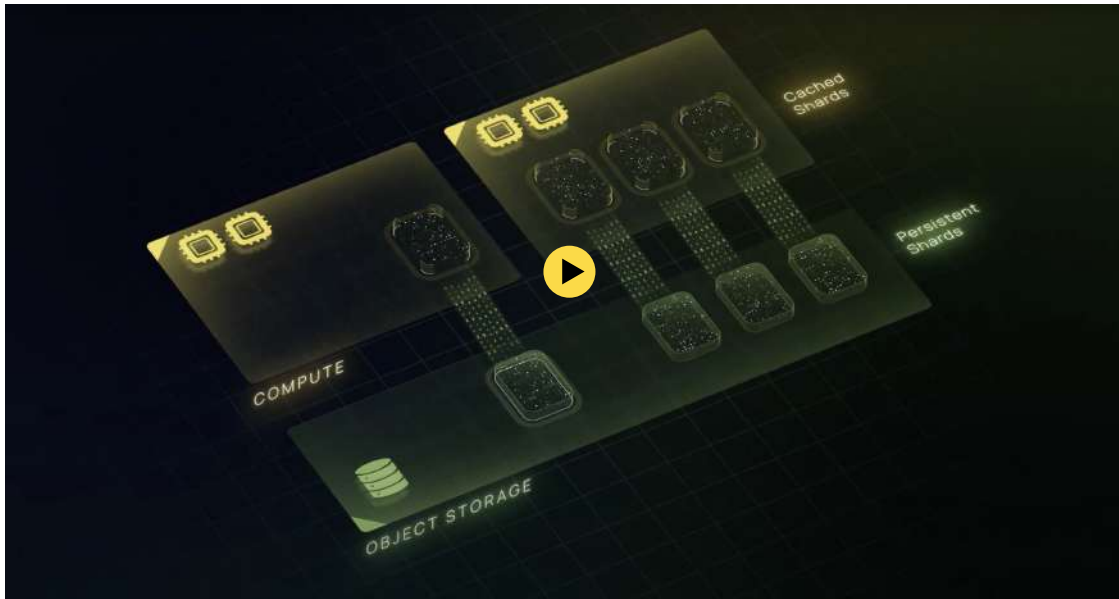


# How Yellowbrick Avoids Data Load and Query Conflicts



Traditional data warehousing solutions often struggle with maintaining consistent query performance while loading data, frustrating users and delaying insights needed to make business impacting decisions. Yellowbrick's patented Direct Data Accelerator® architecture enables consistent query performance during concurrent data loads.

## Don't Take Our Word For It: See How Our Customers Benefit

ushur®

Ushur the Customer Experience Automation™ Platform had a data platform that was slowing down customer onboarding and getting in the way of their ambitious growth goals. Building per-customer data cubes was time consuming and costly and resulted in stale data. With Yellowbrick, Ushur had the confidence to let their application query the data platform directly even while data was being loaded. They sped-up customer onboarding, removed the cost and effort of building cubes and improved data latency.

[Learn More →](#)

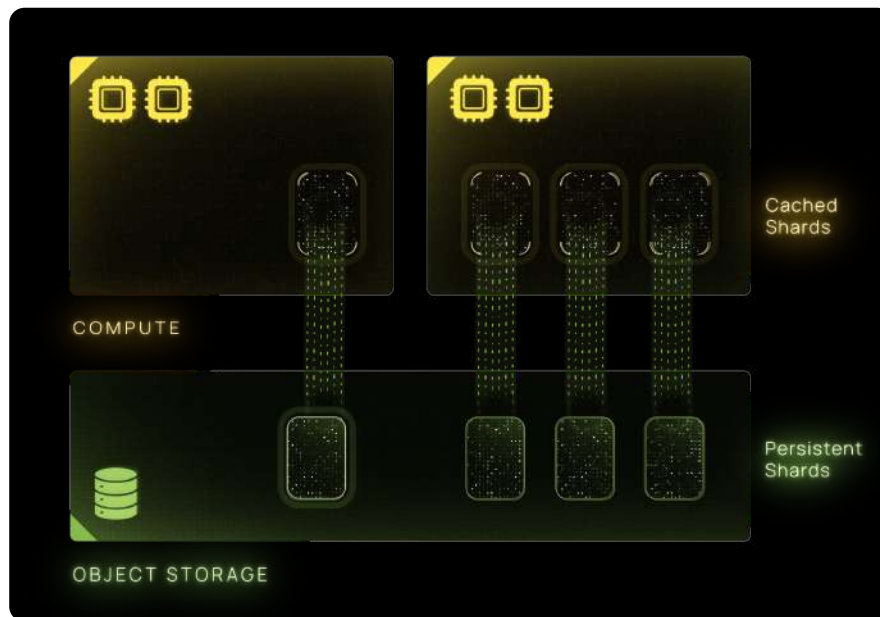
## PROOF POINTS

### Avoid Data Load & Query Conflicts

Are you struggling to balance data engineering ingestion and transformation tasks with end user query workloads? You are not alone. Solutions like AWS Redshift often struggle resulting in complex and costly workarounds and stale data. Yellowbrick's data platform is built differently. Imagine a world where you don't have to stress about data engineering jobs impacting query performance or vice-versa.

