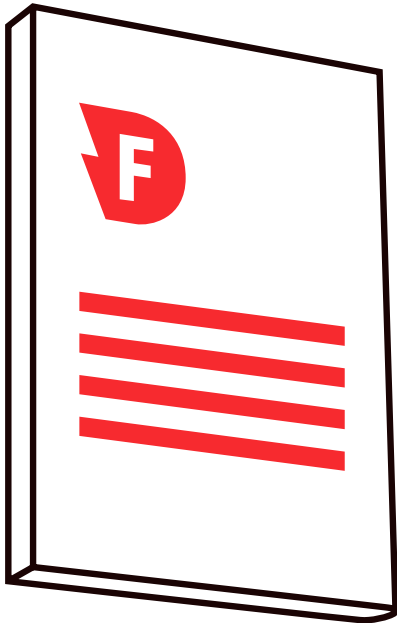


**FIREBOLT**



# The Firebolt Cloud Data Warehouse Whitepaper

3	Introduction
5	Firebolt Overview
7	Firebolt flexible infrastructure
7	Firebolt engines
10	Cloud Storage
13	Metadata service
15	Composable Data Services
15	Data management
17	Query processing
19	Security
21	Observability
23	Workspace
24	Running mixed analytics workloads
27	Summary
27	About Firebolt
28	About AWS Marketplace
28	Contact

# Introduction

Modern cloud data warehouses (CDW) are a critical element in delivering insights through business intelligence, ad-hoc exploration, extract-load-transform(ELT), data science, machine learning (ML), reporting, etc. However, they struggle when serving user-facing data intensive applications (data apps), characterized by high concurrency and low latency workloads over large amounts of data. Today in order to build such applications, developers resort to using specialized systems tuned for specific access patterns that suit the application need. Data needs to be copied from the CDW to these systems, introducing freshness delays, operational complexity, and additional costs. These are the problems that Firebolt is designed to solve. Unlike conventional analytics platforms that juggle with general-purpose cloud data warehouses, proprietary caching layers, and data lakes, Firebolt integrates these capabilities into a singular, efficient, and cost-effective solution. It addresses the core challenges of high latency, concurrency issues, complexity and high total cost of ownership (TCO) that plague many analytics implementations.

Firebolt delivers performance with the simplicity of relational databases. It offers a solution that provides swift query responses and high concurrency across numerous users by combining cloud elasticity in the form of efficient compute and optimized storage, with distributed processing. This approach simplifies the analytics stack, reducing the need for separate infrastructure silos and specialized skills.

Firebolt redefines data analytics by offering a unified platform that enables high concurrency and sub-second latency, empowering data engineers to deliver powerful insights faster and more cost-effectively than ever before.

- **Price-Performance Efficiency:** Firebolt drives performance and cost-efficiency, ensuring organizations can optimize their TCO.
- **High Concurrency and Low Latency:** Designed to handle numerous concurrent users and queries with sub-second response times, Firebolt addresses the needs of customer-facing analytics and emerging data apps.

- **Multidimensional Elasticity:** Firebolt's flexible infrastructure is adaptable to varied workload demands enabling infrastructure right-sizing.
- **SaaS Simplicity:** As a managed service, Firebolt offers just-in-time resource provisioning, online scaling, layered security, and workload observability, making it easy to deliver analytics-at-scale.
- **Developer Productivity:** With SQL compatibility and native support for semi-structured data, Firebolt streamlines data processing and accelerates the development lifecycle.

Firebolt goes beyond the capabilities of earlier cloud data warehouses, which primarily offered elastic scaling and distributed processing but fell short on delivering low-latency and high concurrency. Firebolt combines high concurrency, low latency analytics with the elasticity and scalability of general purpose data warehouses. In this whitepaper, we will describe Firebolt's architecture and components.

# Firebolt Overview

Firebolt cloud data warehouse is designed to address modern analytics workloads. The objective is to deliver concurrency and performance for mixed workloads at the lowest Total Cost of Ownership while empowering developers to build data products and analytics experiences easily and rapidly.

Delivering analytics experiences requires managing the lifecycle of data through various phases: data integration, storage, processing, analysis and serving. Firebolt enhances each phase of this lifecycle, improving overall efficiency and effectiveness of delivering insights. Firebolt delivers these capabilities in the form of data services.

For example, data integration activities such as ELT require handling large volumes of data and preparing them for subsequent analysis, ensuring data follows business rules, and is accurately transformed before being delivered for consumption. Firebolt built data management capabilities such as fast parallel ingestion, low-latency updates/deletes, ACID compliance and multi-stage query execution to deliver efficient ELT. Similarly, Firebolt continues to build specialized capabilities into the data services layer described later. Data services map workload needs to infrastructure components. Firebolt's flexible infrastructure and composable data services are packaged together to provide a high performance, easy to use, fully managed data warehouse.

