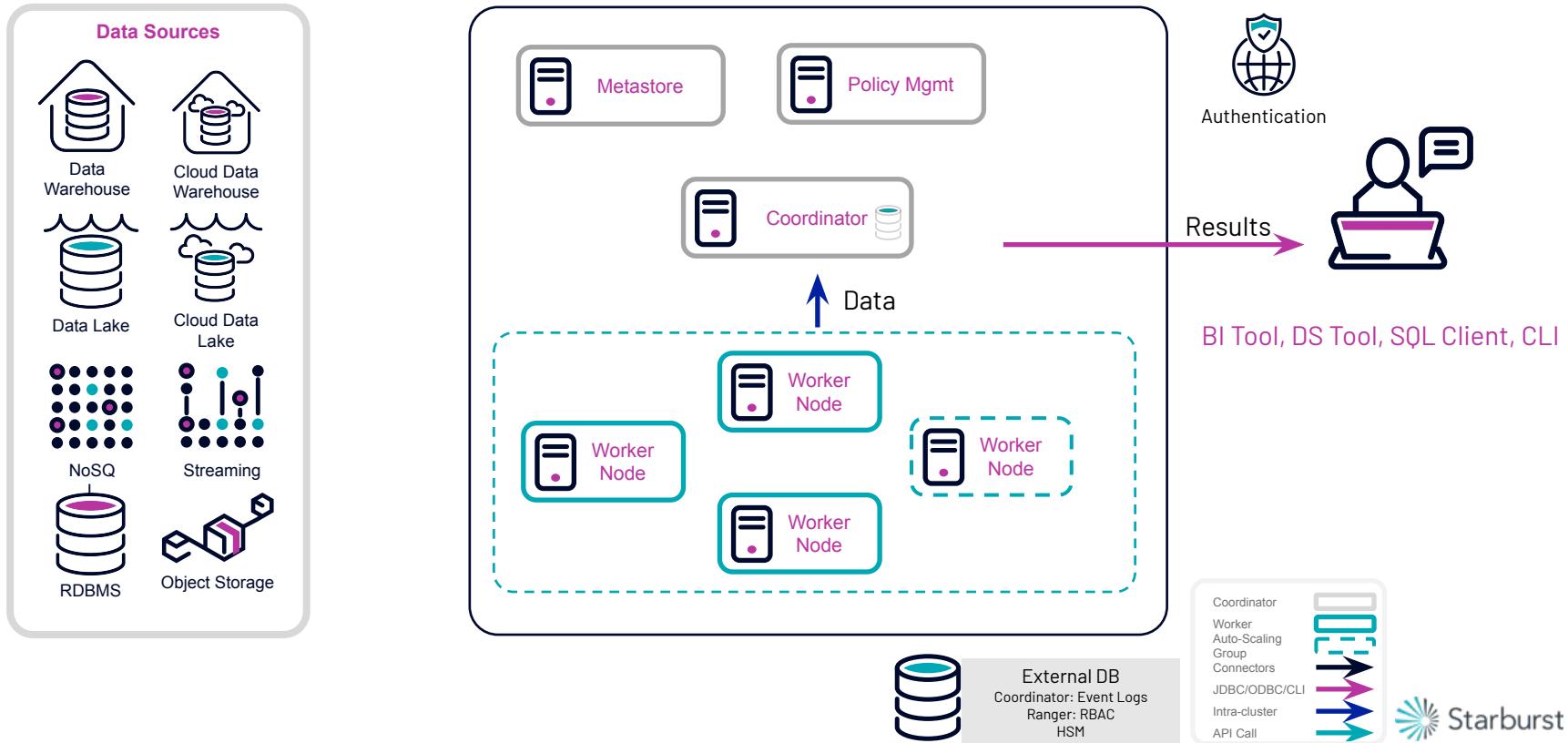
# Query step: Tasks assigned



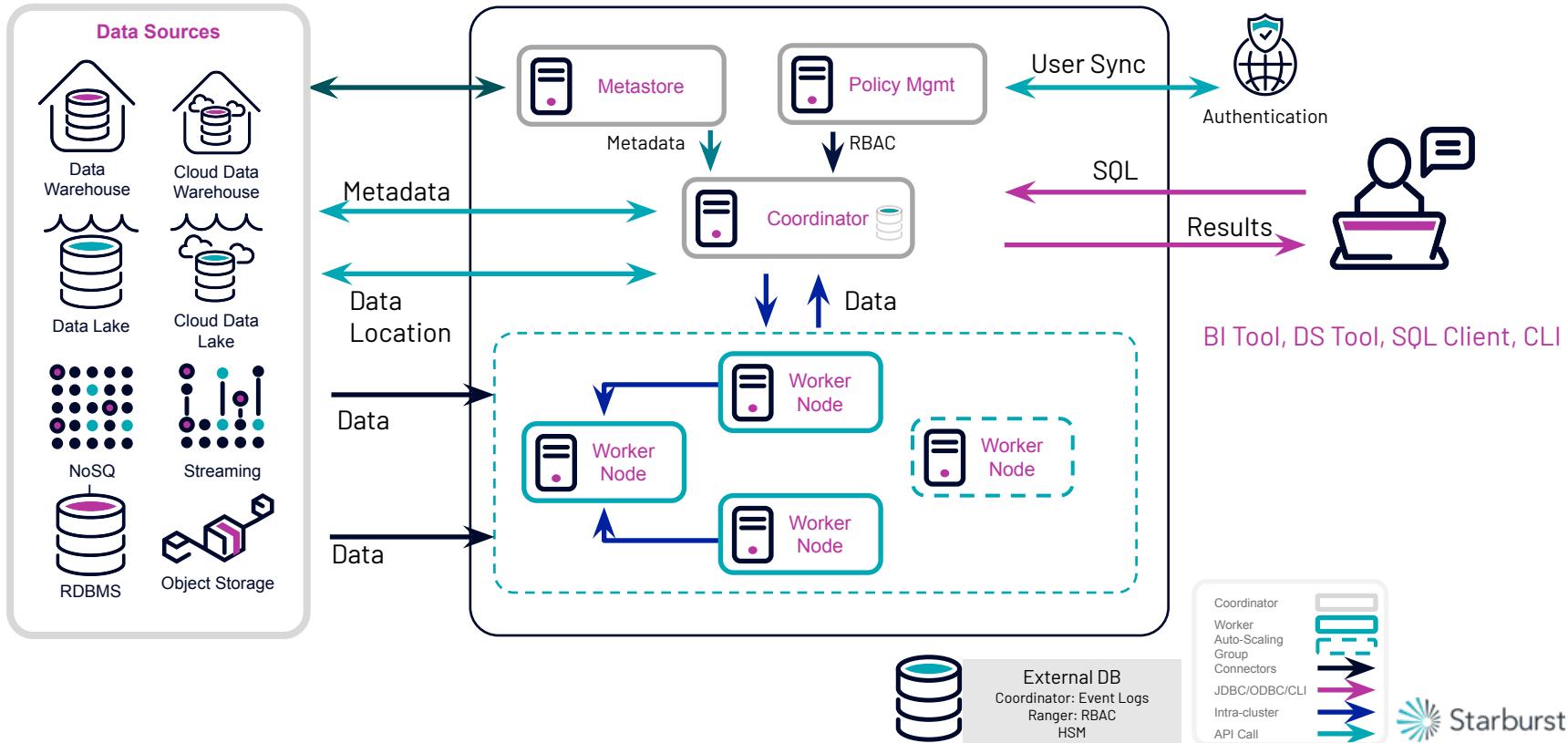
# Query step: Tasks executed



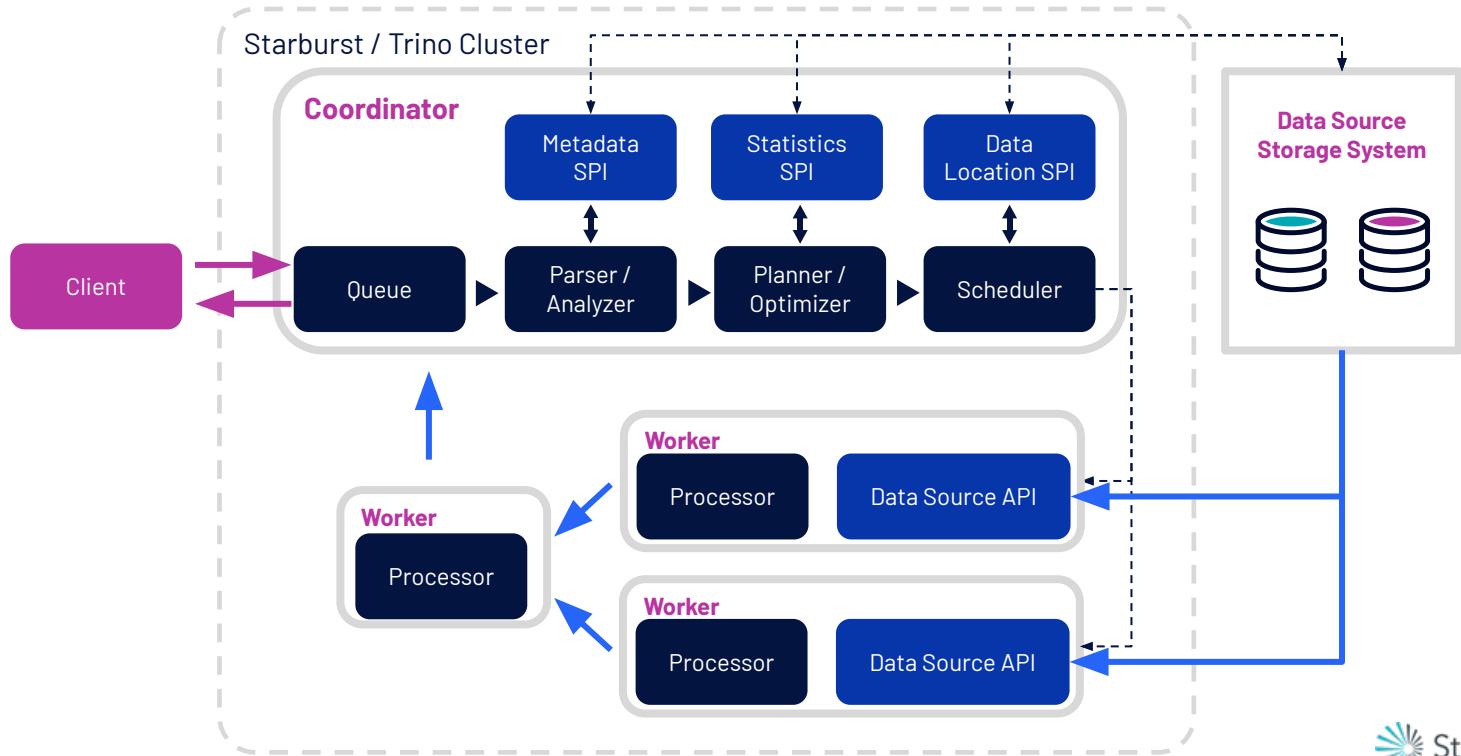
# Query step: Results returned



# Starburst logical architecture



# Starburst / Trino execution flow



# Lesson summary

## Starburst features: Architecture

1. Starburst clusters deploy a single coordinator and multiple workers.
2. The coordinator analyzes the user requirements, allocates resources, and distributes tasks to the workers.
3. The workers perform the tasks and return the results to the coordinator.
4. Starburst Galaxy and Starburst Enterprise both allow users to connect to multiple data sources. This is one of the ways in which optionality is ensured.
5. Starburst Galaxy and Starburst Enterprise offer fully-fledged data lake analytics platforms built on top of Trino, an open-source query engine.

