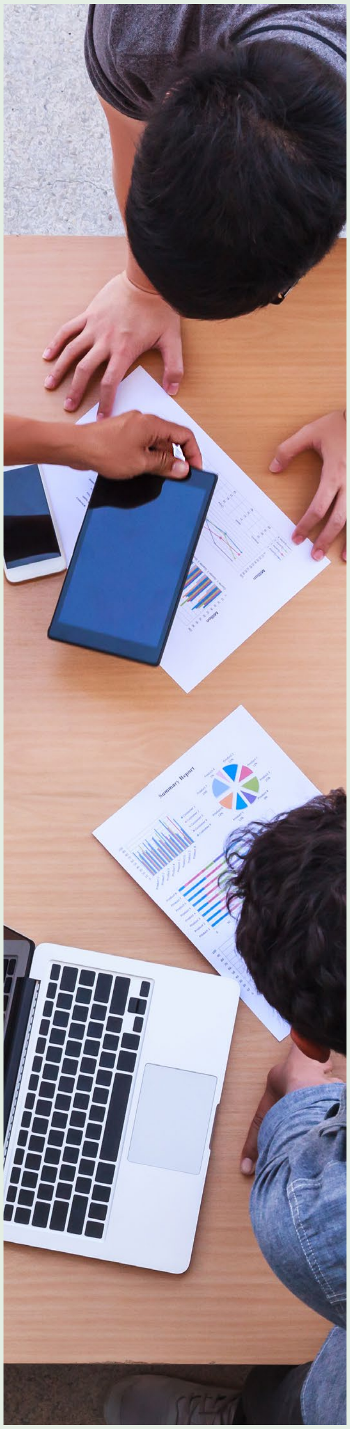
WHITE PAPER



THE MODERN DATA ESTATE

Table of the content

1 Data as a Transformative Asset

"In God we trust, all others must bring data." – W. Edwards Deming

After decades of languishing as the exhaust from the business processes, in the last decade "data" has earned its rightful place as an asset of the business. A 2020 study found that data-driven organizations were 162 percent more likely to exceed revenue targets than others. Other estimates may be more, or less, optimistic, but all agree that data-native organizations outperform those that are not, by a long margin.

Successful data-native organizations foster a **culture of data** to compete, differentiate and responsibly leverage the innate power of data and analytics. With modern design patterns for managing and utilizing data in cloud ecosystems, experts predict that, in the coming months, 75 percent of centralized analytics initiatives will make way for a federated model where local business domain analytics units will come to share authority and decision-making with the central team.

However, for the most part, in this "data economy", **data is a loss-leading asset**, meaning that while it is invaluable to conducting and competing in business, the value of data is not directly represented in a balance sheet, unless the business itself is in the business of selling data i.e. third party data, insights from sensors, etc.

To optimize a loss-leader asset like data, today's data-native enterprise should enable direct data access to every part of the organization, or otherwise **democratize data**.

The method to enable this for data, which still needs to be a governed corporate asset, is to enable a new kind of architectural pattern, called "data-mesh" and treat data assets as **"data as a product"** (DaaP), with defined ownership and governance structures. This is a paradigm shift from the legacy data governance methods using data stewards managing data centrally in legacy data-hub-based data management structures like data lakes and data warehouses. In the world of **"data as a product"**, the "stewards" are replaced with "data-product-owners" and a centralized governance gives way to **"federated governance"** methods.

The following thesis discusses the construct of treating data as a product, the organizational and technology blueprint necessary to drive this and the implementation method using an innovative decentralized architecture that leverages an open-source codebase to accelerate the time to market by 2x with no degradation of performance metrics.

2 Data as a Business

Whether data is used by internal consumers or to create monetizable assets, it needs to be managed and governed like a product - that has a clear addressable market and a defined path for success.

2.1 Making it easy to do Business with Data

There are four tenets to consider for enabling a data-product organization.