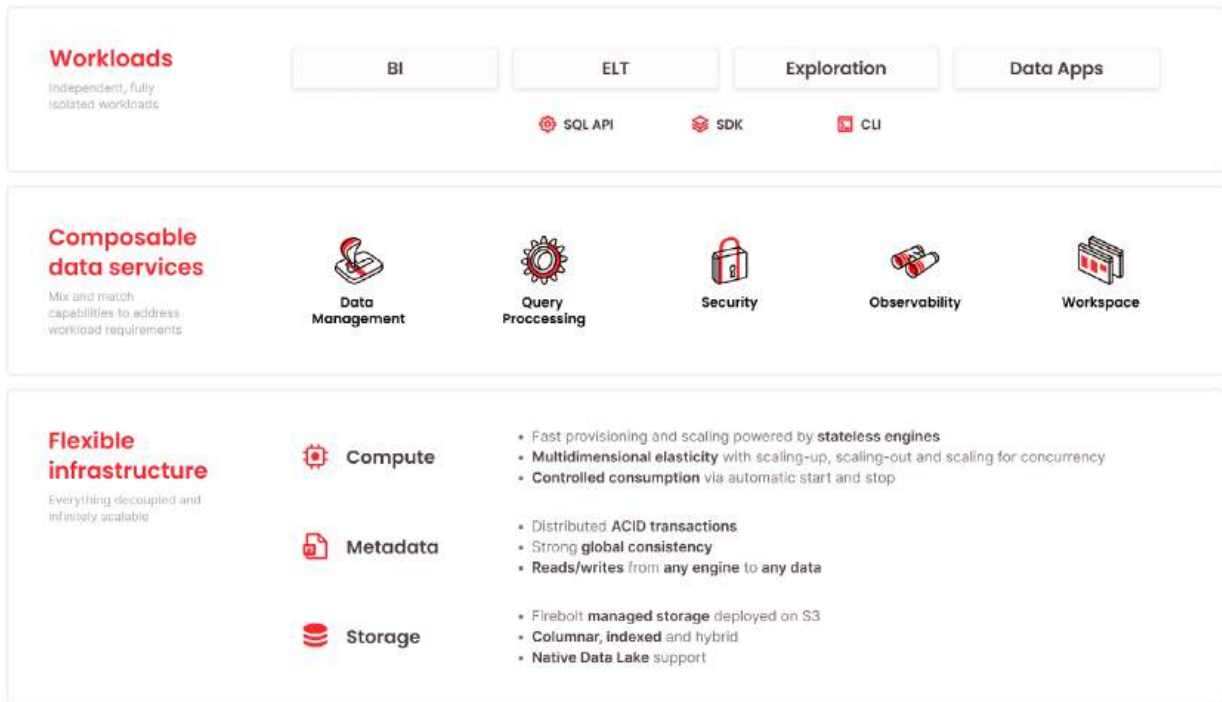


Figure 1. Firebolt platform overview

In the following section, we will take a closer look at each layer in the Firebolt architecture and then cover how Firebolt provides streamlined data management and intelligent query processing to deliver cost effective performance for various data analytics workloads.

# Firebolt flexible infrastructure

At the core of Firebolt is its flexible infrastructure. It is built on a three-way decoupled architecture to provide scalability for each layer of the infrastructure stack: compute, storage and metadata. Firebolt engines represent a stateless compute layer that is responsible for query execution and data management. Multidimensional elasticity, workload isolation and the ability to write from any engine to any Firebolt managed database are key attributes of this layer. Firebolt managed storage serves as the data layer delivering capacity savings and efficient access to data. The highly available metadata service serves as the glue for the infrastructure providing a consistent view of all the metadata enabling ACID compliance and distributed writes. All the above layers work in tandem to address price-performance challenges that analytics platforms experience today. In the following sections, each of the layers and their attributes are described in detail.

## Firebolt engines

Firebolt Engines provide the compute power to address the unique needs of large scale data processing and high concurrency serving workloads.

One of the core attributes of engines is multidimensional elasticity (shown in the figure below). It provides the ability to tailor your engine configuration, through:

- the choice of node type (S,M,L or XL) for vertical scaling (scale-up / down)
- the number of nodes per cluster for horizontal scaling (scale-out / in)
- the number of clusters per engine for concurrency scaling

Engines can be configured or scaled in the form of single node, multi-node or multi-cluster options providing flexibility for the workload at hand. This granular configurability allows you to adjust the compute power of your engines as your data needs grow, ensuring that you always have the right amount of horsepower for your data tasks.

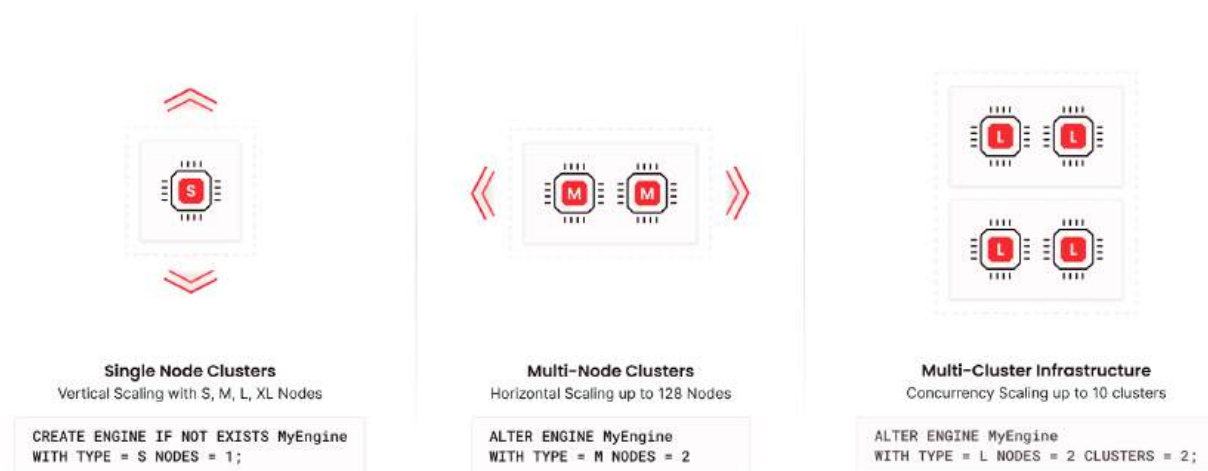


Figure 2. Multidimensional Elasticity of Firebolt Engines

As analytics workloads are launched and expanded, the scaling options from Firebolt allow you to:

- **Start Small and Scale Up:**

- Initiate with a single-node cluster.
- Scale vertically using Small (S), Medium (M), Large (L), or Extra Large (XL) building blocks as needed.

- **Horizontal Scaling for Enhanced Performance:**

- For faster data ingestion or more intensive distributed data processing, expand horizontally.
- Add compute nodes one at a time, up to a maximum of 128 nodes.
- This gradual expansion allows dynamic sharding of large fact tables, particularly beneficial when a table exceeds a single node's capacity.

- **Concurrency Scaling:**

- Increase the number of clusters from 1 to 10 to enhance concurrency without altering the application access endpoint.
- Additional clusters increase concurrency linearly, with each cluster having direct access to the entire dataset.

The scaling approach described above addresses dynamic workloads while addressing budget constraints. Firebolt's scaling methodology does not end here. A

typical data warehouse workload can be broken down into nightly data ingestion, ad hoc analytics workload, high concurrency customer-facing analytics and scale-out ELT etc. Each of these workloads have distinct profiles and disparate resource requirements. Firebolt provides the flexibility to optimize each of these workloads independently.