# Starburst features

Web UI

# Lesson objectives

Starburst features: Web UI

1. Understand how to sign into Starburst web UI.
2. Identify the key components of the Starburst web UI.
3. Explain how to input SQL in Starburst web UI.



Welcome back!

## Sign in to Starburst Galaxy

Email \*

Password \*



[Forgot your password?](#)

[Sign in to Starburst Galaxy](#)



Query

11/23/22, 12:52 PM

7 table join

11/29/22, 12:58 PM

11/29/22, 1:00 PM

+

Query editor

Saved queries

Query history

Catalogs

Clusters

ACCOUNT

Admin

Access control

Roles and privileges

Cloud settings

AWS

Azure

## Cluster explorer

- aws-us-east-1-free
  - lakehouse
    - information\_schema
    - serverlogs
      - logs\_5min\_ingest\_csv
  - mysql
  - postgresql
  - sample
    - demo
      - astronauts
      - missions
        - id integer
        - company\_name varchar
        - location varchar
        - date varchar
        - detail varchar
        - status\_rocket varchar
      - cost double
      - status\_mission varchar

Run selected (limit 1000)

aws-us-east-1-free

Select catalog

```
2
3   DESCRIBE "sample"."demo"."missions";
4
5   SELECT COUNT(*) FROM sample.demo.astronauts;
6
7   SELECT * FROM sample.demo.astronauts;
8
9   SELECT original_name, year_of_birth, nationality, selection, occupation, mission_t
10  .| FROM sample.demo.astronauts;
```

 Finished

Avg. read speed

Elapsed time

0.43s

Query details

Trino UI

Download

original_name	year_of_birth	nationality	selection
ГАГАРИН Юрий Алексе...	1934	U.S.S.R/Russia	TsPK-1
ТИТОВ Герман Степан...	1935	U.S.S.R/Russia	TsPK-1
Glenn, John H., Jr.	1921	U.S.	NASA Astronaut Group 1
Glenn, John H., Jr.	1921	U.S.	NASA Astronaut Group 2
Carpenter, M. Scott	1925	U.S.	NASA-1
НИКОЛАЕВ Андриян Г...	1929	U.S.S.R/Russia	TsPK-1
НИКОЛАЕВ Андриян Г...	1929	U.S.S.R/Russia	TsPK-2



# Query editor

## A full environment to write and execute SQL statements

- Explore clusters and the catalogs, schemas, tables, views, and columns of their connected data sources
- Run and download results
- SQL function and grammar auto-complete
- Manage queries with separate editor tabs
- Review saved queries, the history of the query statuses, and their performance with the query execution diagram

The screenshot displays the Starburst Galaxy interface, which is a full environment for writing and executing SQL statements. The interface is divided into several sections:

- Left Sidebar:** Contains links for "Query editor", "Saved queries", "Query history", "Catalogs", and "Clusters". It also includes sections for "ACCOUNT" (Admin, Access control, Cloud settings) and a "Recent" section for saved queries.
- Top Right:** Shows a "Saved queries" panel with a "Recent" tab and a "Create new query" button. A specific query titled "Number of Hashtags" is listed, showing it was last updated on Jul 15, 2022, at 11:14 AM.
- Middle Left:** The "Cluster explorer" pane shows the schema of the "rankings" table from the "landing\_twitter" database. The table has columns: id (varchar), url (varchar), date (varchar), content (varchar), user (rowusername varchar, id integer), replycount (integer), retweetcount (integer), likecount (integer), quotecount (integer), hashtags (array(varchar)).
- Bottom Right:** The "Run (limit 1000)" pane contains the SQL query:

```
1 SELECT
2     sum(cardinality(hashtags)) AS num_hashtags,
3     date(cast("date" AS TIMESTAMP)) AS "date"
4 FROM
5     rankings
6 GROUP BY
7     "date";
```

The query status is "Finished" with an average read speed of 31.9K rows/s, an elapsed time of 0.37s, and 196 rows returned. The results table shows the count of hashtags per date, with three rows: 255, 300, and 268.

# SQL's workhorse: the SELECT statement

## [SELECT — Trino Documentation](#)

```
[ WITH [ RECURSIVE ] with_query [, ...] ]
SELECT [ ALL | DISTINCT ] select_expression [, ...]
[ FROM from_item [, ...] ]
[ WHERE condition ]
[ GROUP BY [ ALL | DISTINCT ] grouping_element [, ...] ]
[ HAVING condition]
[ WINDOW window_definition_list]
[ { UNION | INTERSECT | EXCEPT } [ ALL | DISTINCT ] select ]
[ ORDER BY expression [ ASC | DESC ] [, ...] ]
[ OFFSET count [ ROW | ROWS ] ]
[ LIMIT { count | ALL } ]
[ FETCH { FIRST | NEXT } [ count ] { ROW | ROWS } { ONLY | WITH TIES } ]
```