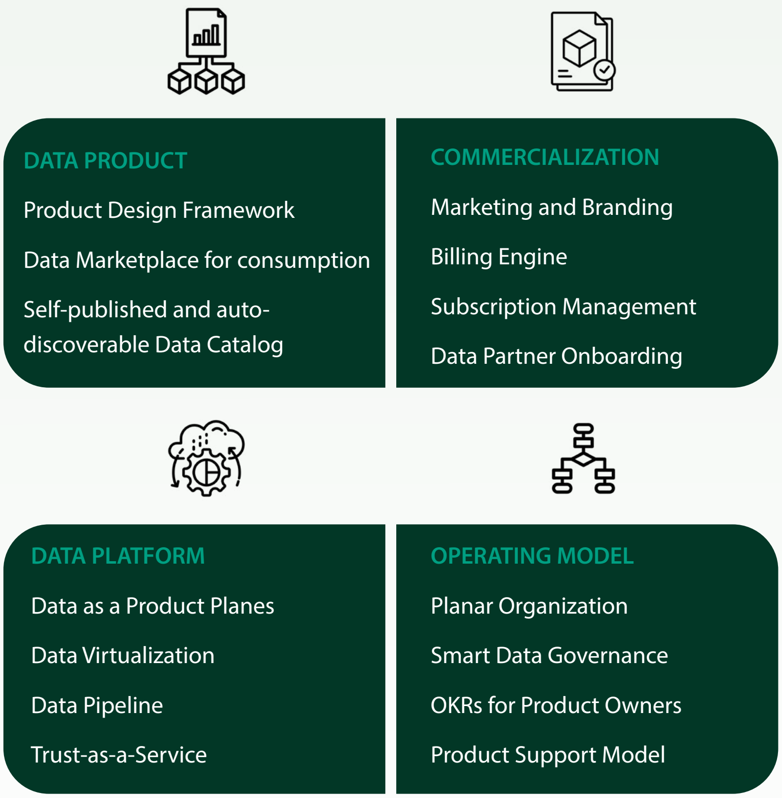
2.1.1 Data Product(s)

Data products are defined by domains, typically within Business Units. Data Product Owners have end-to-end responsibility for defining, defending, and leading "manufacture' of the data product.

The design of the "data product platform" (see below) ensures that there is semantic consonance and content arbitration amongst the data-products. The data products will need a "marketplace" that allows them to be "discovered" by potential consumers.

2.1.2 Data Product Platform (DPP)

This is the "mall" for the data-products. It is the core platform that provides capabilities to people who manage the data content, but also enables the development of data products. It is crucial for the platform to have capabilities for observability and tracking data product usage. It needs to have provisions to virtualize data from multiple sources, organize data products into "planes", manage access controls, track lineage, and instill trust on the content that flows through the platform.



The ecosystem of a Data Product

2.1.3 Operating Model

The legacy data governance structures would need to be restructured to create a federated structure where content ownership lies with the Data Product Owners (DPO). As the DPOs will be business owners, their performance would be measured by business OKRs that have a monetary value. In addition, AI and Machine Learning can be used to auto fix data quality and integrity, auto discover lineage, manage data privacy, and master data, without the need for human data stewards and data engineers. This also lessens the need to have a central data organization of any kind.

2.1.4 Commercialization

The data product organization will not be successful if the mechanics are not put in place for monetization. One of the key components would be to build a "billing engine" that can track not only usage and consumption of the data products, but also the "cost" of sourcing the attributes that make the data product, or the COGS. This function would need to be able to manage "subscription" and licensing of the data products and manage the onboarding and registration of internal and external data product consumers and partners.

2.2 Glidepath to Meshifying

Most organizations are steeped in centrally governed legacy hub-based model, centered around integrating, and curating data from "source systems" to legacy constructs like data lake or data warehouse, using complex and expensive data pipelines, and then eventually farming it out to consumers using yet another pipeline through data marts or a services network.

To dismantle this historic monolithic data infrastructure and germinate a modern data estate leveraging data-as-a-product pattern, here are the five sequential logical steps that should be considered.

The Maturity Journey for realizing the potential for Data as a Product

2.2.1 Virtualize

The first step would be to virtualize the existing data hub and connect it with data that is not yet in the hub or data-debt. This enables a quick win for the consumers and lays the foundation of a data-mesh pattern, by treating the hub as one of the **data-nodes** in the **Analytic Plane.**

2.2.2 Advertise

Once a sufficient *(Pareto rule – 80%)* consumable data is virtualized, the next step would be showcasing the data assets that are available for consumption. A **marketplace** design pattern, built on a self-published auto discoverable data catalog allows consumers to search for their data assets, by researching the data-products.

