Starburst

Starburst 101

Aug 20, 2025

Connection before content

Lester Martin – https://linktr.ee/lestermartin

- Developer Relations @ Starburst
  - Blogging & forums
  - Webinars & videos
  - User groups & events
  - Training & tutorials
- 30+ years of technology experience
  - Started journey on TRS-80 Model III
  - Played most roles, but a programmer at my core
  - ½ career in OLTP and ½ in data analytics
  - Decade+ of "big data" experience to include
    - Trino/Starburst, Hadoop, Hive, Spark
    - NiFi, Kafka, Storm, Flink
    - HBase, MongoDB

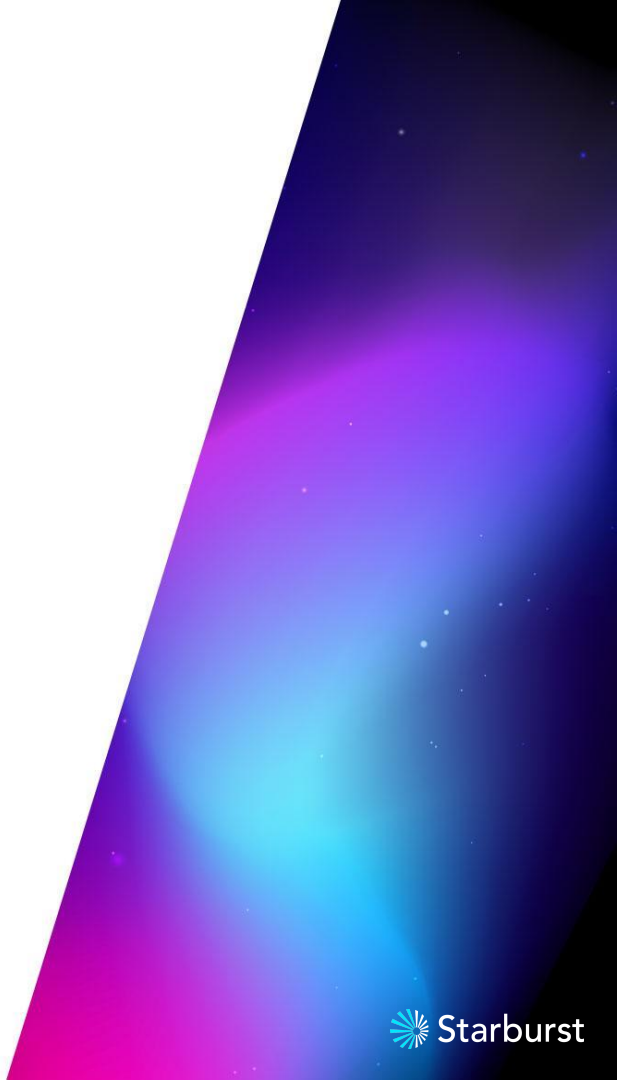
devrel@starburst.io

Starburst

Agenda and objectives

Part 1: Introduction to Trino and Starburst

Part 2: Starburst demo



Starburst

Part 1 agenda

- Getting to know Trino & Starburst
- The Query Engine & Architecture
- Connectors & Catalogs
- The Open Data Lakehouse
- Data Products

Starburst

Getting to know Trino & Starburst

History of Trino (before)

Querying large volumes of data was difficult and time consuming

Early 2000s: Data generation and collection has skyrocketed due to the rise of the Internet

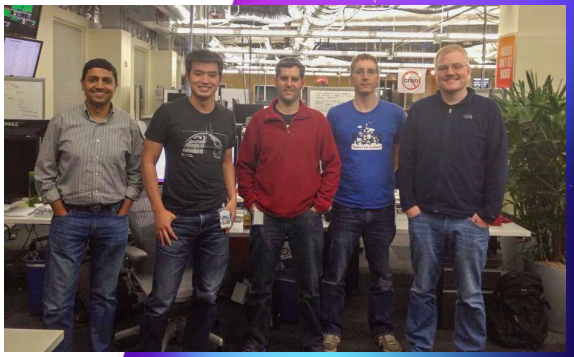
2006: Hadoop was designed to meet the needs of large datasets on a scale previously unimaginable

2008: Facebook created Hive to query terabytes of data in Hadoop using a SQL-like interface. Data consumers were limited by the number of queries they could run — often fewer than 10 in one day

History of Trino

A new query engine designed to solve the data accessibility problem

2012: Trino (*formerly known as Presto)* is created by Martin Traverso, Dain Sundstrom, David Phillips and Eric Hwang at Facebook



Trino is an open source query engine that:

- *Harnesses the power of distributed computing*
- *Separates compute from storage*
- *Super fast and performant*
- *ANSI-SQL BASED!!!!! Which means… SQL on anything!*

History of Trino - ETL Processing

From purely interactive use-cases to multiple workloads

2013: Released into production at Facebook for interactive use cases

2014: Users start scheduling batch/ETL queries with Trino instead of Hive

2018: 50% of existing ETL workloads and 85% of new workloads on Trino

Why?

- *Trino can communicate with disparate data sources to federate data*
- *Trino is a distributed, massively parallel processing system*
- *Faster, Cheaper and ANSI-SQL BASED!*

Soon others caught on, and teams like Salesforce and Lyft started utilizing Trino for Batch/ETL capabilities.

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

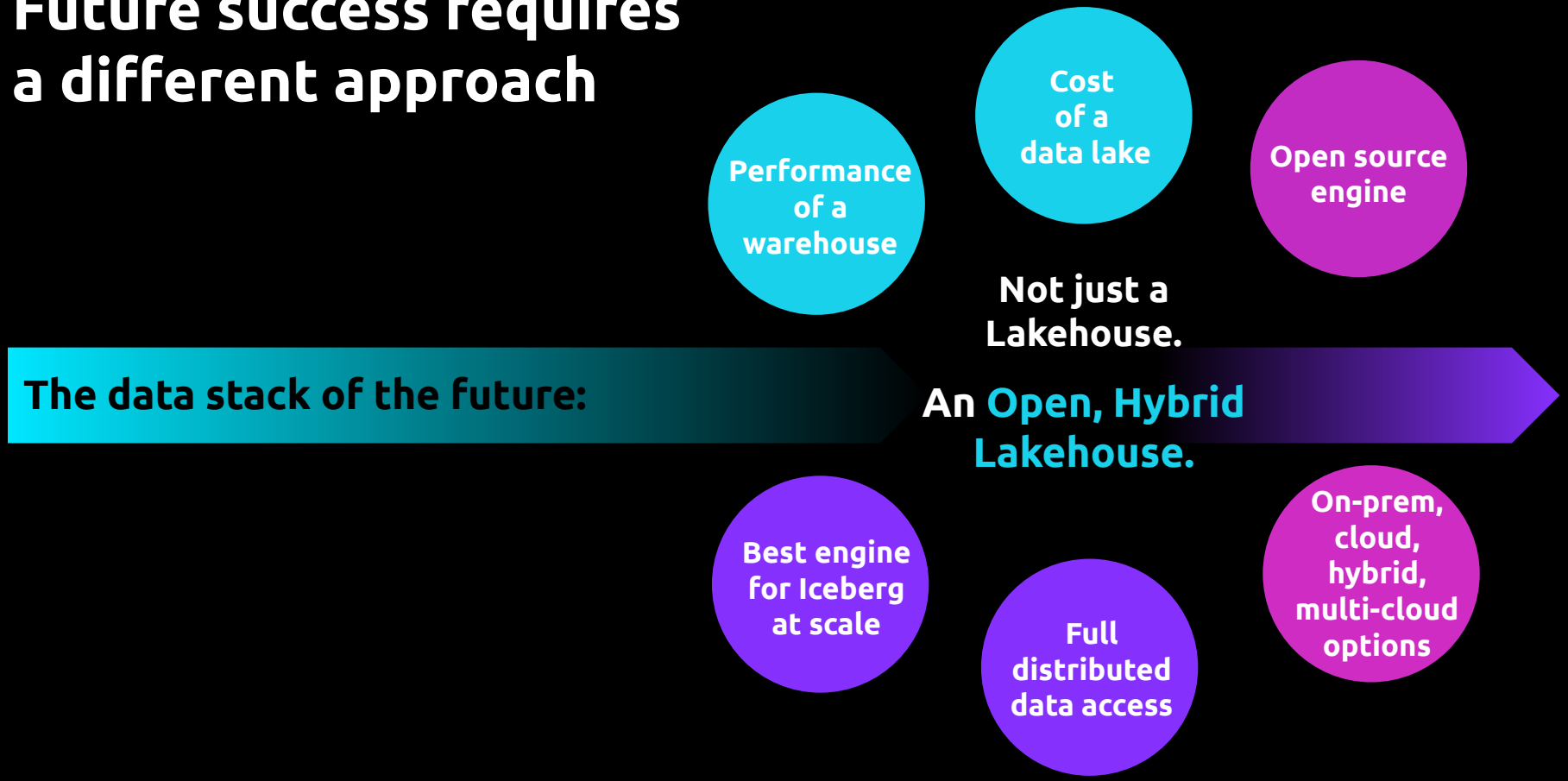
$$$
(and time!)