- **Address mixed workloads with workload isolation**

- Eliminate overprovisioned compute infrastructure with dedicated, right-sized, on-demand compute for each workload.
- Eliminate noisy neighbor issues with dedicated engines with a globally consistent view of all data, ensuring isolation and performance guarantees for each workload type. For example, your ingestion workloads will not get in the way of your customer-facing analytics while still providing a transactionally consistent view of the data.

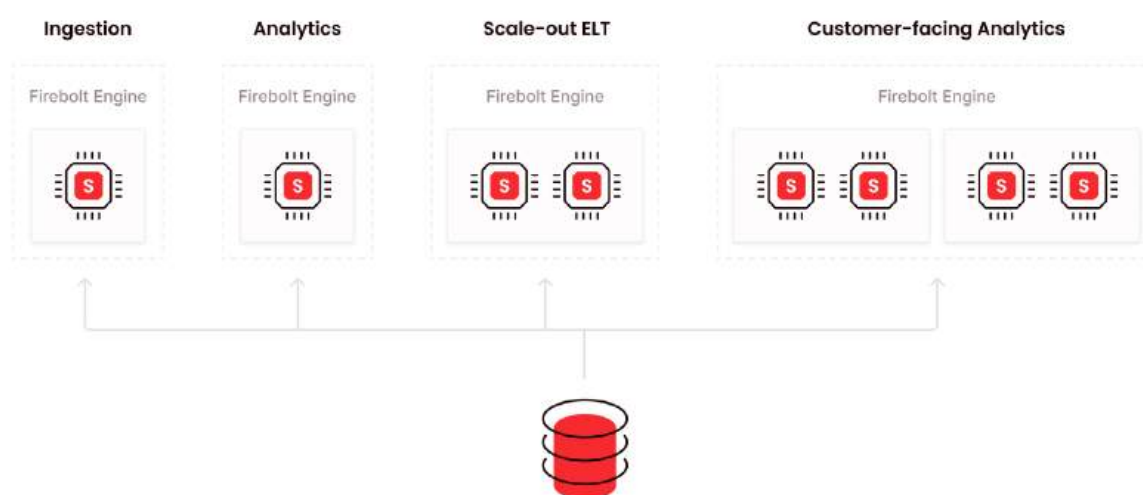


Figure 3. Workload Isolation with Strong Consistency across engines

Firebolt simplifies delivering the above configuration with a simple SQL API-driven interface. You can create, modify, and even scale your engines with simple SQL commands. The platform's built-in features like auto-start and auto-stop add layers of efficiency and cost-effectiveness, ensuring that you're using resources only when you need them. With this approach, each of the workloads mentioned above consume compute resources only when needed, eliminating idle time and driving lower consumption. The compute layer described above is complemented by a managed storage layer, designed for scale, speed and ease of maintenance, which we will discuss next.

# Cloud Storage

Firebolt's cloud storage is designed to deliver capacity savings and data access efficiency through a managed storage layer that stores data efficiently on Amazon's simple storage service (S3). Additionally, native integration to data lakes is provided through direct access to common open file formats.

## Firebolt managed storage

Firebolt's managed storage layer leverages S3's scalability, durability, security, cost-effectiveness, and high availability. However, the choice of S3 introduces challenges in the form of cold reads, request rate limits, and object immutability. To work around these challenges, Firebolt adopted a multi-pronged approach which includes:

### **1. Using tiered storage architecture:**

Leveraging main memory and NVMe SSD on local nodes, S3 data is cached to optimize data retrieval rate and minimize cold reads. This approach provides data locality and eliminates expensive network traversal to read data. Along with tiered storage, Firebolt implemented adaptive prefetch to read data from S3, reducing the impact of cold reads.

### **2. Optimizing physical data layout:**

Firebolt also separates metadata from data to eliminate performance bottlenecks and automatically optimizes S3 storage bucket layouts to maximize read throughput. This ensures that S3 rate limiting has little impact on storage performance.

### **3. Implementing delete logs:**

Finally, the use of a delete log addresses object immutability and allows for updates and deletes in the data warehouse. All these capabilities are addressed collectively through the Firebolt File Format (F3).

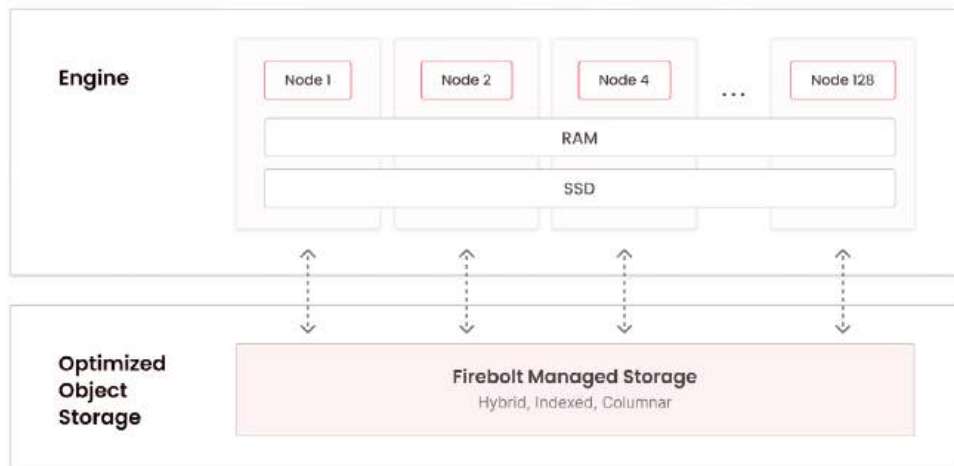


Figure 4. Firebolt Storage

While these storage foundations are essential, to fully support low latency queries there is a need for critical optimizations in the form of indexes. Indexes are an integral part of Firebolt’s storage architecture providing direct, fast access to raw and/or aggregated data. Firebolt implemented sparse indexes to deliver orders of magnitude faster data access through range-level data pruning. Additionally, Firebolt’s aggregating indexes reduce the overhead associated with calculating aggregations.

**Sparse indexes:** A sparse index dramatically reduces the amount of data fetched and processed for each query. It is the primary index of a table, composed of any number of columns listed in any order of a table, and is declared at table creation time. Sparse indexes offer coarse granularity (by representing a range of rows) compared to row based indexes and hence consume less memory while maintaining fast access to large amounts of data. As data is ingested, Firebolt automatically sorts and compresses data based on the sparse index. When queries are processed, data is accessed using the sparse index to provide fine grained data pruning to reduce resource consumption. Scanning less data lowers network utilization and reduces cpu consumption, resulting in lower TCO.