# Expected SQL knowledge

## Projecting, filtering, ordering, aggregating and joining

- Basic SELECT statement with single table with specific columns to limit the width of the data returned
- Leveraging WHERE clauses to limit the rows of data
- Sorting the results with ORDER BY
- Performing aggregations such as COUNT, MAX, SUM and AVG
  - Across the entire table population
  - For each rolled up section identified in a GROUP BY clause
- Melding two (or more) tables together with the functionality of the JOIN keyword

LINEITEM (L_)	ORDERS (O_)
SF*6,000,000	SF*1,500,000
ORDERKEY	ORDERKEY
PARTKEY	CUSTKEY
SUPPKEY	ORDERSTATUS
LINENUMBER	TOTALPRICE
QUANTITY	ORDERDATE
EXTENDEDPRICE	ORDER-PRIORITY
DISCOUNT	CLERK
TAX	SHIP-PRIORITY
RETURNFLAG	COMMENT
LINESTATUS	
SHIPDATE	
COMMITDATE	
RECEIPTDATE	
SHIPINSTRUCT	
SHIPMODE	
COMMENT	

# Instructor demonstration

## Walkthrough of Starburst Galaxy UI (10 mins)

The screenshot shows the Starburst Galaxy UI query editor interface. On the left, there's a sidebar with navigation links: Query editor, Saved queries, Query history, Catalogs, Clusters, ACCOUNT (Admin, Access control, Cloud settings), and a user account section for lester.martin@starburstdata.com.

The main area has a timestamp of 10/21/22, 3:49 PM. It displays a query editor with the following SQL code:

```
1 SELECT
2     i_color,
3     SUM(inv_quantity_on_hand) AS items_for_this_color
4 FROM
5     tpcds.tiny.inventory
6     JOIN tpcds.tiny.item ON inv_item_sk = i_item_sk
7 GROUP BY
8     i_color
9 HAVING
10    SUM(inv_quantity_on_hand) > 250000
11 ORDER BY
12    items_for_this_color DESC;
```

The status bar at the bottom indicates the query is Finished, with an average read speed of 69.6K rows/s, an elapsed time of 3s, and 81 rows. There are links for Query details, Trino UI, and Download.

The results table shows the following data:

i_color	items_for_this_color
sky	3682355
siena	3287606
thistle	3155934
papaya	26
red	64

A small icon in the bottom right corner shows a profile picture with a count of 26 notifications.

# Hands-on exercise

**Lab 1: Execute queries in Starburst Galaxy (40 mins)**

# Lesson summary

## Starburst features: Web UI

1. Starburst can be accessed through a web UI, or other client tool, and requires a username and password to access it.
2. The Starburst Galaxy navigation bar contains controls for the query, catalog, cluster, admin, access control, and cloud settings.
3. SQL is input and executed directly in the web interface.

# Starburst features

**Connectors & catalogs**

# Lesson objectives

Starburst features: Connections & catalogs

1. Explain the difference in cluster management between Starburst Galaxy and Starburst Enterprise.
2. Describe how data sources are integrated into the Starburst cluster.
3. Explain the hierarchy and difference between catalogs, schemas, and tables.
4. Describe how Starburst deploys a single point of access across a rich ecosystem of technologies.
5. Explain the role of query federation in the Starburst cluster, and how it differs from traditional query federation.

# Connectors & catalogs

Consumption models

# Is it a cluster OR does it contain clusters? YES!!

## Starburst Enterprise

Starburst Enterprise is deployed as a cluster with flexible configuration options.

Catalogs are configured for the cluster.

The web UI is bound to a specific cluster instance.

## Starburst Galaxy

A Starburst Galaxy account can have multiple clusters, each with their own sizing and auto-scaling configurations.

Catalogs are managed independently and can be linked to any number of clusters.

The web UI shows all configured clusters.

**Regardless of which consumption model is used, configuring clusters is an administrative function.**

# Starburst Enterprise – The cluster

The web UI directly shows the catalogs in the Cluster explorer.

The screenshot displays the Starburst Enterprise web interface. On the left, a sidebar includes links for 'Query editor', 'Data products', 'Domain management', 'INSIGHTS' (with 'Overview', 'Query overview', 'Cluster history', and 'Usage metrics' sub-links), and system status ('Version: 375-e', 'Environment: testertesterlab01', 'Uptime: 1h 43m'). The main area features a 'Cluster explorer' sidebar with a tree view of databases and tables, such as 'bootcamp', 'deltalake', 'hive', and 'iceberg'. A central tabbed workspace shows 'Tab 3' containing a query editor with the following code:

```
ORDER BY app_name, ip_address
SELECT count(*) FROM hive.logdemo.logs_5min_ingest_orc
SELECT *
SELECT app_name, log_level, message_id, ip_address
FROM hive.logdemo.logs_daily_part_orc
WHERE app_name IN ('CRM', 'ERP')
ORDER BY app_name, log_type, log_level, message_id, ip_address
```

The query results are displayed in a table:

app_name	log_level	message_id	ip_address
CRM	DEBUG	AAD-8414	191.101.35.191
CRM	DEBUG	AAD-8414	191.101.35.191
CRM	DEBUG	AAD-8414	191.101.35.191
CRM	DEBUG	AAD-8414	191.101.35.191
CRM	DEBUG	AAD-8414	191.101.35.191

# Starburst Galaxy – Manages clusters

The web UI includes an additional cluster hierarchy in the Cluster explorer.