Starburst

What we're seeing: data stacks can't keep up

Trends	Pains
Exploding data volume, velocity, variety	Data stacks struggle to meet demand
AI/ML accelerating data demands	Costs out of control, ROI pressure
Self-serve and democratization	Growing security, compliance risks
Hybrid/multi-cloud the norm	Vendor lock-in limits agility, innovation
Growing use of open-source (ex: Iceberg)	Open source difficult to self-support

Future success requires a different approach



Starburst

Open, Hybrid Lakehouse requires an open engine

- Open-source query engine.
- Separates compute and storage.
- Queries across all data sources.
- Iceberg was designed for Trino.

Proven at exabyte scale/high concurrency:

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

Trino open source users

Starburst is the Trino company:

- **Bringing Trino to the enterprise**
- **Cofounded by Trino creators**
- **#1 Trino committer**
- **Largest team of Trino experts in the world**

Thriving open source community:

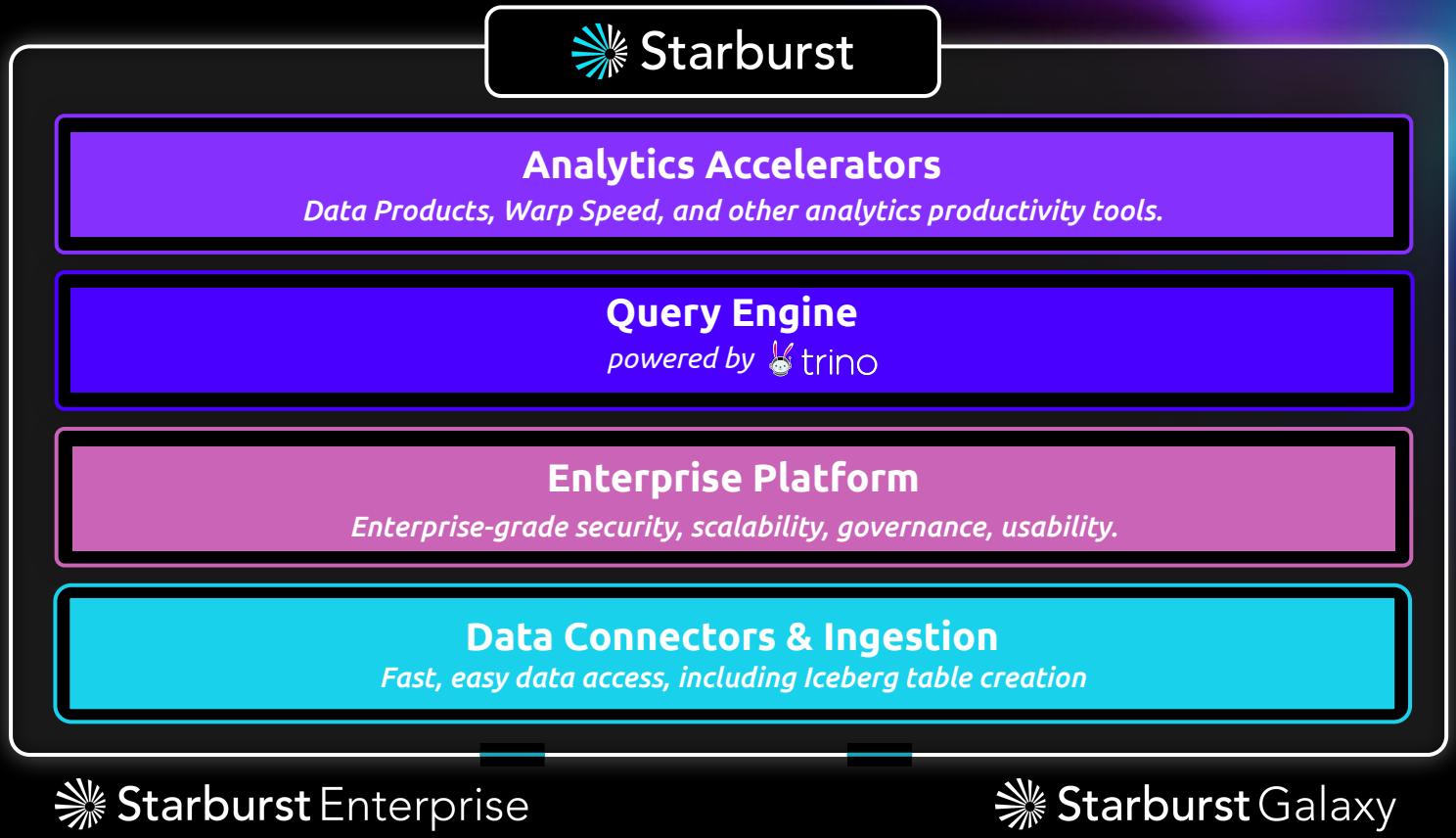
11300+
SLACK MEMBERS

10,000+
GITHUB STARS

750+
CONTRIBUTORS

Starburst

Starburst is an Open, Hybrid Lakehouse platform



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

HSBC · citibank · Volkswagen · sanofi · GILEAD · vmware · BRITISH AIRWAYS · COMCAST

OCBC Bank · BARCLAYS · Carrefour · emis health · Optum · SS&C · Unity · orange™

SOCIETE GENERALE · DBS · zalando · align · NOVANT HEALTH · zoominfo · sky · verizon

Starburst

The Query Engine & Architecture

Lesson objectives

Starburst features: Architecture

1. Understand the role that workers and coordinators play in Starburst clusters.
2. Analyze a typical Starburst execution flow in detail.
3. Differentiate Starburst from Trino.

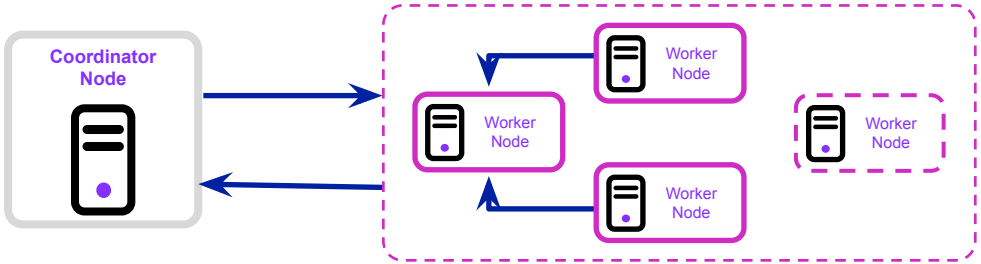
Server stereotypes

Coordinator node

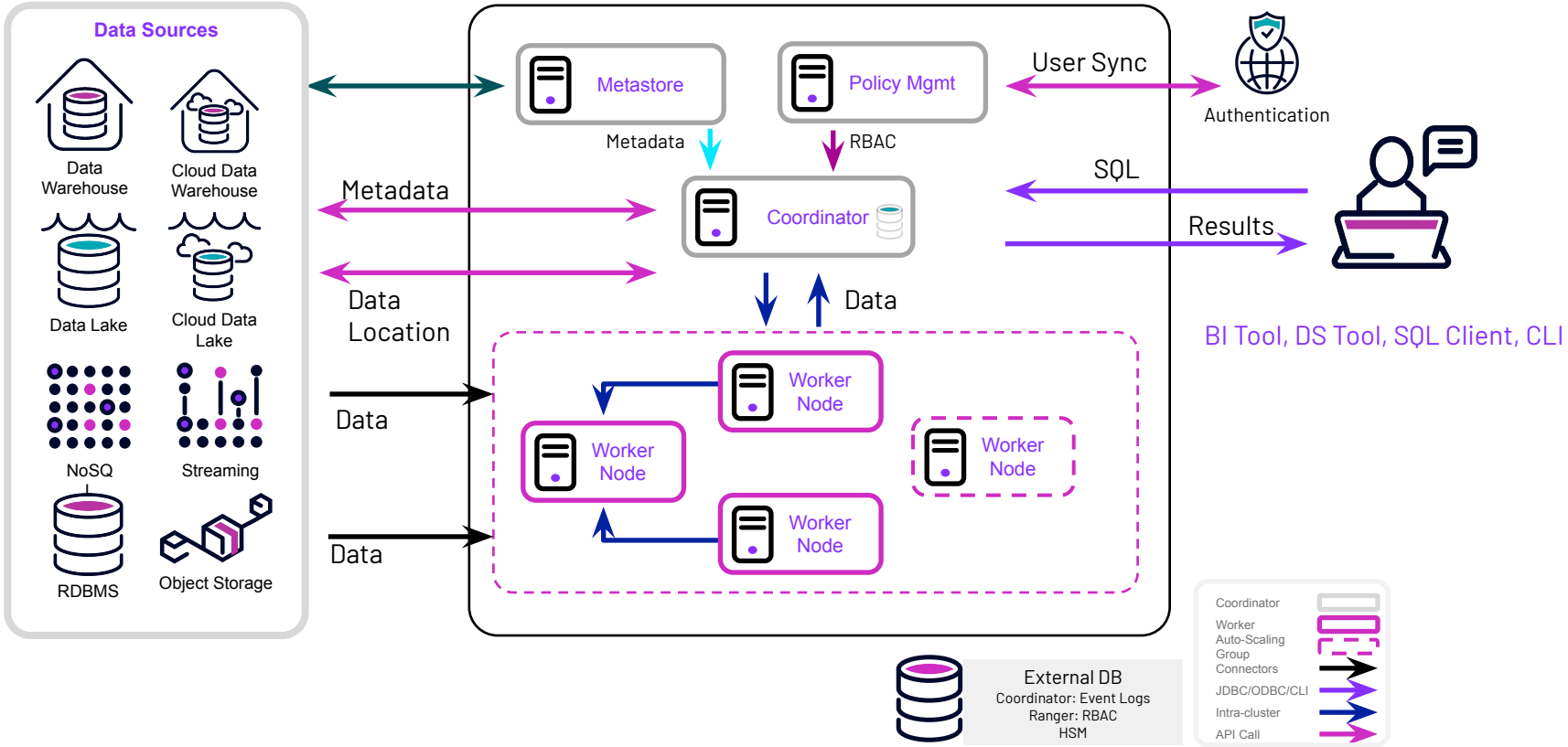
Server that is responsible for parsing statements, planning queries, and managing Trino worker nodes.