**Aggregating indexes:** Firebolt added aggregating indexes to deliver access to fresh data, fast. Analytics workloads frequently use functions such as COUNT, COUNT DISTINCT, SUM, AVG, etc that require a lot of computational resources. The use of materialized views to pre-compute and store aggregations is a common

technique used in data warehousing to address these workloads. However, materialized views lead to the dilemma of stale data delivered fast vs fresh data delivered slow. Aggregating indexes ensure that aggregations are always fresh and fast. Implementing aggregating indexes is a simple task. Developers can use one line of SQL to create the new aggregating index, as new tables are created, or modify the table and add an aggregating index any time after. You can have multiple aggregating indexes associated with a table. During ingestion, Firebolt automatically maintains each aggregating index which can include raw data, aggregations, and other operators, delivering complex aggregations at low-latency against fresh data. Aggregating indexes access precomputed aggregates and eliminate the need to access the underlying raw data.

## Integration with S3 based data lake

In the previous section, we discussed Firebolt's managed storage option. There is also a need to integrate with data managed by external Data Lakes. To this end, Firebolt enables data integration and exploratory analysis of data on S3 based data lakes through direct read access or with external table definitions. Currently, Firebolt supports direct reading from Parquet and CSV files. For requirements to read JSON, Parquet, CSV, ORC or AVRO files, Firebolt allows external table definition on these open file formats. This capability allows federated querying of data managed on data lakes just as if they are part of the Firebolt database.

To ingest data into Firebolt's optimized storage format, the "COPY FROM" command can be used, with support for Parquet and CSV files. Schema inference, file level filtering, error logging are provided with this command to simplify the process of onboarding data.

Exporting query results and data to external data lake is provided through the "COPY TO" command. Data can be exported using Parquet, CSV or JSON formats.

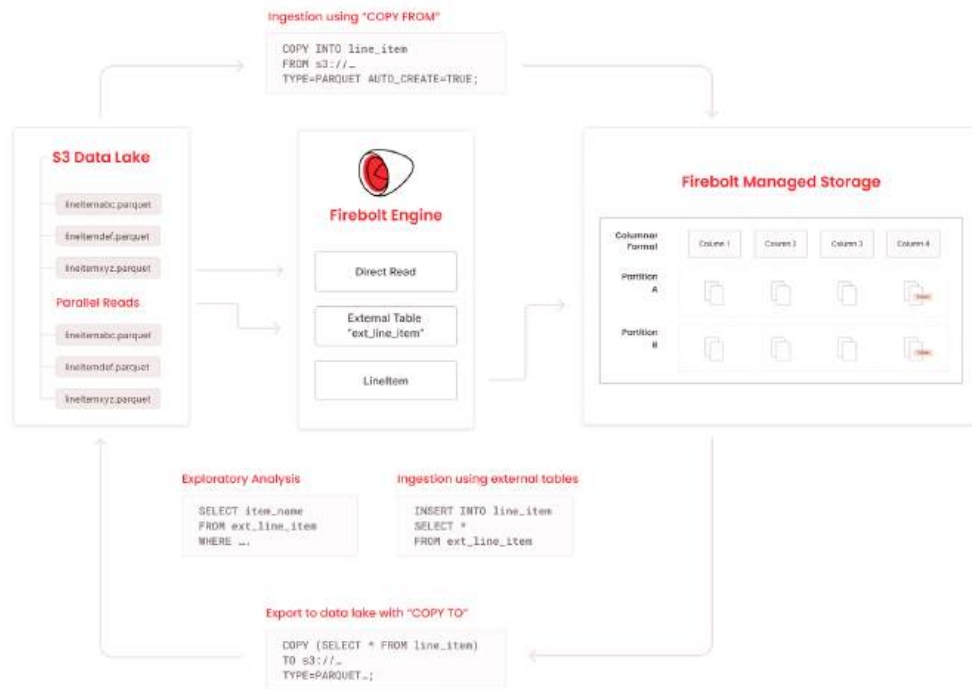


Figure 5. Data Lake Integration

Compute and storage are foundational infrastructure elements, providing a globally consistent view of the analytics infrastructure, including data and system information. However, coordination across infrastructure and service components is the function of a distributed metadata service, covered in the next section.

## Metadata service

Firebolt's distributed metadata service maintains a consistent view of the entire analytics landscape, from initial registration to delivering insights to end users. While much of the activity within a data warehouse centers on the compute and storage layers, the metadata, often unseen by the user, is critical in a distributed system for ensuring smooth operations. As 1) objects are created, modified, or removed, 2) data is ingested, updated, or deleted, and 3) new users are on boarded or infrastructure is scaled, the state of the system and hence its metadata is constantly changing. The metadata service is responsible for presenting a globally consistent system image at all times by managing transactional consistency within Firebolt.

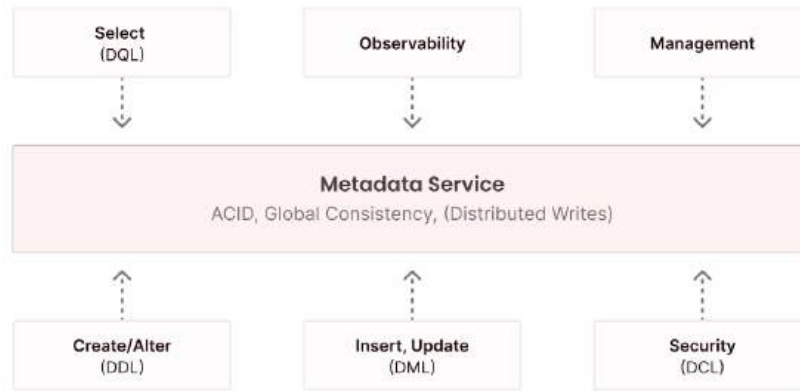


Figure 6. Metadata related operations

The metadata service is a critical infrastructure component that decouples metadata from the compute and storage layers. This decoupling makes the compute layer fully stateless, ensuring that any engine can be used with any available database or any database can be accessed from within any engine, enabling workload isolation and scalability.