The screenshot shows the Starburst Galaxy web application interface. On the left is a sidebar with navigation links: 'Query', 'Saved queries', 'Query history', 'Catalogs', 'Clusters', 'ACCOUNT' (with 'Admin', 'Access control', and 'Cloud settings' sub-options), and a user profile for 'lester.martin@starburstdata.com' (account admin). The main area has tabs for '10/21/22, 3:49 PM' (active), '10/21/22, 8:02 PM', and '11/7/22, 4:00 PM'. Below these tabs is a 'Run (limit 1000)' button and dropdowns for 'aws-us-east-1-free' and 'Select catalog'. The central part of the interface is the 'Cluster explorer' which displays a hierarchical tree of clusters and databases. The tree shows two main clusters: 'aws-us-east-1-free' and 'aws-us-east-1-sm...'. Under 'aws-us-east-1-free', there are databases like 'lakehouse', 'mysql', 'postgresql', 'sample', 'students', 'system', 'tpcds', 'tpch', and 'covid-analytics-f...'. Under 'aws-us-east-1-sm...', there are 'information...', 'serverlogs', 'logs\_5mi...', 'mysql', 'postgresql', 'sample', 'students', 'system', 'tpcds', 'tpch', and 'covid-analytics-f...'. To the right of the tree is a code editor window containing a SQL query:

```
1 SELECT
2     i_color,
3     SUM(inv_quantity_on_hand) AS items_for_this_color
4 FROM
5     tpcds.tiny.inventory
6     JOIN tpcds.tiny.item ON (inv_item_sk = i_item_sk)
7 GROUP BY
8     i_color
9 HAVING
10    SUM(inv_quantity_on_hand) > 250000
11 ORDER BY
12    items_for_this_color DESC;
13
14
15
16
17
18 CREATE SCHEMA
19 students.instructor;
20
21
22 CREATE TABLE
23 students.instructor.nation AS
24 SELECT
25     *
26 FROM
27     tpch.sf1.nation;
```



# Starburst Galaxy – Cluster list

The web UI allows existing clusters to be managed and new ones to be created.

The screenshot shows the Starburst Galaxy web interface. The top navigation bar includes the logo, a search icon, user information for 'lester.martin@starburstdata.com accountadmin', and a dropdown menu. The left sidebar has sections for 'Query' (Query editor, Saved queries, Query history), 'Catalogs', and 'Clusters' (which is selected and highlighted in grey). The 'ACCOUNT' section includes Admin, Access control, and Cloud settings. The main content area is titled 'Cluster' and 'View clusters'. It contains a brief description: 'A cluster in Starburst Galaxy provides the resources to run queries against numerous catalogs. You can access the data exposed by the catalogs with the query editor or other clients.' Below this is a 'Create cluster' button. A table lists three clusters: 'aws-us-east-1-f...' (Running, Stop), 'aws-us-east-1-s...' (Suspended, Resume), and 'covid-analystics...' (Stopped, Start). The table columns are Name, Status, Quick actions, Catalogs, and Size.

Name ↑	Status	Quick actions	Catalogs	Size
aws-us-east-1-f...	Running	Stop	lakehouse, mys...	Free
aws-us-east-1-s...	Suspended	Resume	lakehouse, mys...	Small
covid-analystics...	Stopped	Start	covid	Free

# Connectors & catalogs

Data source options

# Rich ecosystem of data source connectors

- Live access to 50+ modern and legacy enterprise data sources
- Federate access to multiple sources without moving or copying data
- Combine external data with data that cannot move
- Join data stored in different formats - relational, non-relational, structured, unstructured, streaming, object stores
- Integrate with existing security solutions
- Deliver data warehouse functionality to the data lake
- Continue to use the tools you know and love while getting more value out of your data

ORACLE ICEBERG Google Cloud

snowflake CLOUDERA teradata.

50+  
supported  
connectors



# Rich ecosystem of data source connectors

*Open-source & Starburst Proprietary*

Data Source  
Connectors

Real-time Analytics



Data Lakes



NoSQL Stores



Applications



Relational DBs



# **Connectors & catalogs**

Putting it all together

# Connecting to data sources

## Catalog

A catalog is an instance of a data source connector.

Multiple catalogs can use the same underlying connector with different configuration settings.

## Schema

A way to organize tables.

Analogous to the organization concepts in popular RDBMS tools.

## Table

A set of unordered rows, which are organized into named columns with data types.

Includes views.

**Three-part name for a table's unique identifier: `catalog.schema.table`**

# Multiple ways to reference tables & views

**Cluster explorer**

- aws-us-east-1-free
  - lakehouse
  - mysql
  - postgresql
  - sample
  - students**
    - information\_schema
    - instructor**
      - customer
      - nation**
        - nationkey
        - name
        - regionkey
        - comment
    - markmorrissey
    - yourname
  - system
  - tpcds
  - tpch
- aws-us-east-1-small
- covid-analystics-free

▶ Run selected (limit 1000) ◀

aws-us-east-1-free students Select schema ⋮

```
1 SHOW CATALOGS;
2
3 SHOW SCHEMAS FROM students;
4
5 SHOW TABLES FROM students.instructor;
6
7 DESCRIBE students.instructor.nation;
```

information\_schema  
instructor  
markmorrissey  
yourname

Finished Avg. read speed 4.7 rows/s Elapsed time 0.85s Rows 4

Query details Trino UI Download

Column	Type	Extra	Comment
nationkey	bigint		
name	varchar(25)		
regionkey	bigint		
comment	varchar(152)		

# What can we do with multiple catalogs?

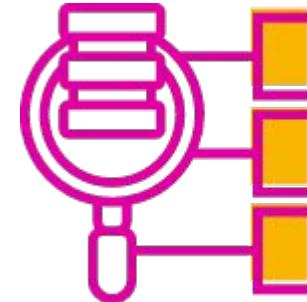
## Single point of access for a variety of data sources

- Same UI/CLI/API for access all of your data
- Configure security and governance in one place