Worker nodes

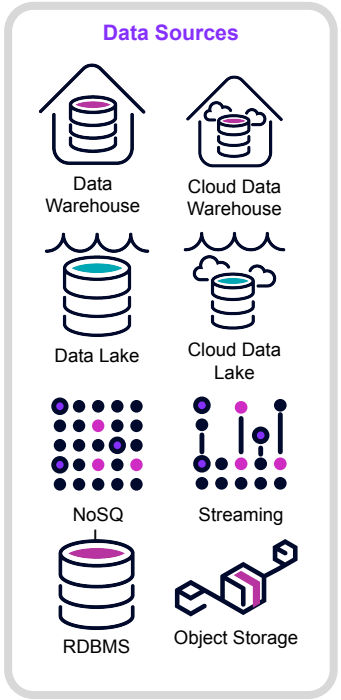
Server which is responsible for executing tasks and processing data. Worker nodes fetch data from connectors and exchange intermediate data with each other.

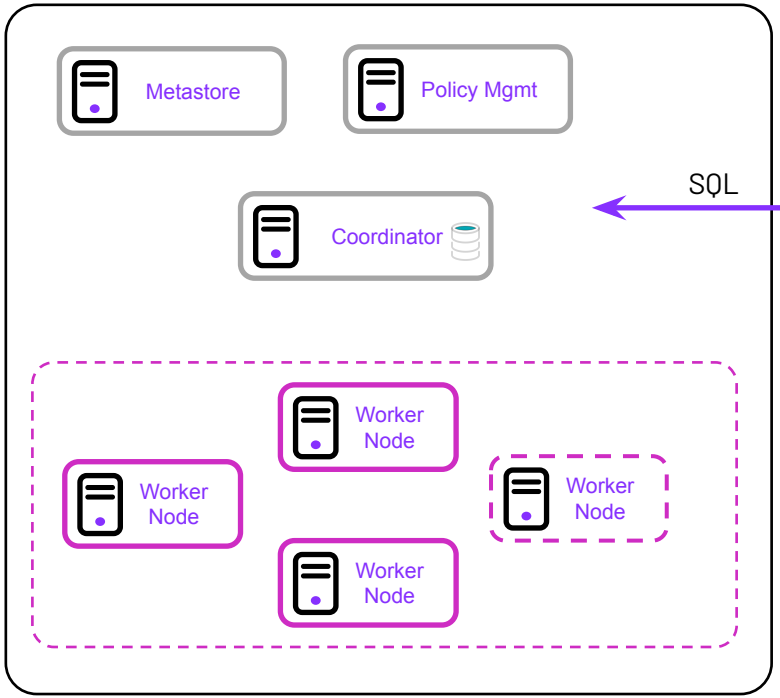


Starburst logical architecture

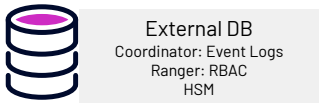


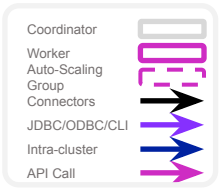
Query step: Statement submitted



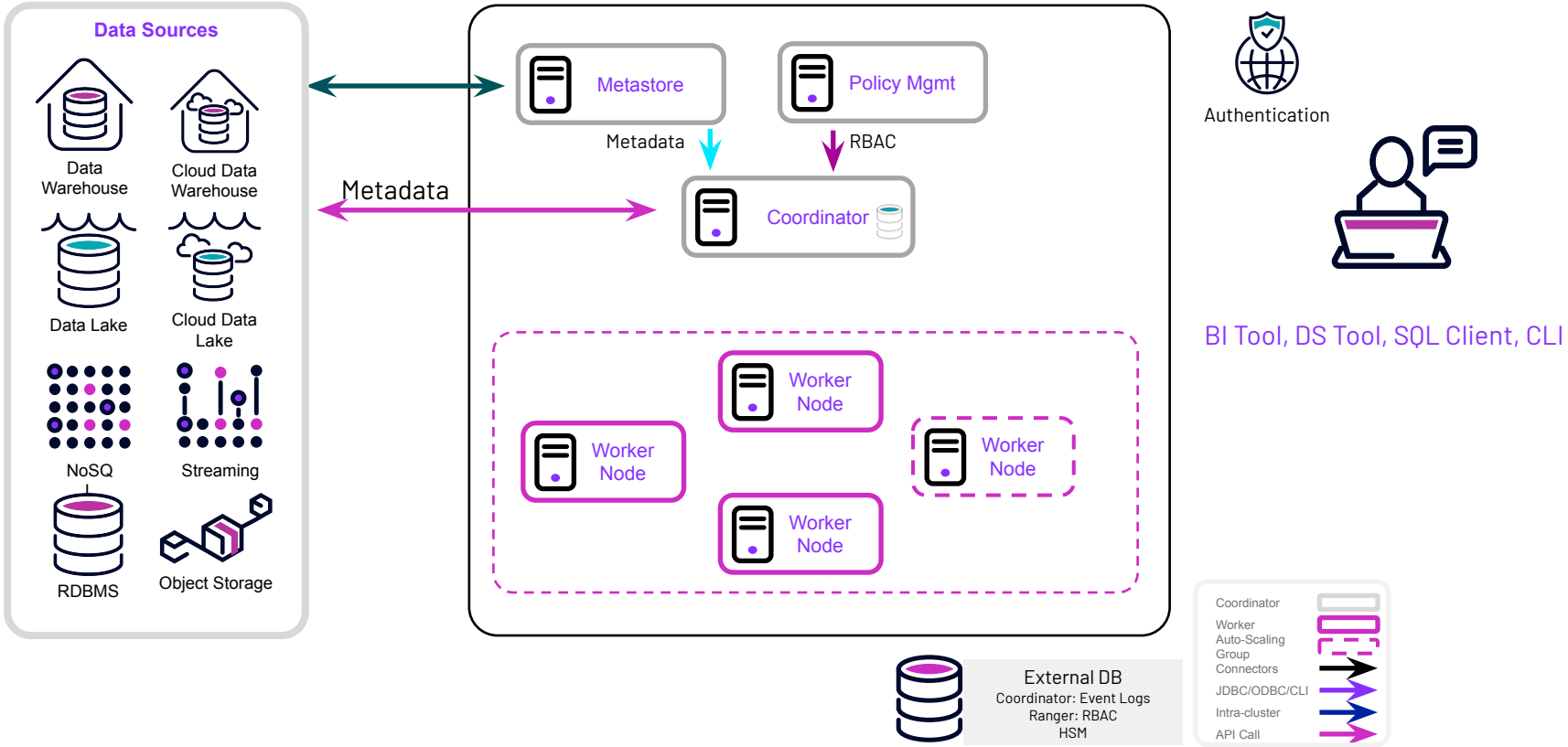


BI Tool, DS Tool, SQL Client, CLI

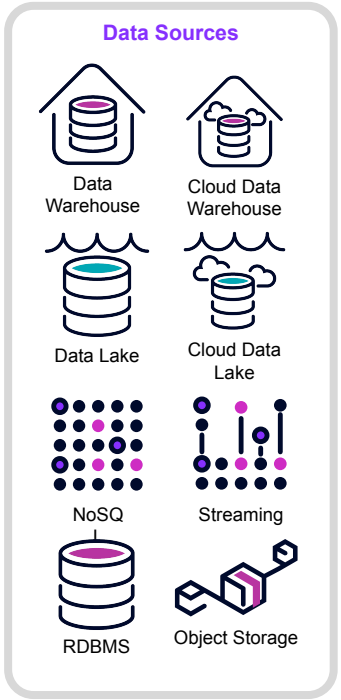


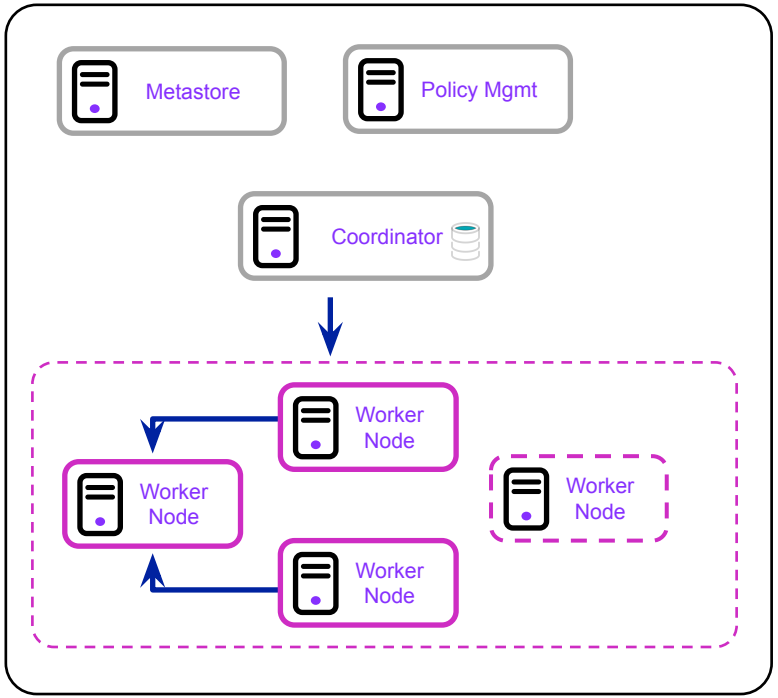


Query step: Query validated