Yellowbrick addresses this challenge with an innovative architecture designed for high- demand, concurrent operations. Our approach uses a combination of decoupled storage and compute, advanced workload management and a hybrid storage engine to enable uninterrupted query execution while loading data.

## Decoupled Compute & Storage



In Yellowbrick, storage and compute are independent. By separating these two functions, multiple clusters can run data loads and analytics queries at the same time, ensuring that data is not only current but also accessible for analysis with low data latency.

Yellowbrick's data platform permits multiple data loads concurrently, treating them like any other query. The system can allocate up to 150 concurrent query slots per cluster. For environments with exceptionally heavy data load demands, a dedicated load cluster can be provisioned, fully isolating the impact of loading operations from user queries. Writes can be initiated in any cluster with data remaining transactionally.

Data written to Yellowbrick is intelligently distributed in shards across the disk and cached on compute nodes. This sharding allows for efficient data retrieval and management, providing fast read and write capabilities. In addition, when data is written directly to an object store, Yellowbrick utilizes the maximum network bandwidth by compressing shards in memory in parallel across all cores before writing. Whether data is loaded from external sources or inserted as a result of ELT operations, all new data follows the same path – it is written into shards and garbage-collected as necessary, maintaining system efficiency and data integrity.

---

## Advanced Workload Management

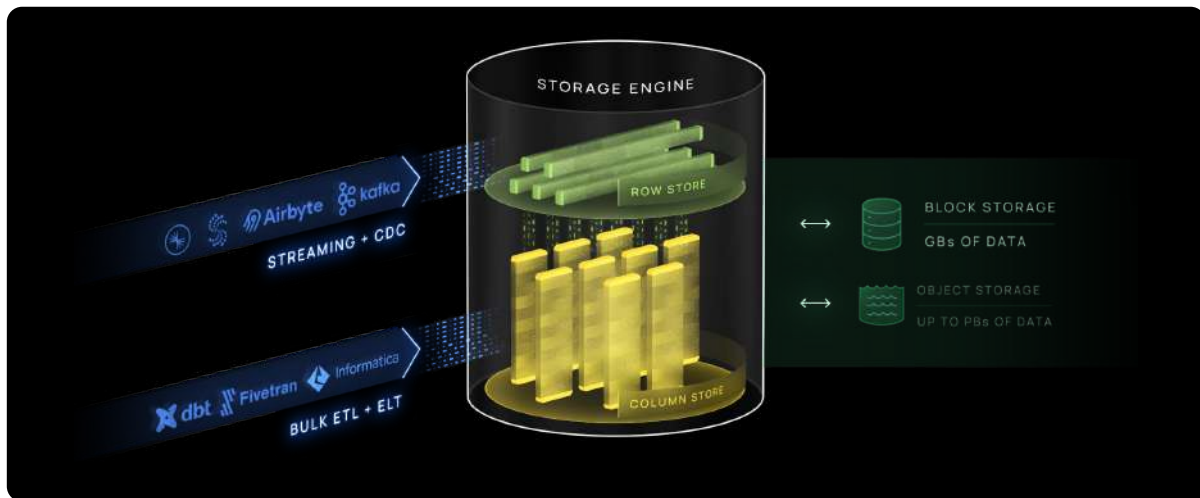
In a busy system, prioritization is key to ensuring key workloads get the resources they need at the right time.

Yellowbrick's sophisticated workload management system runs on every cluster. Each compute cluster can adopt a different, configurable workload management profile. In our workload management implementation, compute, memory and temporary storage resources are split across pools. Rules map incoming queries to a particular pool based on attributes including user, role, application, database, query tag, and others. By default, data loading operations are subject to the same priority and scheduling rules as regular queries. This unified approach ensures that neither data loads nor individual queries monopolize system resources.

Queries can be assigned different priorities, throttled, and automatically cancelled and restarted within a different pool if they exceed given limits. Pools can be configured to allow mixed workloads (e.g. data loads and queries) to run on the same compute cluster without the need to manually partition workloads across different clusters. We have measured rates as high as 20,000 queries per second through our workload management system.

Yellowbrick's default workload management policies are designed to balance competing priorities out of the box. The ability to tailor policies means that you can deliver the best outcomes for your specific application, guarantee data latency and query performance SLAs, and balance costs.

## Hybrid Storage Engine



Most modern data warehouse implementations are backed by column stores alone. While this approach can result in high data compression and good query performance, their ability to support single record and micro-batch loads efficiently is compromised. Yellowbrick features a hybrid storage engine design that combines a front-end row store and a back-end column store that allows data to be inserted, bulk loaded and queried simultaneously without impacting data latency. The hybrid storage engine is completely transparent to users and applications, requiring no changes or maintenance.

The data in a Yellowbrick table spans both the row store and column store and, from the perspective of a query or data load, appears as a single logical table. Data can be inserted into the row store on a record-by-record basis at high speed and is instantly accessible. Rows are automatically flushed into the column store over time.

Bulk loads of large amounts of data are inserted directly into the column store via parallel connections to the workers, bypassing the row store. Transactions are managed across the row and column store by using a common transaction log with a "read committed" level of isolation and multi-version concurrency control (MVCC), ensuring consistency and durability for both streamed and stored data.