2.2.3 Productize

Once the foundations of virtualization and marketplace are in place, a full-fledged data-mesh platform can be unleashed. This is the stage where data products are appropriately defined, cataloged, and arranged in the **"planes"** that need to be inter-connected. At this stage the **data-product-owners** are identified, roles defined and the commercial OKRs agreed upon. Depending on the maturity of an organization, this could be the longest pole in the Data-as-a-Product tent.

2.2.4 Rationalize

As the Data Product Platform gets hydrated with data-products and adopted by the data consumers, conflicts across content and semantics would arise. Instead of the legacy approach of commissioning a team to create a common-semantic-model or an enterprise data model, define an **ontology** trained from the enterprise corpus and define a **knowledge-graph** where all the data-products are organized. Leveraging entity-resolution techniques, a consuming application can navigate the graph to get to the right data-product and its associated content.

In a more advanced setting, one can also build a **large-language-model** and train it with the data-product corpus, thereby enabling a conversational access to information, across all the data-products.

2.2.5 Commercialize

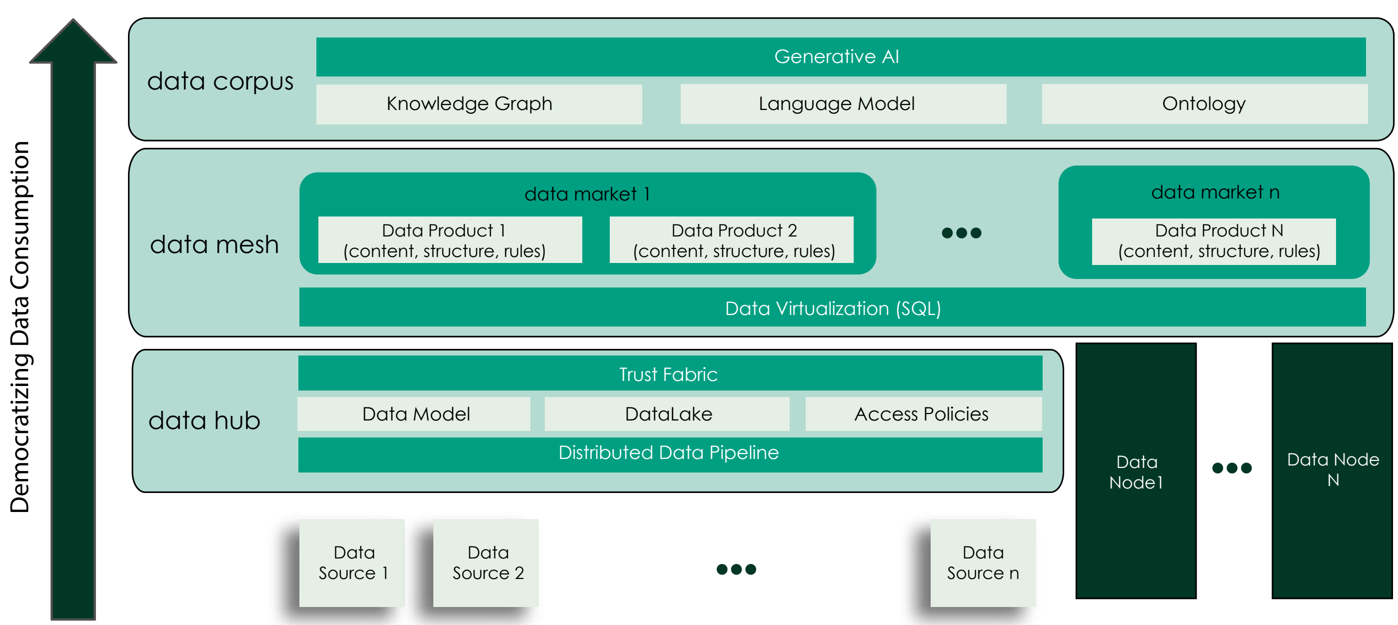
The last step in maturity is to truly transform data as a business. The core to that is to build and deploy a **billing-engine** that can metric the **cost of an attribute** to create a cost-model for data product and then apply a **pricing-model** based on factors such as consumption patterns, data-product reputation, and business impact.