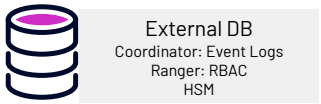


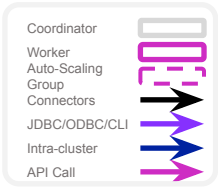
Query step: Tasks assigned



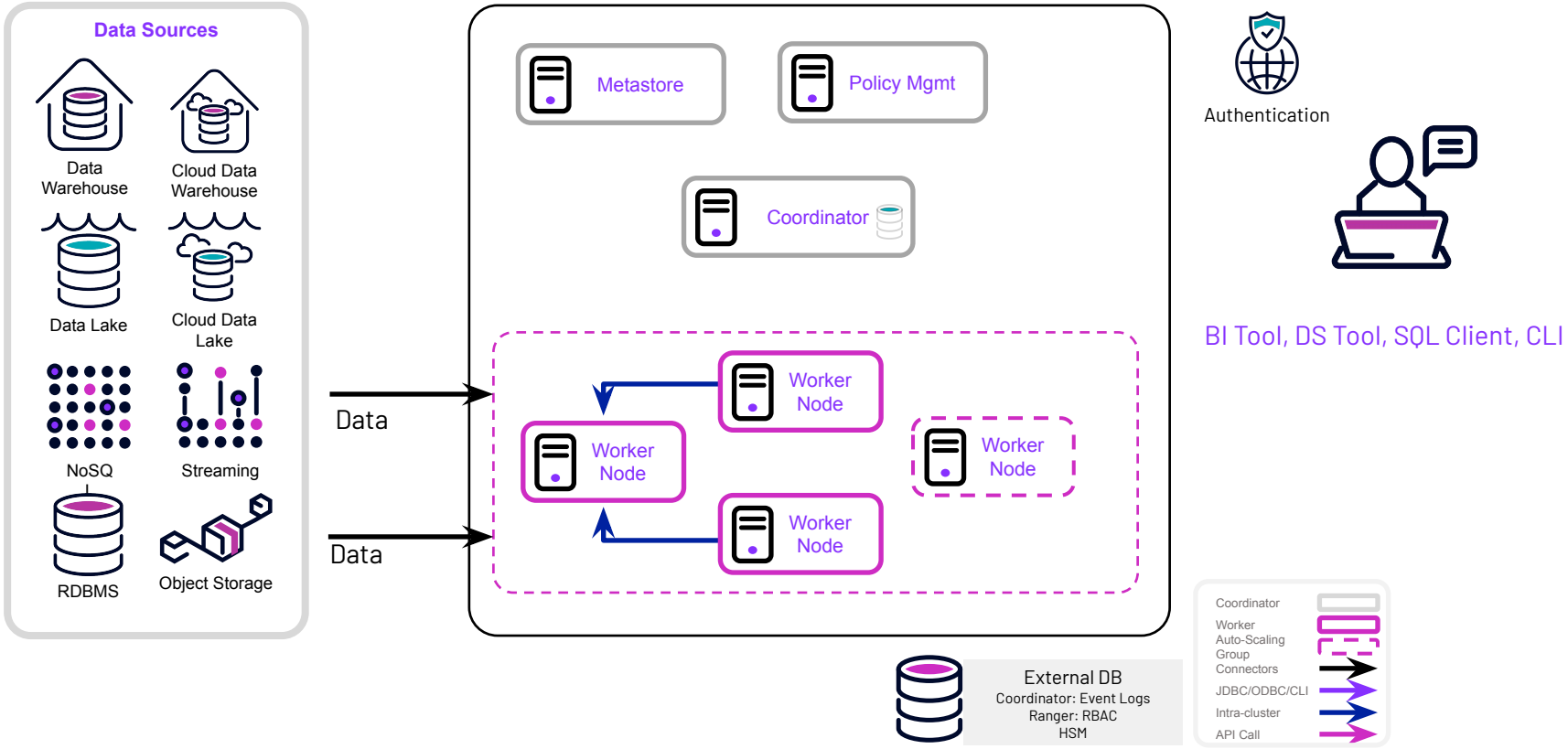


BI Tool, DS Tool, SQL Client, CLI

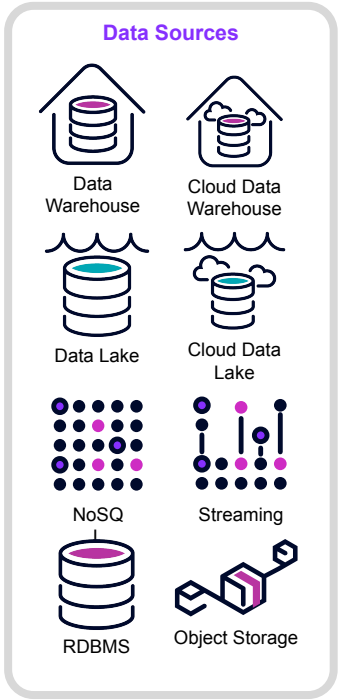


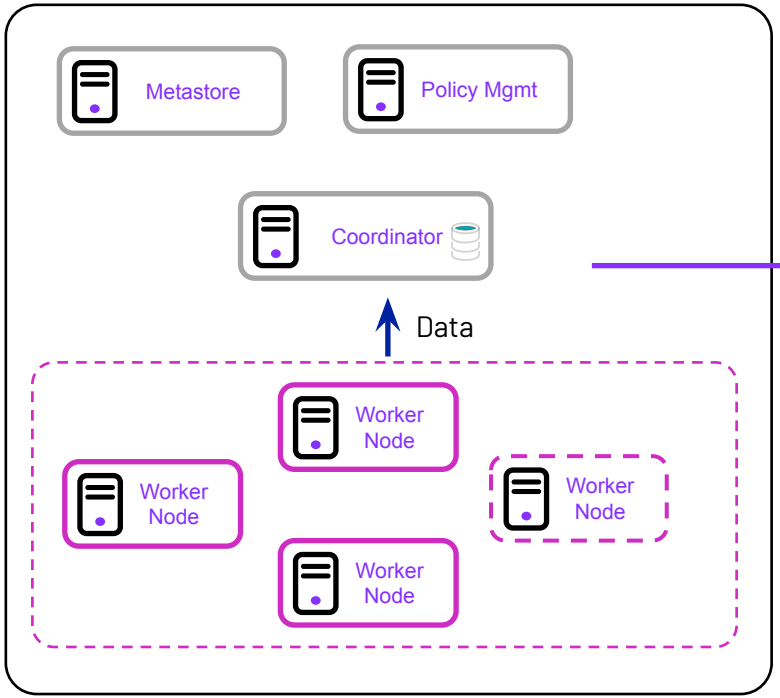


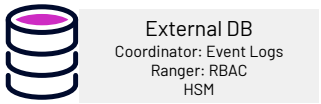
Query step: Tasks executed

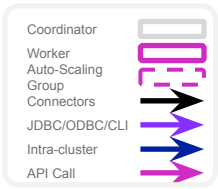


Query step: Results returned

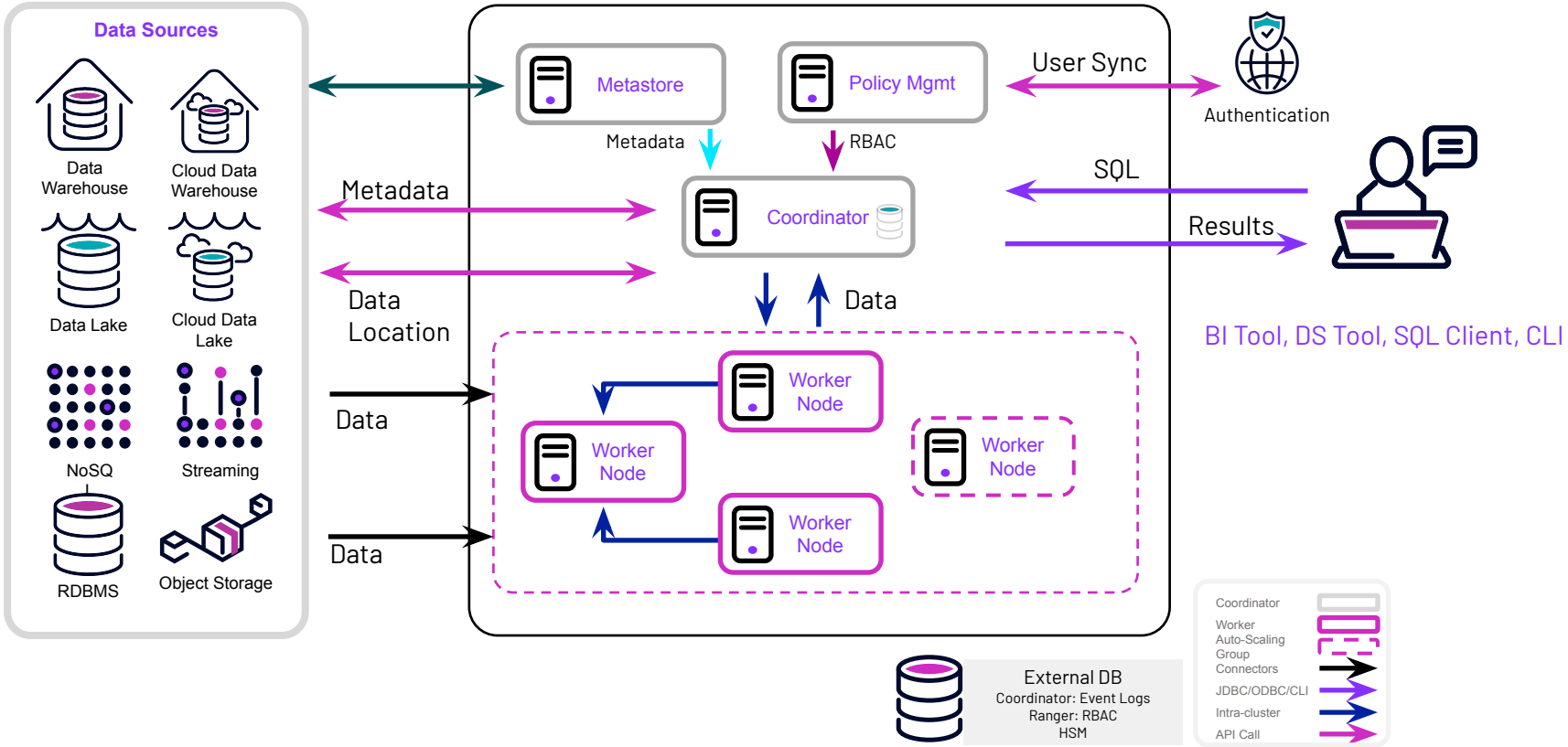








Starburst logical architecture



Starburst

Connectors & Catalogs

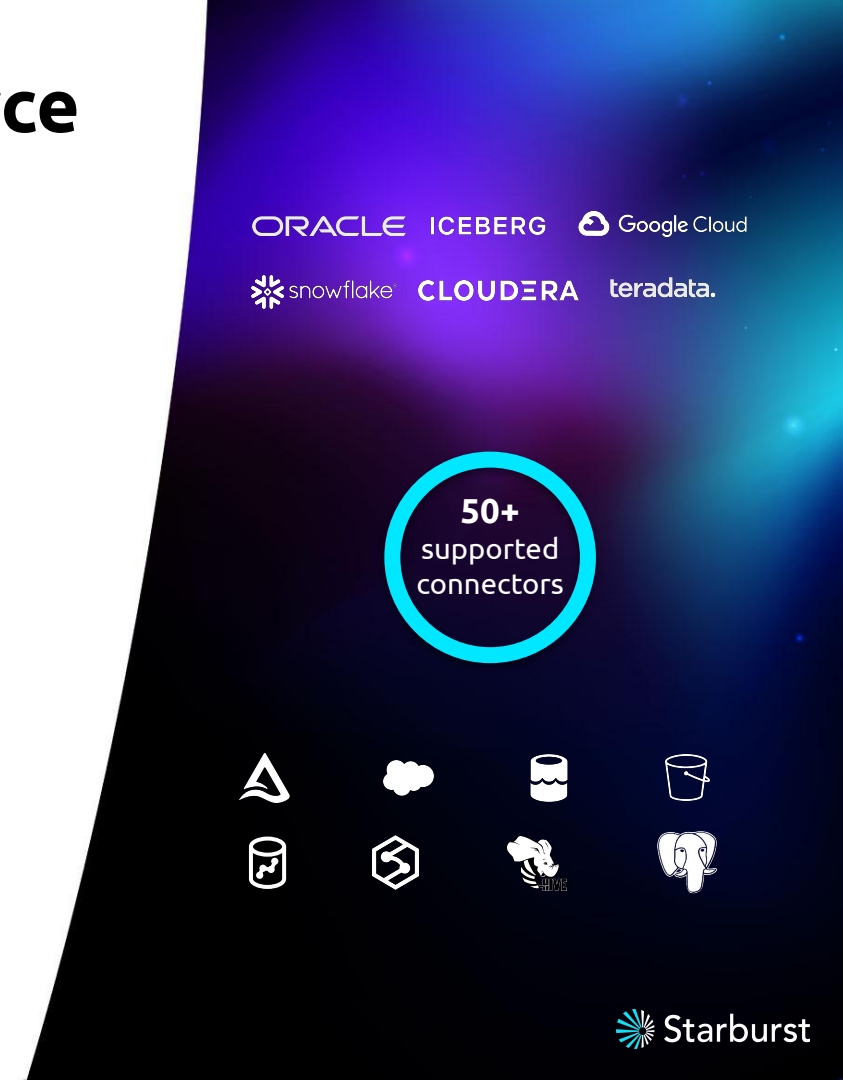
Lesson objectives