Moreover, the metadata service supports caching and incremental updates, enabling a large number of fast, concurrent operations while always presenting a consistent view. Firebolt metadata service features a low latency distributed transaction manager which ensures ACID transactions and global consistency. This metadata service is supported by a high performance key-value backend. Even though Firebolt is an analytics platform, as opposed to transactional, the ability to execute hundreds of transactions per second is enabled by the metadata service.

Metadata information can be accessed from any engine. Firebolt also provides a serverless SQL API endpoint in the form of the System Engine to enable monitoring and managing of data warehouse resources. The system engine is provided at no cost. System level metadata is accessible through information\_schema objects that cover security and observability for all objects within the Firebolt platform.

So far, we have covered the infrastructure elements within Firebolt. In the upcoming sections, we cover how data services leverage the benefits of the infrastructure.

# Composable Data Services

Data Services are building blocks that run on top of the flexible infrastructure described in the previous section. Data services support data management, query processing, security and observability needs and requirements. For example, to deliver rapid ingestion and fast interactive queries, data management and query processing need to leverage the capabilities of the underlying infrastructure.

## Data management

Analytics professionals face a range of daily challenges, including building data models, integrating new data sources, meeting tighter ELT timelines, staying updated with changes across various data sources, managing infrastructure costs, and ensuring that resources are available and transactions are consistent. Firebolt provides a solid data management foundation to tackle these challenges squarely by supporting the entire data management lifecycle, ranging from data modeling to ingestion, transformation, and deletion.

Firebolt enhances data modeling by providing techniques to efficiently organize and access data tailored to specific business needs. It handles diverse querying demands such as table joins, large data aggregations, and detailed filtering through features like join accelerators, aggregating indexes and sparse indexes. Firebolt employs join accelerators to optimize resource-intensive join operations, maintaining efficiency and lowering integration costs. It also uses aggregating indexes to quickly access precomputed data. These capabilities translate into support for data models including star schema, snowflake or a denormalized one-big-table model. While these approaches address structured data, Firebolt supports semi-structured data using array data types along with a comprehensive suite of functions and lambda expressions. Semi-structured data can be processed using schema-on-read or by flattening for performance.

From a data modification standpoint, each request in Firebolt is treated as a distinct implicit transaction with its own ID and timestamp. To increase scalability, Firebolt implements optimistic concurrency, allowing multiple transactions to

proceed simultaneously without locking. Each transaction checks for concurrent modifications to the same data before committing - if conflicts are detected, the transaction is rolled back. Firebolt also maintains multiple versions of data items, enabling transactions to operate on snapshots of the database at specific moments, ensuring that changes made by one transaction remain isolated until completed.

Ingestion is typically the first step in onboarding data. Firebolt provides a COPY FROM command to support these activities. The schema inference process is built into the COPY FROM command to simplify data discovery and ingestion. With this process, rows from the data files on S3 are sampled to automatically infer column names and data types, thus simplifying table schema creation. The entire ingestion process adopts a parallel-pipelined approach to move data rapidly from the data lake into an optimized columnar format. Ingestion is an atomic process that automatically leverages multiple stages to streamline the flow. Once ingested, data is stored in internal structures called tablets (see figure below). Data in the tablets is sorted, compressed, and indexed for efficient storage and retrieval. Table design and choice of primary index play an essential role in determining the compression ratio and data pruning efficiency. Note: Firebolt supports column, partition, tablet, or range-level data pruning techniques.

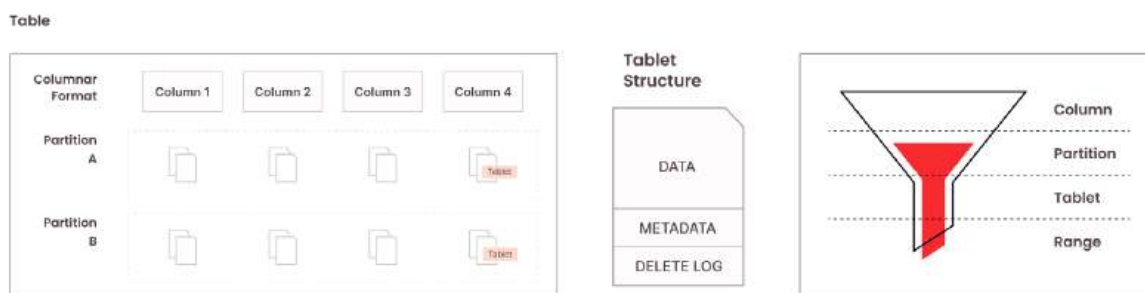


Figure 7. Firebolt conceptual data structure

The size of source files and the engine configuration used for ingestion can directly impact data ingestion time windows. Firebolt's multidimensional elasticity allows users to configure engines to meet price-performance objectives during ingestion. Incrementally adding nodes to the engine drives faster ingestion.

Firebolt also supports fast data updates and deletes by using a delete log for each tablet to track changes. With this approach, the tablet structure does not need to be immediately updated, thus minimizing the performance impact of updates and deletes. However, frequent deletes and updates can fragment tablet and table-level data. Firebolt has mechanisms to address potential fragmentation and optimize table and tablet quality.

While data modifications are necessary, these operations are resource intensive when maintaining precomputed aggregates, rollups, or materialized views. To ensure consistent query acceleration without manual user intervention, Firebolt automatically maintains all provided indexes, including aggregating indexes - even when aggregating indexes contain non-update-friendly aggregates such as COUNT DISTINCT, MIN and MAX functions.

## Query processing

In this section, we look at how the infrastructure and data management foundations dovetail into Firebolt's query processing stack. To consistently meet the low latency, high concurrency demands of data applications, the query processing stack needs to scale and adapt to dynamic query patterns, variations in concurrency, volume and velocity of data.

Firebolt's query processing stack comprises multiple layers: First is the admission controller, where the query enters the system, followed by the query planner and optimizer. Last is the run-time engine.