3 Data as a solution

3.1 Re-imagining the Erstwhile Data Platform

The blueprint of a data platform that leverages data-mesh is geared towards a changed consumption pattern, where today's sophisticated consumers seldom look for charts, graphs and tables, but seek specific answers to granular queries.

In a next generation democratized data architecture, there are 3 patterns of design that are layered to provide the capabilities to data product owners to package and socialize their data.



Re-imaging the Erstwhile Data Platform – The blueprint for the future

3.1.1 Data Hub

This is the persistent layer where the data physically resides. Legacy data-lakes, data marts and data warehouses can be subsumed in this design pattern. In addition to authorized and curated data sources like these expensive assets, any other data source that is useful to the business, but which hasn't made its way through the central hubs, like spreadsheets, documents etc. can also be part of this tier.

For the functioning of the data-as-a-product design pattern, all these data reservoirs need to be encapsulated as a "data-node". It essentially means that they have a common interface, ideally RESTful, where they can serve up three (3) things: its content, the structure of this content and the rules that create the content, ideally as a JSON document.

3.1.2 Data Mesh

This tier is the key to enable the federated pattern of data-as-a-product work. The underlying technology and design in this enable **"virtualization"** of the content in the data nodes, in a ultra performant design pattern called MPP (massively-parallel-processing) without creating any persistence. It allows for the content, the structure, and the rules to be exchanged for interpretation and subsequent use by the data consumers. It also enables a **"marketplace"** where consumers can browse, shop, combine and "purchase" the data from the "data-product-owners" for their own use.

3.1.3 Data Corpus

The data consumption pattern in the next generation data platform, undergoes another tectonic shift, as consumers of content evolve into a more sophisticated interaction model, where information is delivered via a conversation, rather than charts, graphs and sheets. A **"knowledge graph"** that is built on an **"ontology"** which is bespoke to an organization's semantic understanding of the business serves as a guide to this Q&A type information consumption. Pre-trained large language models **("LLM")** that are primed with bespoke prompts or **"prompt engineering"** become the dominant design pattern. To enable this corpus on enterprise data, the underlying "data-mesh" tier becomes the foundation on how these LLMs, graphs and ontologies are constructed.

3.2 Pathway to Data Mesh Realization

The implementation journey for a practical data-mesh follows the pathway described in Section 2.2.

3.2.1 Product Increment 1: Virtualization Platform

To achieve the solution we describe, the initial step is to virtualize the present data hub and link it with data that is not yet in the hub or data-debt. This action provides a quick benefit to the consumers and forms the groundwork of a data-mesh pattern.

3.2.2 Product Increment 2: Marketplace Platform

Once enough consumable data is virtualized, the subsequent step is to exhibit the accessible data assets for consumption. This is done through a marketplace design pattern that depends on a self-published auto-discoverable data catalog, allowing consumers to explore their data assets by analyzing the data-products.

3.2.3 Product Increment 3: Product Platform

Once the virtualization and marketplace fundamentals are in place, a decentralized and democratized data as a product platform can be launched. This is the stage where data products are adequately defined, cataloged, organized, and must be interconnected.

3.3 Implementation Best Practice – Trino and Starburst

Trino is an open-source, massively parallel processing (MPP), distributed ANSI SQL analytics engine deployed at exabyte scale at the largest internet companies in the world (Facebook, Netflix, LinkedIn, etc). Originally created at Facebook in 2012 as Presto, and rebranded in 2020, Trino is designed to perform analytics on large volumes of distributed data. Trino's architecture separates data storage and data computation, and virtualizes data with performance, scalability, and flexibility for modern large-scale data processing needs.