Starburst features: Connections & catalogs

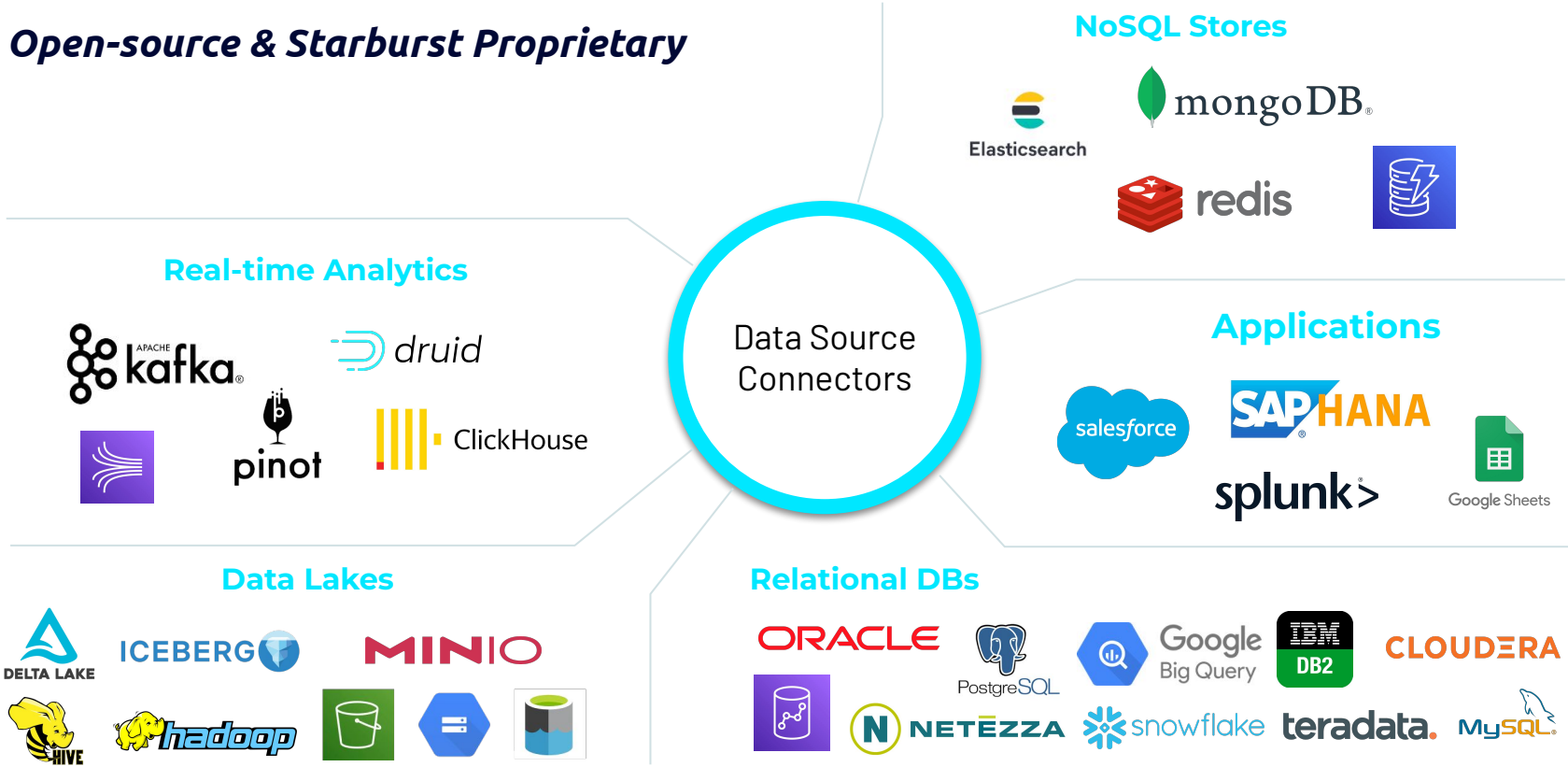
1. Describe how data sources are integrated into the Starburst cluster.
2. Explain the hierarchy and difference between catalogs, schemas, and tables.
3. Describe how Starburst deploys a single point of access across a rich ecosystem of technologies.
4. Explain the role of query federation in the Starburst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

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



Rich ecosystem of data source connectors



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.