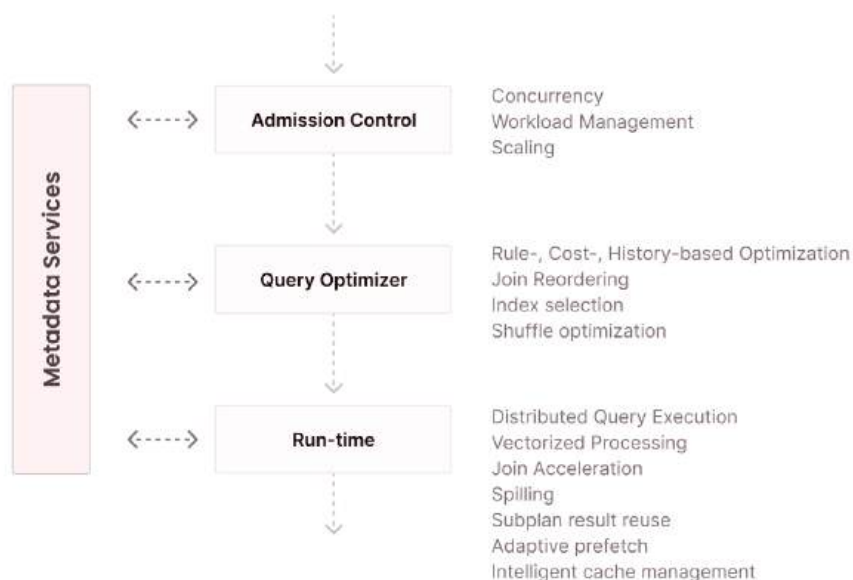


Figure 8. Firebolt query processing stack

At the forefront of query processing is the autoscaling-capable admission controller, designed to handle high concurrency and scale with minimal contention points. This component acts as the gatekeeper of resources, routing queries where and when the required resources are available.

Next in line, the query planner and optimizer employs a combination of cost-based and history-based optimization techniques. The query optimizer considers factors like data statistics and index availability, aiming to minimize resource usage while improving the execution time without the user having to worry about it. Furthermore, Firebolt's optimizer has the ability to learn and adapt based on historical data. By understanding past query patterns and outcomes, the optimizer can make informed decisions, leading to consistent performance.

The next significant component in the stack is the run-time or execution engine where the optimized query plan is put into action. In distributed data warehouses, the run-time orchestrates data retrieval, distribution, parallel processing and shuffling as a cluster wide operation. Users leverage techniques like indexing and partitioning to manage performance complemented by the run-time which leverages multi-threading and vectorized query execution behind the scenes. Additionally, the run-time manages memory and disk I/O, executing complex operations like joins and window functions. Firebolt's join acceleration and subplan result reuse facilitate low latency and high concurrency. Data applications often

have predictable query patterns. The planner automatically detects common subplans and when applicable the results are reused. This subplan result reuse reduces resource utilization. Designed for memory efficiency, Firebolt's single node runtime extensively uses caching of subplan results.

To this point, we covered data management lifecycle and query processing capabilities. From a management perspective, every workload has specific security, observability and collaboration needs. These foundational elements are covered next.

## Security

Security is a critical component of running a data warehouse as a service. Sensitive data in the wrong hands can wreak havoc, resulting in loss of customer trust and damaging the ability to conduct business. The proper security controls are needed to ensure data is always secure. Firebolt leverages a layered, shared responsibility model to secure service elements. Security elements on Firebolt include infrastructure, network access, identity management, access control, and data protection, all based on the SQL object model. A simplified Firebolt object model is shown on the next page.

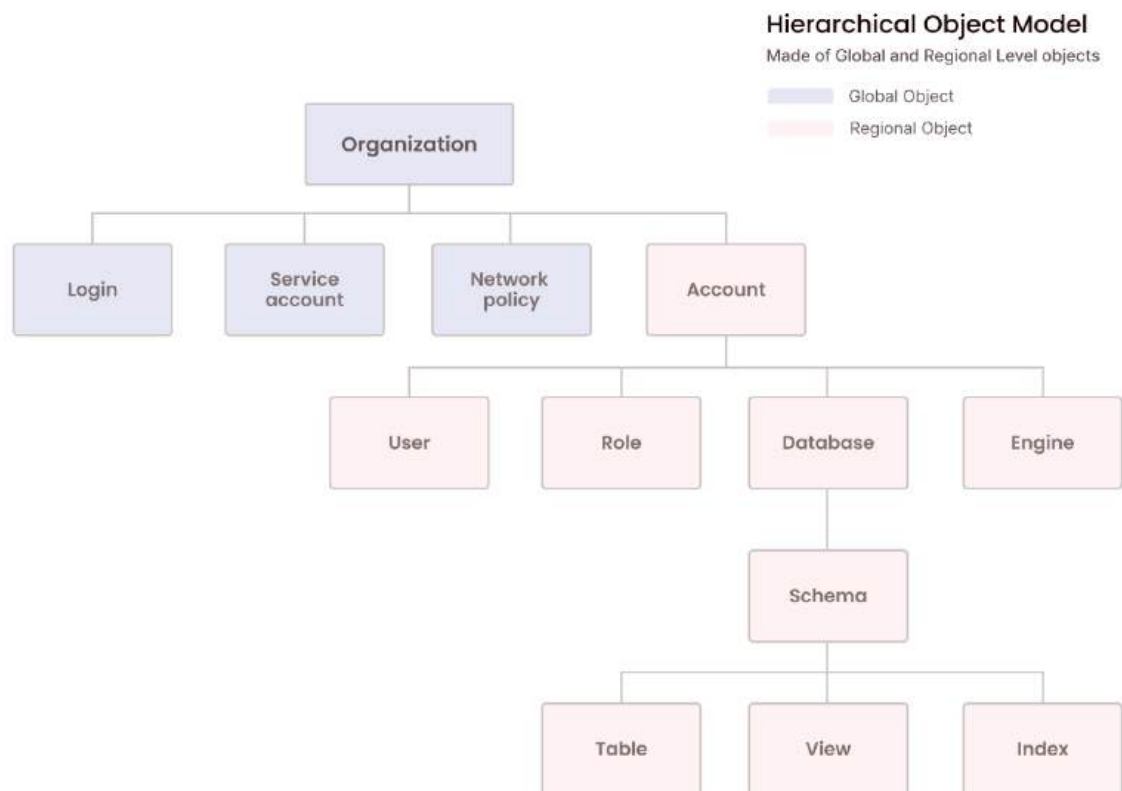


Figure 9. Firebolt object model

The concept of a top-level organization establishes global visibility and governance across all analytics resources. Authentication and network access are controlled at this layer. For authentication purposes, Firebolt employs Auth0 for identity verification. Firebolt also supports multi-factor authentication (MFA) and single sign-on (SSO) integration to bolster security measures and reuse existing infrastructure. To control network traffic from allowed IP addresses only, Firebolt provides customizable network policies. These network policies act as allow or deny lists, permitting or denying access from specific IP addresses or ranges. Additionally, this control can be customized for specific authenticated users or service accounts.