Starburst is an enterprise-grade solution built on top of Trino. Starburst extends core Trino capabilities and provides a range of tools and technologies that are essential in the creation of Data Products. Data products are the result of analytical queries that connect to multiple data sources, aggregating and analyzing those sources to create usable federated datasets. Starburst tools and technologies support the deployment of Data Products in three ways:

- By providing fast and efficient access to data, advanced analytics and visualization capabilities, Data Products can be quickly defined and evaluated.
- By providing a scalable and flexible platform for running analytics workloads Data Products can be fully developed on the platform
- By providing responsive access to multiple data sources in near real time allows Data Products to live on the platform, using virtualized source data, thus avoiding unnecessary duplication.

Starburst consists of a data consumption layer that sits between data sources on one end, and existing analytics tools on the other. We have leveraged Starburst with our clients' existing technology investments to run BI workloads, build pipelines, and run AI/ML workloads using analytics tools and languages, including Power BI, Tableau, Looker, ThoughtSpot, Superset, dbt, Metabase, Python, Jupyter, and R, as well as other tools with standard ODBC/JDBC.

Starburst includes additional proprietary data source connectors. The source data itself is never actually stored within the platform. It is available in Amazon, Google, and Azure ecosystems as marketplace offerings or as direct Helm chart deployments in their respective Kubernetes services.

Starburst offers query acceleration that can both index and cache portions of the data lake and can lower query execution time on the lake The autonomous indexing and smart caching enables higher performance for data catalogs using the Apache Hive, Apache Iceberg, or Delta Lake connectors by transparently adding an indexing and caching layer.

Starburst provides fine-grained built-in native role-based access control (RBAC) for data sources, and audit logs and query logging capabilities, from which details of query history, including single-query statistics and query plans, can be accessed.

Fault-tolerant execution is a mechanism in Trino that enables a cluster to mitigate query failures by retrying queries or their component tasks in the event of failure. With fault-tolerant execution enabled, intermediate exchange data is spooled and can be reused in the event of an outage or other fault during query execution. Fault-tolerant execution allows Starburst to run interactive and ELT workloads in the same query engine and platform.

Additionally, Starburst allows for cross-cloud connectivity for access to data across disparate clouds, regions, and ground-to-cloud. The cluster-to-cluster connectivity enhances data products to become a gateway for data access across geographies while ensuring access controls are in place, and data residency requirements are honored.

Data Products built on this platform provide organizations with a self-service platform capable of employing the data-as-a-product methodology. This approach allows teams to create and share a curated, end-to-end self-service solution, or to mix that solution with other tools across the wider data ecosystem.