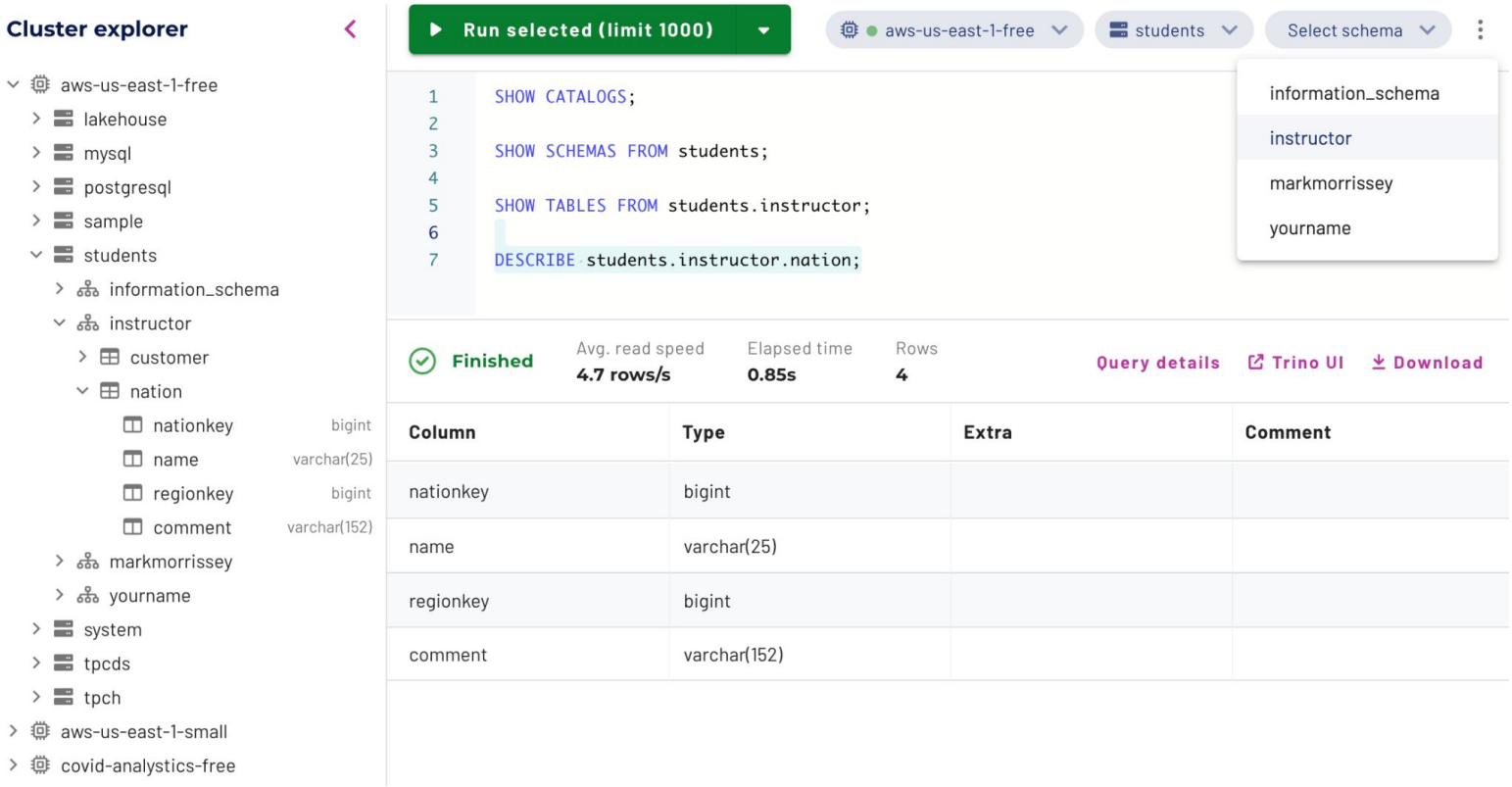
Table

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



Starburst

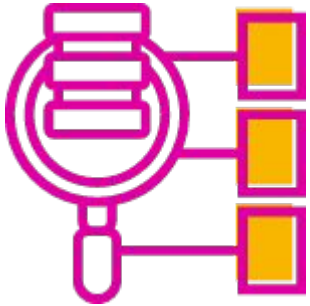
What can we do with multiple catalogs?

Single point of access for a variety of data sources

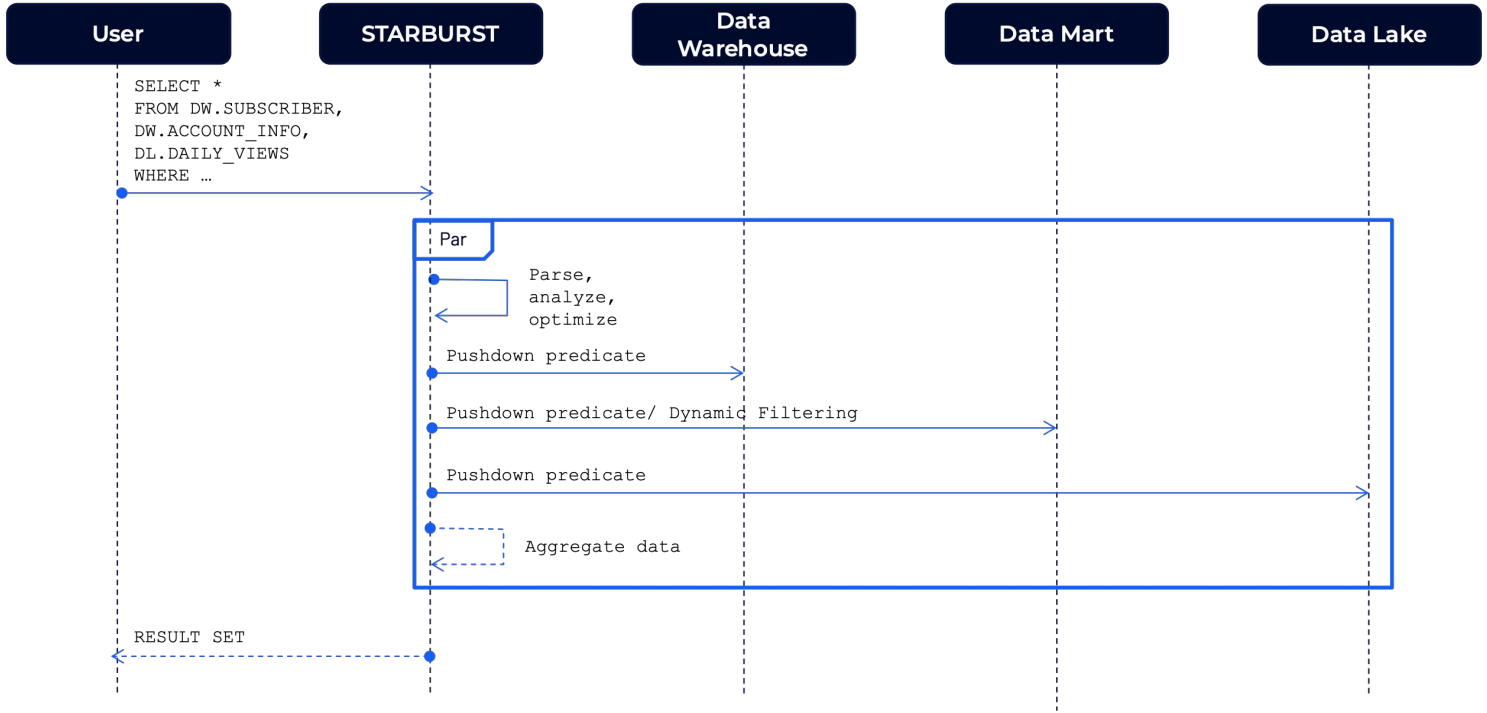
- Same UI/CLI/API for access all of your data
- Configure security and governance in one place

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 with Starburst



Starburst

Starburst

The Open Data Lakehouse

Lesson objectives

Trino, Iceberg, and the Open Data Lakehouse

1. Explain the data lake and data lakehouse.
2. Introduce Apache Iceberg.
3. Discuss the power of the open data lakehouse.

The Data Lake

Born out of the internet age and big data boom

The Good

- In 2006, Apache Hadoop emerges so unstructured data can be processed at a scale previously imaginable
- Shift toward parallel processing
- Capitalize on low cost object storage
- Allows for greater flexibility (schema on read)

The Bad

- Inability to support transactions, updates, or modifications
- Difficult to get top tier performance
- Lack of data quality and inconsistent data formats
- Insufficient data lineage and limited data discoverability

The Data Lakehouse

Upleveling your data lake to the next level

- Utilize the ***separation of storage and compute*** to apply the reliability, performance, data quality of the data warehouse to the openness and scalability of the data lake
- ***Increased performance and scalability*** through the use of indexing and caching via your query engine and modern table formats
- Tackle ***unstructured, semi-structured, and structured*** analytical data all in a data lakehouse - creating a place for AI/ML & BI use cases alike

Apache Iceberg

Applying data warehouse principles to the data lake

- Created by Ryan Blue & Daniel Weeks at **Netflix** in 2017.
- Solve the challenges of performance, data modification and schema evolution in the lake.
- Uses open data concepts (orc, parquet, avro) and architecture.
- Seen enormous interest and adoption over the last 3 years.
- Applies SQL behavior like hidden partitioning and schema evolution the the lake also offering modern warehouse SQL such as MERGE, UPDATE, DELETE, and Time Travel.