The next step in the security chain is authorization or access control. Authorization ensures the user has appropriate permissions to access the system or system-level resources it attempts to use. Firebolt implemented role-based access control (RBAC) mechanisms for this purpose. Firebolt's RBAC system provides built-in (e.g., `account_admin`, `org_admin`, etc.) and user-defined roles to control access to Firebolt objects. Built-in roles come with a pre-built set of permissions while user-defined roles are used to customize access to specific objects for different users.

For example, a user-defined role, say “sales,” can be granted to a user “John,” with “usage” only permission to the “sales\_db” on a specific engine. Permissions are assigned using standard “Grant” and “Revoke” SQL statements.

From the data encryption perspective, Firebolt supports data-in-transit and data-at-rest encryption models. Firebolt uses strong encryption methods to secure data over the network and to allow protection from various attacks (e.g. eavesdropping, replay attacks, data tampering and man-in-the-middle attacks). Key management is essential for protecting sensitive information and secrets and is performed through a robust system to securely generate, store, rotate, and retire encryption keys. Secure transmission of encryption keys from the key management system to the encryption or decryption components is performed to prevent interception or tampering. Moreover, Firebolt validates access with each transaction, ensuring that data is always accessed securely and appropriately. This validation, combined with secure key management and network controls, significantly reduces potential attack vectors from both external and internal threats, fortifying the overall security posture of Firebolt’s data warehouse as a service.

## Observability

As a fully managed service, Firebolt eliminates the need to manage low level compute, storage, networking and security components. However, understanding workload profiles, resource consumption and spend patterns are vital to managing performance and cost. Compute resources should be right-sized and scaled on-demand to tackle workload needs, thus avoiding overprovisioning and waste. All this requires observability into resources used.

Firebolt’s observability starts at the organization level. At this level, comprehensive visibility is provided into security, resource consumption, and billing. This global view allows organizations to understand trends in terms of spend across business units and their granular resources. Analytics resources in Firebolt themselves are tied to accounts in the form of engines and databases. Firebolt provides visibility into access control, engine configuration, query execution metrics, and storage utilization at this level. For example, system configuration is visible through information\_schema views such as ‘engines’, ‘databases’, ‘tables’ and more.

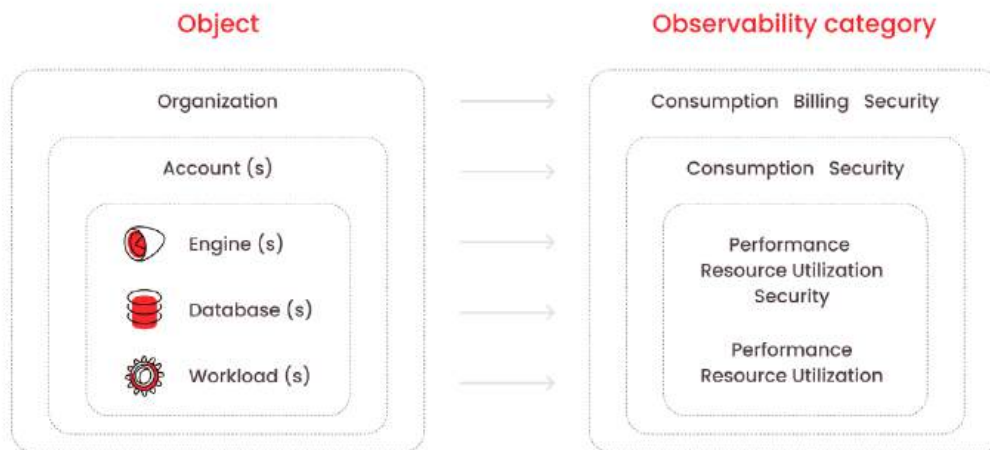


Figure 10. Firebolt observability

From a workload perspective, Firebolt’s observability focuses on query history and engine history.

**Query history** allows users to access detailed information about past queries (via SQL interface), such as query text, execution time, data volume scanned, and the user executing the queries. This data can be used to identify slow or inefficient queries, recognize patterns in query performance, and implement optimizations like adjusting SQL queries or modifying indexing strategies based on business needs. Query execution metrics can be accessed through SQL from ‘engine\_running\_queries’ or ‘engine\_query\_history’ information\_schema views. Detailed statistics on execution time, amount of data scanned, CPU and memory usage can be tracked at the individual query level. Granular metrics help evaluate each query for its efficiency and resource utilization. Furthermore, the user/service account information provides visibility into access controls as well.

**Engine history** provides insights into engine utilization and performance metrics, available through the Firebolt user interface. This includes tracking engine start and stop times, resource consumption, and performance across different workloads. This data aids in making informed decisions about engine configuration adjustments—like scaling computational resources to match workload demands—and determining optimal times for scaling operations to balance performance and cost. From an overall engine sizing standpoint, ‘engine\_metrics\_history’ provides engine utilization over time. This engine-level view aggregates overall CPU, memory and local solid state storage usage, providing insights into engine right-sizing.

## Workspace

Analytics requires cross-functional collaboration across different roles. The personas that participate in the whole development lifecycle can vary from organization to organization and can include administrators, data architects, data engineers, data owners/stewards, devops engineers, data analysts, application developers, and others.