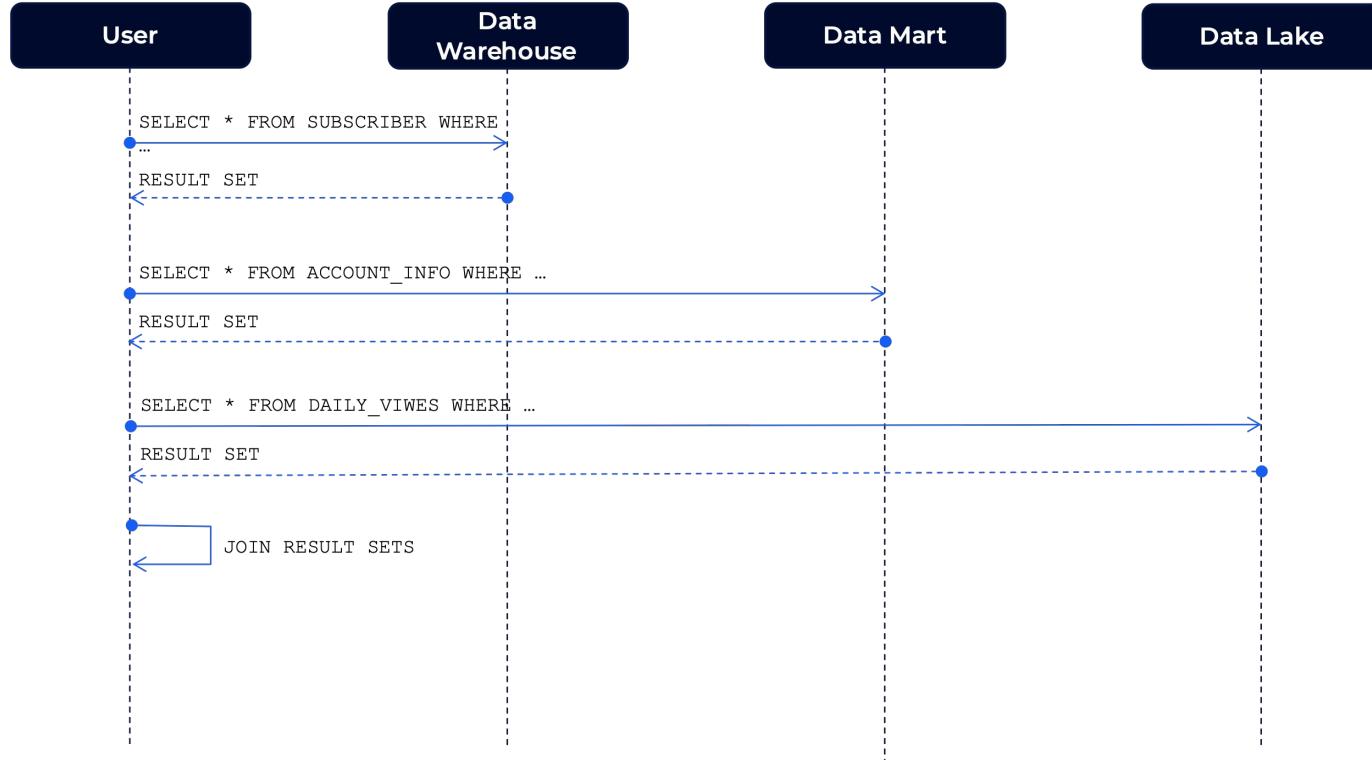


## Query federation

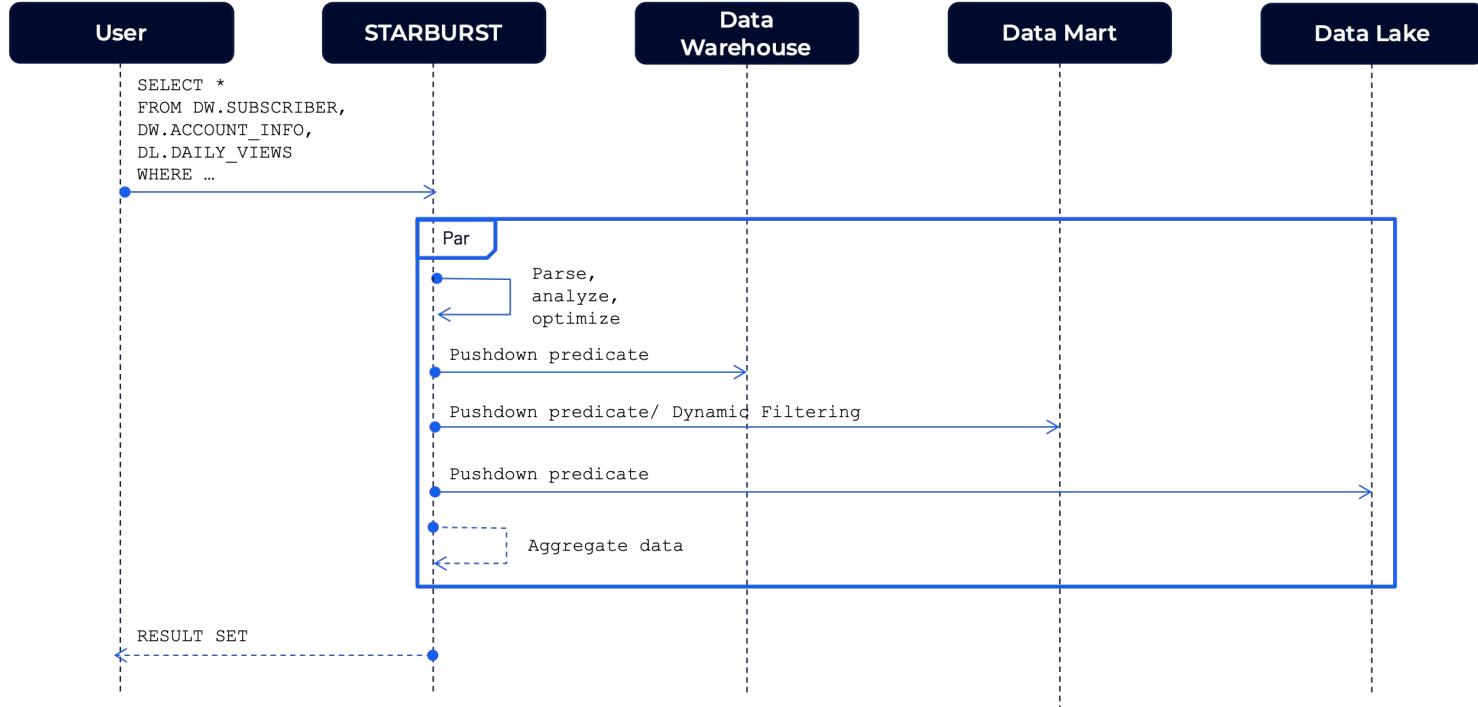
- Access data from multiple systems within a single query
  - For example, join historic log data stored in an S3 object storage with customer data stored in a MySQL database