The Open Data Lakehouse

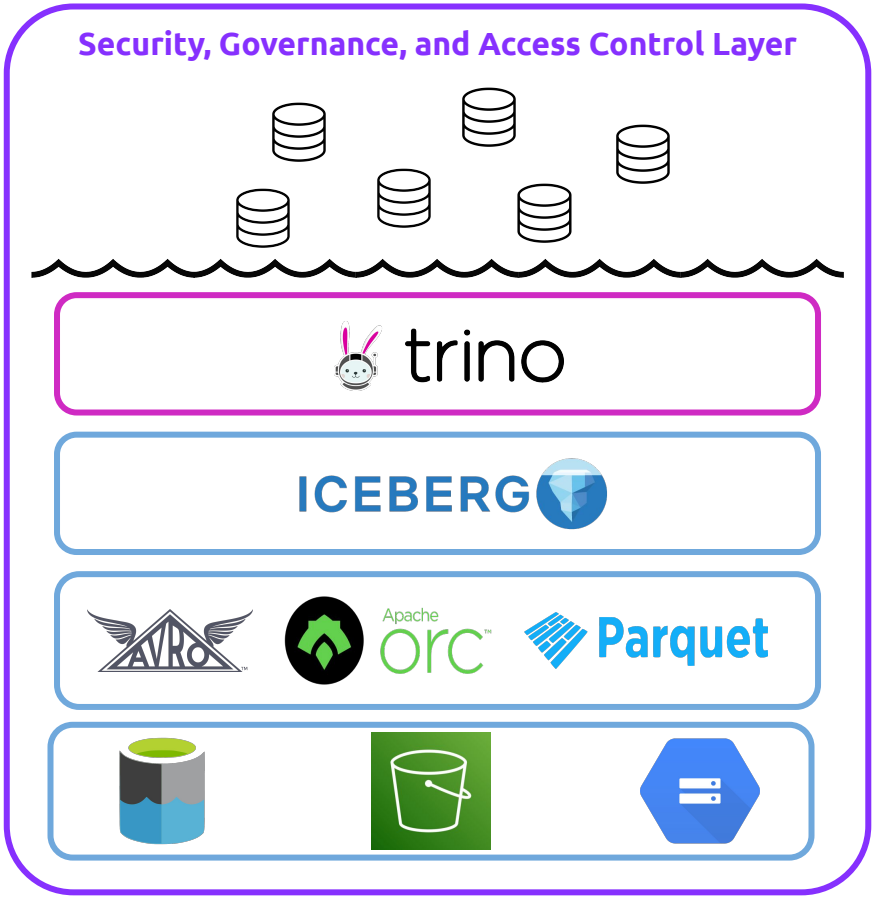
Global federated access to data sources beyond the lake

Compute engine

Table formats

Open file formats

Commodity storage



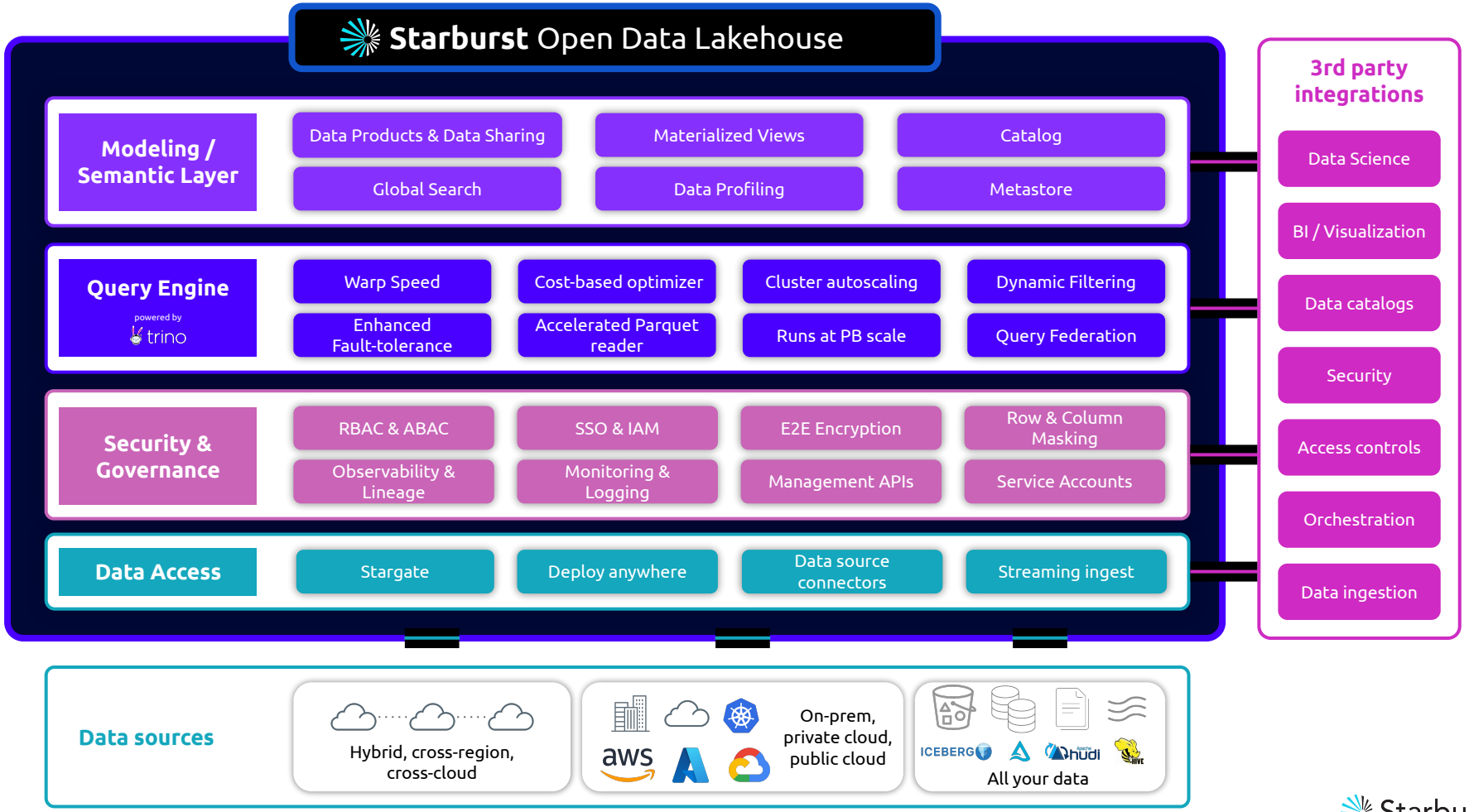
Access data in the orbit

Powers the data lakehouse

Center of gravity

Enables data lakehouses

Starburst


Starburst Open Data Lakehouse
Modeling / Semantic Layer
Data Products & Data Sharing
Materialized Views
Catalog
Global Search
Data Profiling
Metastore
Query Engine
powered by trino
Warp Speed
Cost-based optimizer
Cluster autoscaling
Dynamic Filtering
Enhanced Fault-tolerance
Accelerated Parquet reader
Runs at PB scale
Query Federation
Security & Governance
RBAC & ABAC
SSO & IAM
E2E Encryption
Row & Column Masking
Observability & Lineage
Monitoring & Logging
Management APIs
Service Accounts
Data Access
Stargate
Deploy anywhere
Data source connectors
Streaming ingest
3rd party integrations
Data Science
BI / Visualization
Data catalogs
Security
Access controls
Orchestration
Data ingestion
Data sources
Hybrid, cross-region, cross-cloud
aws
On-prem, private cloud, public cloud
ICEBERG
hudi
All your data

Starburst

Starburst

Data Products

Lesson objectives

The power of data products

1. Explain the basics of data products.
2. Describe the components and defining qualities of data products.
3. Build federated data products from multiple data sources using Starburst.

Data’s biggest challenge

There’s a great divide between data producers and data consumers

- Data consumers fail to ***accurately*** convey their needs to data producers
- Data producers struggle to understand the ***business value*** attached to various requests

Why is everything so grey?

- Data exchanges hands too many times to count
- Consumers will then manipulate said data themselves inaccurately

What are data products?

Data products are ***curated datasets*** packaged to ***create value*** for downstream consumers

- ***Curated datasets:*** Data products are demand-driven and built for a specific need
- ***Create value:*** Data products create value by presenting data in a way that makes it more useful and more accessible

The components of data products

Abstracted data

The most important part of any data product is the data inside it taking the form of:

- Tables
- Views
- Materialized views

Metadata

The table definition associated with the data, including:

- Business context
- Tags
- Lineage information
- Statistics
- Data samples
- Ownership

Access Patterns

Intended access plan for the end user, including:

- Who has access to specific data
- Compute
- How the data is accessed
- Usage examples

Defining qualities of data products

Minimum Viable Data Product

Not all data products have to have each component in this list to be defined as a data product.

The minimum viable data product is a curated dataset created for a specific use case to add value by allowing others to self-serve the dataset for insight.

Independent entities

Each data product has all of the structural components to do its job as a discrete object.

Access to the data product should give you all the information you need to gain insights.