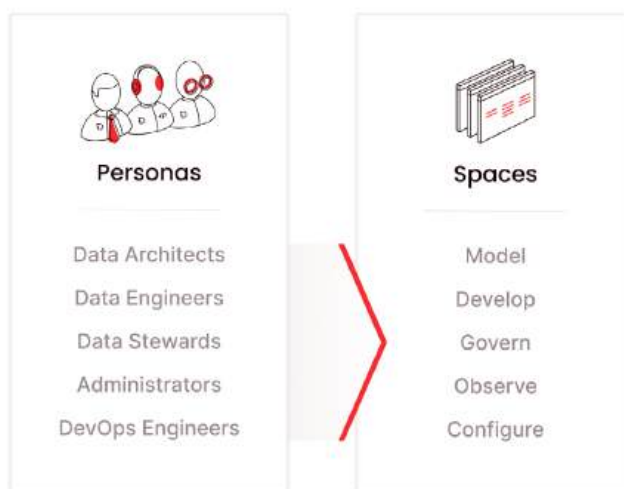


Figure 11. Workspace and Personas

Firebolt organized the user experience to address configuration, data modeling, development, governance and monitoring. These capabilities manifest as “workspaces” within the Firebolt WebUI to simplify adoption, visibility, and management of the analytics infrastructure. For example, the security administrator will leverage the “Configure” workspace to lay the security foundations through single sign-on, network policy, and multi-factor authentication setup. The organization and accounts hierarchy provides additional granularity while interacting with objects as an administrator or a developer.

# Running mixed analytics workloads

The challenge in any analytics platform is the ability to address the full range of workloads from ELT to interactive analytics that have different workload profiles. Let's review how these workloads benefit from Firebolt's capabilities.

- **High-Performance Ingestion**

Firebolt excels at data ingestion, supporting both batch and trickle data loads. Its ingestion framework provides schema inference to simplify the process of onboarding data and is built to handle high volumes of data efficiently, minimizing latency and ensuring data is quickly available for analysis. Compute infrastructure can be shaped according to ingestion needs to address tight data ingestion windows.

Firebolt's batch processing capabilities allow for the rapid ingestion of large datasets, making it suitable for scenarios where data is collected in intervals and processed in bulk. The ability to tailor engine topology and on-demand nature of Firebolt allows scaling of ingestion while helping manage costs.

Near real-time ingestion of numerous small files in the form of trickle ingestion is crucial for operational analytics use cases that require continuous data flow into the warehouse. Firebolt manages concurrent writes and reads to ensure that transactional consistency is maintained throughout the process. It also provides tools to reduce fragmentation of the underlying storage due to near real-time ingestion of small files in the form of trickle ingestion.

- **Efficient ELT**

Firebolt's architecture uniquely combines a columnar storage format with a sophisticated multi-stage distributed query execution engine, specifically crafted to enhance performance and manage large-scale data transformations with exceptional efficiency. Firebolt breaks down queries into smaller sub-tasks, which are then processed in parallel across multiple nodes. This distributed nature of processing ensures that Firebolt can scale horizontally to handle increases in data volume or query complexity without a drop in performance. It effectively utilizes all available compute resources, thereby

optimizing operational efficiency.

Complex joins, especially across large datasets, are notoriously challenging and resource-intensive. Firebolt addresses this by employing techniques like subplan reuse, where the results of intermediate queries or joins are cached, making the join operations significantly faster and more efficient.

Firebolt's architecture is inherently elastic, allowing it to dynamically adjust and scale resources according to the fluctuating demands of data volume and velocity, typical in ELT (Extract, Load, Transform) processes. This elasticity ensures that Firebolt can maintain optimal performance levels, adjusting resources in real-time to handle spikes or drops in data processing requirements.

- **Customer facing analytics and Business Intelligence**

Lookup queries and aggregations are fundamental to customer facing analytics and BI workloads. Firebolt is designed to execute these operations with exceptional speed and efficiency.

Firebolt's sparse index, significantly reduces the time required to execute lookup queries through highly efficient data pruning. Sparse indexes reduce the amount of data that need to be transferred from object storage resulting in reduced networking and computational needs. The platform's ability to perform fast aggregations is powered by its aggregating index. This allows for quick summarization of data, essential for reporting, dashboards, and analytical insights.

Customer facing analytics requires that query response times are consistent and are not impacted by other workloads on the same data warehouse. Firebolt isolates latency sensitive workloads on their own dedicated infrastructure preserving the end user experience.

Table on next page summarizes how Firebolt capabilities support different workload profiles.

<b>Workload</b>	<b>Firebolt capability</b>	<b>Benefit</b>
ELT	Multi-stage Distributed Execution	Address complex queries
	Join Acceleration/Subplan reuse	Optimize joins on normalized data model while reducing memory footprint
	Multidimensional elasticity	Address cost efficiency by right sizing infrastructure
Customer facing Analytics	Primary Index	Efficient data pruning to reduce data movement and faster queries
	Subplan reuse	Use in-memory data to support sub-second latency
Business Intelligence	Aggregating Index	Fast, efficient computations that eliminate summary tables and secondary data pipelines
	History based optimizer	Consistent response time with high quality query plans
	Workload Isolation	Eliminates impact of external workloads on customer facing analytics

# Summary

Beyond traditional analytics, the need to support operational and customer-facing interactive analytics has completely changed what is required from a cloud data warehouse. Decoupled compute, storage and metadata, tiered data management, specialized indexing, advanced query optimization, and execution are all needed to deliver order-of-magnitude improvements in speed, scale, and efficiency. Firebolt combines these capabilities with ease of use, security, governance and observability to provide a data platform that addresses the day-to-day requirements of analytics delivery. Firebolt's performance acceleration and decoupled infrastructure enable high performance and concurrency, combined with improved cost management and the simplicity of SQL. Sub-second aggregations, large scale joins, rapid ingestion, concurrency scaling, ease-of-use, cost optimization are all delivered in a single platform: Firebolt. Adoption is only a click away.

## About Firebolt

[Firebolt](#) is the world's fastest cloud data warehouse, purpose-built for high performance analytics. It provides orders of magnitude faster query performance at a fraction of the cost of the alternatives by combining the simplicity, elasticity and low cost of the cloud with the latest innovations in analytics. Companies that adopted Firebolt have been able to deploy data warehouses in weeks and deliver sub-second performance at terabyte to petabyte scale for a wide range of interactive, high performance analytics across internal BI as well as customer facing analytics use cases.

Firebolt is founded by members of the original team at Sisense, A global BI leader, and backed by Zeev Ventures, TLV Partners, Bessemer Venture Partners and Angular Ventures.

# About AWS Marketplace

[AWS Marketplace](#) is a curated digital catalog that makes it easy for customers to find, buy, consume, and manage third-party software, services, and data that customers need to build solutions and run their businesses.

## Contact Firebolt

For more information about Firebolt, you can contact us at anytime at:

<https://www.firebolt.io/contact>

Or e-mail us at: [hello@firebolt.io](mailto:hello@firebolt.io).