# Query federation the traditional way



# Query federation the starburst way



# Instructor demonstration

Create a catalog and a federated query (10 mins)

## Select a data source

Each catalog contains configuration for Starburst Galaxy to access a data source. Configure catalogs and use them in clusters to query data sources in Starburst Galaxy.

Note: Amazon S3, Azure Data Lake Storage, and Google Cloud Storage catalogs support Iceberg, Hive, Delta Lake, and Hudi (Preview) tables.



Amazon S3



Azure Data Lake Storage



Google Cloud Storage



Microsoft SQL Server



MySQL



PostgreSQL

# Hands-on exercise

**Lab 2: Exploring federated queries (15 mins)**

# Lesson summary

## Starburst features: Connectors & catalogs

1. Starburst Galaxy deploys multiple, auto-scaling clusters, whereas Starburst Enterprise deploys a single, configurable cluster.
2. Starburst includes 50+ integrations, allowing for connections to multiple data sources. These include relational and non-relational systems.
3. Tables hold row and columns of data; schemas organize tables; catalogs contain schemas.
4. Starburst allows you to connect to any data source from a single point of access.
5. Starburst aggregates the complexity of query federation, allowing a single SQL statement to join across multiple data sources.

# Starburst features

**Client tools integration**

# Lesson objectives

Starburst features: Client tools integration

1. Understand how client tools integrate with Starburst, and which tools can be used.
2. Explain the impact that client tool integrations like Tableau and DBeaver have for data producers and data consumers.
3. Explain how Starburst Galaxy integrates with both JDBC and ODBC connectors.
4. Describe how Starburst products can be accessed via the Python client.

# Connect with client tools

## Data Engineers

- [CLI](#), [API](#) and [REST](#) access
- SQL clients
  - DBeaver
  - dbVisualizer
  - etc



**DBeaver**

## Data Analysts

- Business Intelligence tools
  - Tableau
  - Power BI
  - etc



## Data Scientists

- Web-based notebooks
  - Jupyter
  - IPython
  - Zeppelin
  - etc



# Connect via DBeaver



DBeaver

	event_time	rbc_ip_address	rbc_app_name	process_id	rbc_log_type	rbc_log_
1	2021-07-25 01:13:43.000	177.101.41.177	App95	1,776	EVENT	TRACE
2	2021-07-25 01:13:43.000	199.101.31.199	HRIS	4,552	EVENT	WARN
3	2021-07-25 01:13:43.000	199.101.41.199	WebLogic	3,302	EVENT	WARN
4	2021-07-25 01:13:43.000	217.101.31.217	App97	4,728	EVENT	WARN
5	2021-07-25 01:13:43.000	175.101.45.175	App05	527	EVENT	INFO
6	2021-07-25 01:13:43.000	185.101.27.185	PostgreSQL	3,648	EVENT	INFO
7	2021-07-25 01:13:43.000	169.101.27.169	App96	2,018	EVENT	INFO
8	2021-07-25 01:13:43.000	115.101.33.115	ERP	5,177	EVENT	DEBUG
9	2021-07-25 01:13:43.000	217.101.33.217	App92	5,820	AUDIT	INFO
10	2021-07-25 01:13:43.000	183.101.31.183	App01	1,820	EVENT	TRACE
11	2021-07-25 01:13:43.000	203.101.47.203	HRIS	4,059	EVENT	TRACE
12	2021-07-25 01:13:43.000	191.101.47.191	HRIS	2,978	EVENT	WARN
13	2021-07-25 01:13:43.000	113.101.41.113	App95	5,047	AUDIT	WARN
14	2021-07-25 01:13:43.000	203.101.45.203	App92	2,409	EVENT	INFO
15	2021-07-25 01:13:43.000	125.101.29.125	Apache	2,102	REQUEST	TRACE
16	2021-07-25 01:13:43.000	251.101.27.251	Apache	1,956	AVAILABILITY	TRACE
17	2021-07-25 01:13:43.000	119.101.31.119	Apache	5,578	THREAT	INFO
18	2021-07-25 01:13:43.000	203.101.29.203	App96	3,318	EVENT	INFO
19	2021-07-25 01:13:43.000	185.101.31.185	App91	5,362	EVENT	WARN
20	2021-07-25 01:13:43.000	135.101.43.135	DB2	4,148	EVENT	TRACE