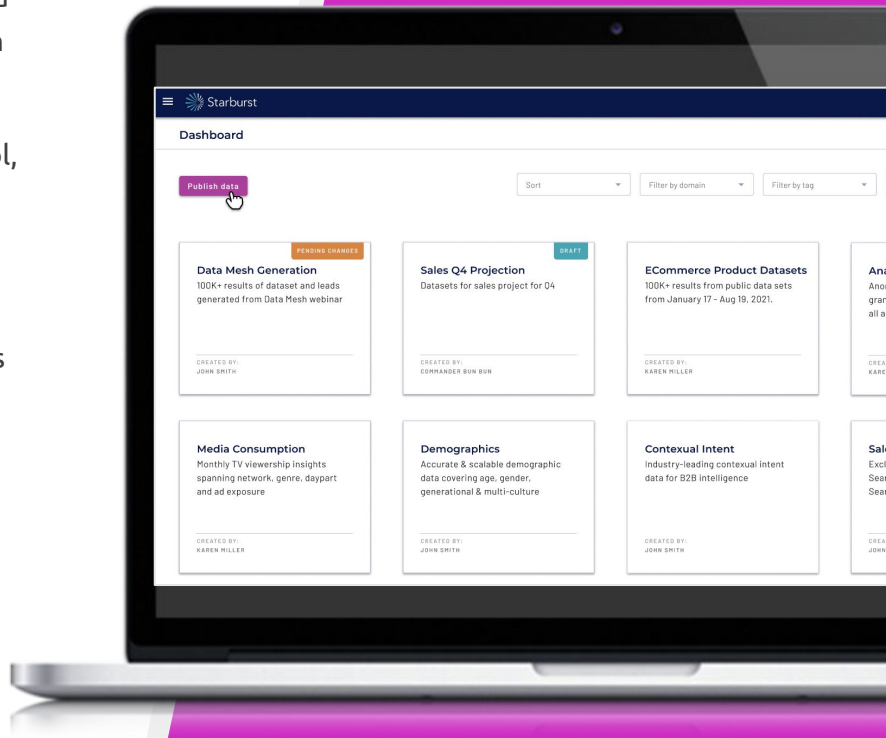
Social dimension

There is a social dimension to data products. They are typically created for others, shared widely, and used across teams.

As such, the collaborative way in which we create them, deploy them, and interact with them is one of their defining characteristics.

Discover, create, publish, manage, and share data products based on multiple datasets

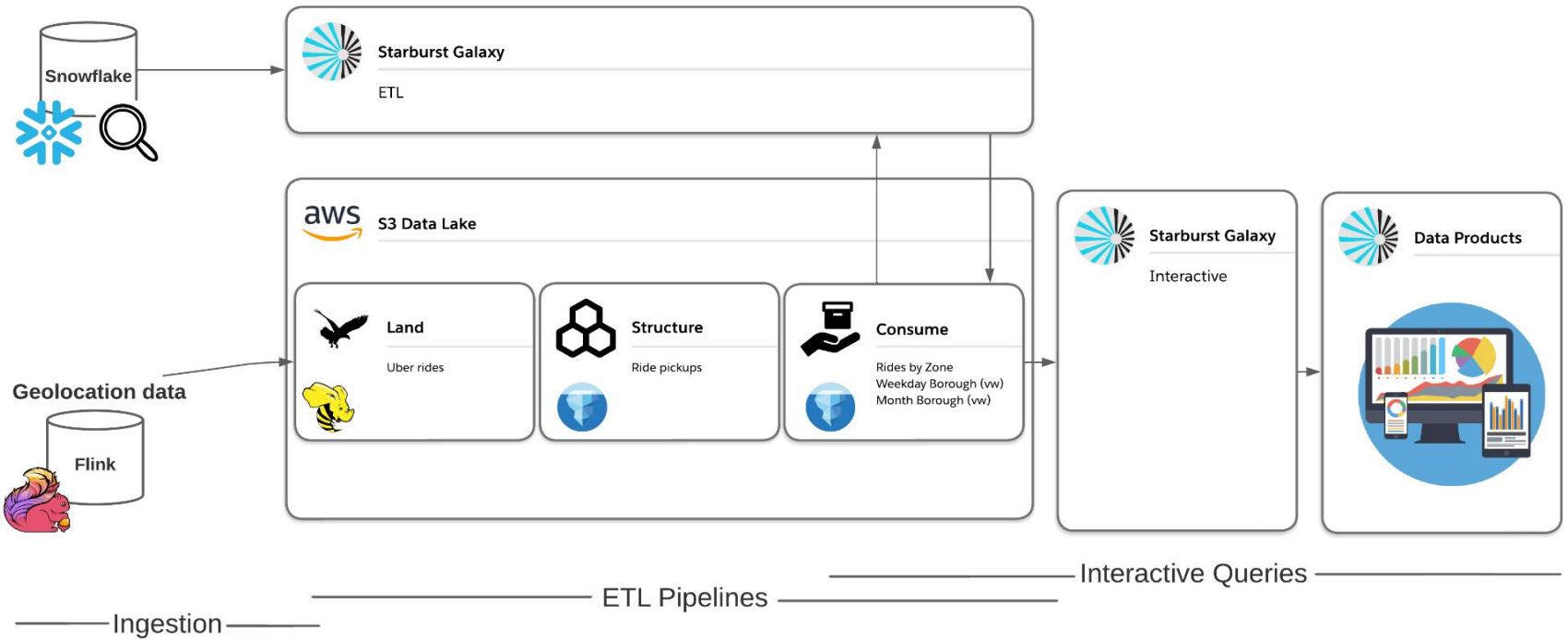
- **Streamlined visibility:** quickly understand the usage metrics, and create, publish, find, and manage curated data products based on multiple data sets
- **Consistent governance:** secure data products with access control, ensuring consistent governance from source level to data products
- **Ultimate accessibility:** query data products that are trusted and approved for frequent business use, rate and share data products internally for use across the organization

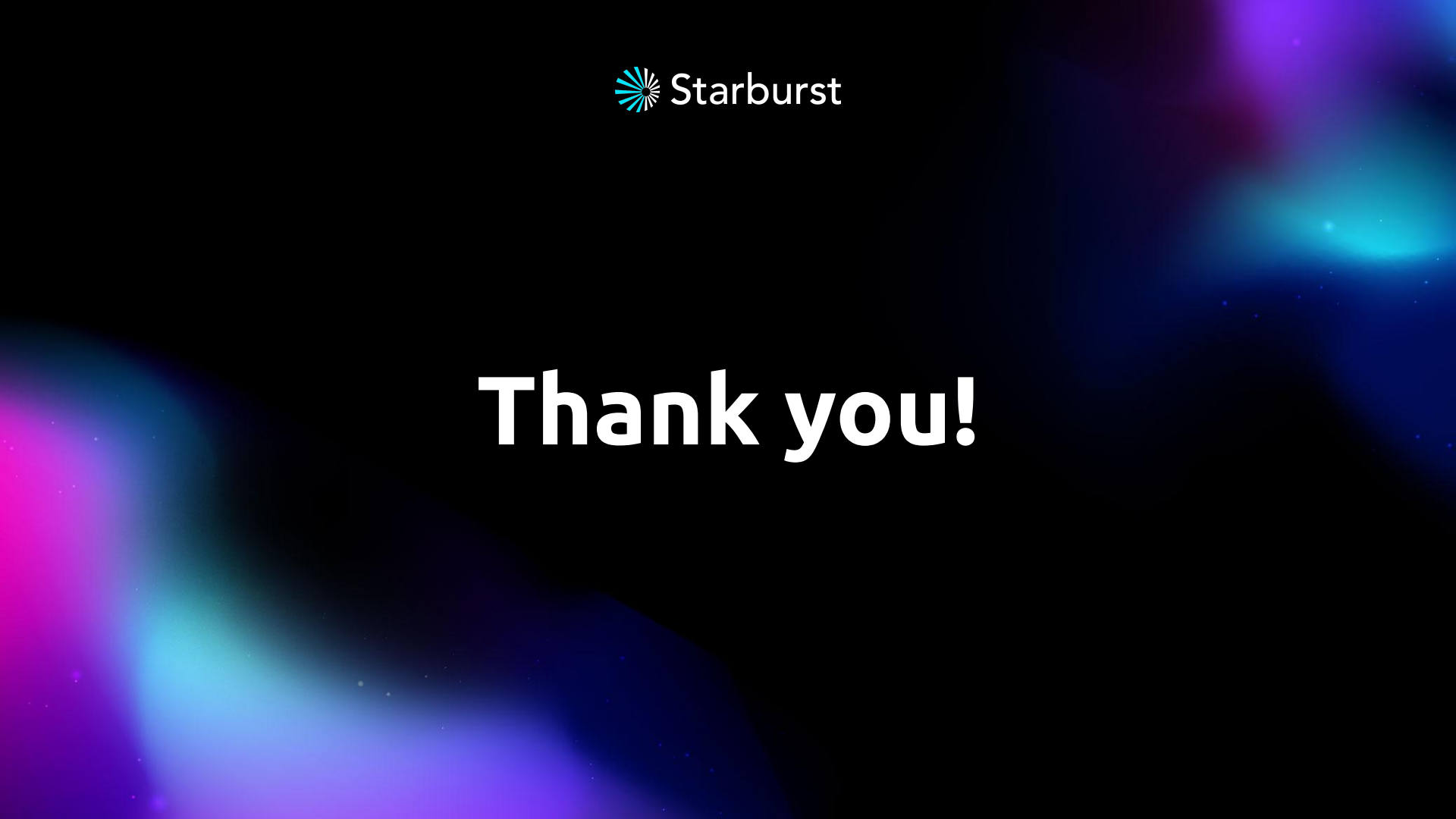


Starburst 101

Part 2: Starburst Demo

Project architecture




Starburst
Thank you!


Starburst