Artifact: Connecting DBeaver to Starburst Enterprise (YouTube)

# Connect via Tableau

The screenshot shows the Tableau Data Catalog interface. On the left, the navigation pane includes 'Connections' (ab1fd59d8f7a...amazonaws.com, Starburst Enterprise by Starburst), 'Catalog' (hive), 'Schema' (logdemo), and 'Table' (logs\_5min\_ingest\_csv, logs\_5min\_ingest\_csv\_stats, logs\_5min\_ingest\_orc, logs\_daily\_part\_orc, logs\_daily\_rollup\_orc, New Custom SQL). The main area displays a connection named 'logs\_daily\_part\_orc (logdemo)' with a status of 'Live'. A table named 'logs\_daily\_part\_or' is selected. A small icon of two overlapping tables is shown above the table name. Below the table, there's a message 'Need more data?' and a note 'Drag tables here to relate them.' followed by a link 'Learn more'. At the bottom, a preview of the table structure is shown with columns: Name, Event Time, and Ip Address.



Artifact: Using Starburst Galaxy to Query in Tableau (YouTube)

# Connect via Jupyter (via Python client)

A screenshot of a Jupyter Notebook interface. At the top, there's a toolbar with various icons. To the right of the toolbar, it says "Kernel" and "Not Trusted" with a dropdown menu showing "conda\_python3". On the far right, there are "Logout" and other user-related options. The main area shows a notebook titled "jupyter Starburst BurstBank 1" with a last checkpoint at 3 hours ago (autosaved). The notebook has tabs for "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". Below the toolbar, there's a "Toolbar" section with icons for file operations like new, open, save, run, etc. The main content area contains a code cell labeled "In [1]:" which contains Python code for data exploration using Trino and Pandas. The code includes imports for pandas, numpy, trino, seaborn, matplotlib.pyplot, sqlalchemy, and sqlalchemy.create\_engine. It also includes warnings handling. Above the code cell, a title "# Burst Bank data exploration Using Python" is followed by a subtitle "In this demo, we will show functionality accessing enterprise data from a Starburst Cluster with data on Hive, Postgresql, SQL Server, among others." To the right of the code cell, there are two buttons: "Markdown Cell" and "Code Cell".

```
import pandas as pd
import numpy as np
import trino
import seaborn as sns
from matplotlib import pyplot as plt
from sqlalchemy import *
from sqlalchemy.engine import create_engine
from sqlalchemy.schema import *
# import ipython-sql

import warnings
warnings.filterwarnings("ignore")
warnings.simplefilter("ignore", category=PendingDeprecationWarning)
```

Artifact: [trinodb / trino-python-client \(GitHub\)](#)

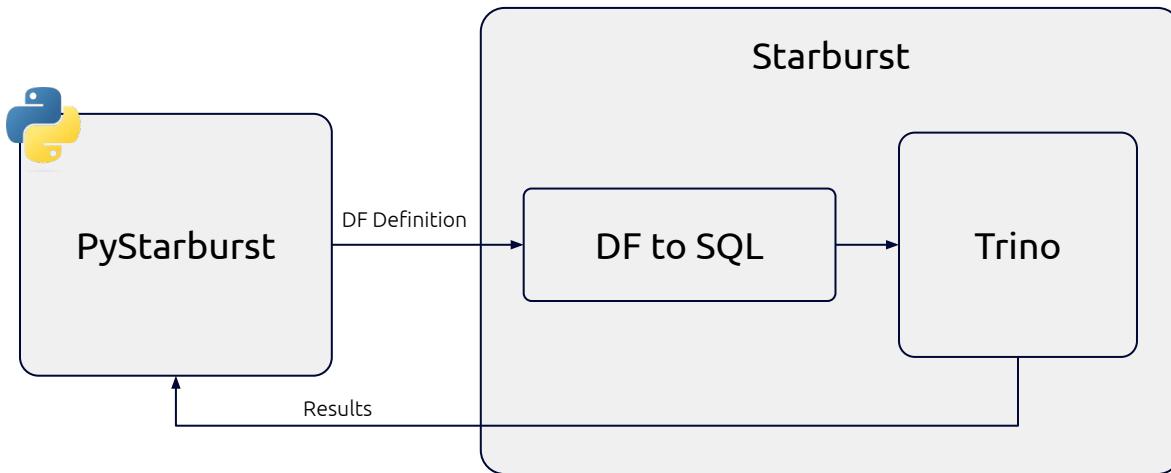
# PyStarburst Overview [Data Engineers/Scientists]

```
df_missions = df_missions.with_column("date", f.sql_expr("COALESCE(TRY(date_parse(\"date\"), '%a %b %d, %Y %H:%i UTC')), NULL)"))

print(df_missions.schema)

df_missions = df_missions\
    .filter(col("date") > datetime(2000, 1, 1))\
    .sort(col("date"), ascending=True)

df_missions.show()
```



- PySpark-like Syntax
- Lazy execution
- Python gets converted to SQL
- Heavy lifting done by Trino

Links: [API Docs](#) & [Example Code](#)

Currently supported on Starburst Galaxy and Starburst Enterprise.

# Lesson summary

## Starburst features: Client tools integrations

1. Starburst Galaxy integrates with a number of client tools, including DBeaver, Tableau, and Jupyter.
2. Client tools connect via the CLI or one of the supported APIs.
3. Data producers use these tools to support ETL pipelines. Data consumers, including data scientists and analysts, use them to achieve business insights.
4. Starburst makes full use of JDBC and ODBC connectors, both of which are versatile and adaptable.
5. Starburst can be accessed via Python clients, often through web based notebooks.