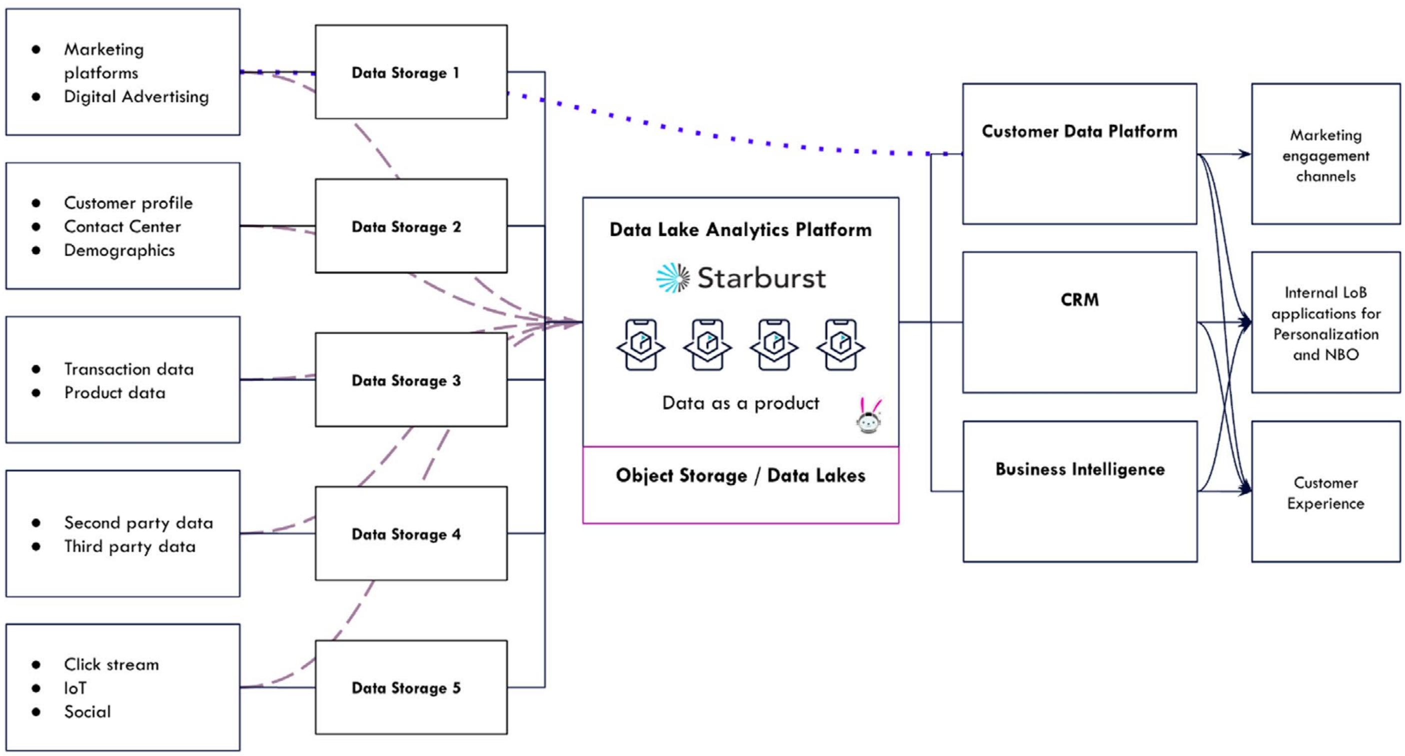


Figure 1: Example of A Data Product Ecosystem with Starburst for a Customer 360 Implementation

4 Evaluating the Solution Patterns

To explore the efficacy of alternative approaches and designs of managing and consuming information in a modern data-estate, we evaluated four (4) different data federation ecosystems for performance, efficacy and cost. The benchmarking exercise tested the performance of the data federation capabilities of a data ecosystem with **Starburst Enterprise** (SEP) against the data ecosystem with **Dremio, Databricks,** and **Snowflake.**

It should be noted that each of these products have somewhat different objectives and feature functionality compared to Starburst. These benchmarks only relate to the performance of federated queries by the different platforms.

4.1 Benchmark Setup

4.1.1 Ecosystem X: Starburst

Starburst was deployed on an Elastic Kubernetes Service with cluster consisting of 10 (1 coordinator and 9 workers) r6i.8xlarge nodes. Each node has 32 vCPUs, 256GiB of memory, and no storage. Out of the 256GiB of memory, 236GiB was given to the workers, and around 9% was reserved. In total we have 2560GiB of memory and 320 vCPUs. Data Connectivity - to connect to the tables in Postgres and ElasticSearch we included the necessary connector details in the catalogs.yaml file.

4.1.2 Ecosystem A: Dremio

Software was deployed from the AWS Marketplace using a cloud formation stack. This Dremio cluster consists of 9 nodes (1 master and 8 workers). We used recommended configurations.

4.1.3 Ecosystem B: Databricks

An All-Purpose Cluster comprised of one xlarge (Driver) and 32 xlarge (workers). The driver consists of 64 vCPUs, 488GiB of memory, and 15.2TB of NVMe SSD storage. While the worker instance consists of 8 vCPUs, 61GiB of memory, and 1.9TB of NVMe SSD storage.

4.1.4 Ecosystem C: SnowFlake

Though this product serves as a warehouse for its own data, they can connect to popular cloud storages like Amazon S3, Google Cloud Storage, and Microsoft Azure using External Tables. For this benchmark, the X-Large cluster was chosen.

4.1.5 Benchmark data

The benchmark data was the TPC-H data with a scale factor of 1000 (1TB of raw data) and 22 standard queries. This data was distributed across an Amazon S3 object storage, a Postgres RDS instance, and ElasticSearch (Amazon OpenSearch Service), all within the same AWS VPC. The data was stored in AWS S3 using three different table formats, file formats, and compressions as seen below.

Table Format	File Format	Compression
Hive	TextFile	NONE
Delta-Lake	Parquet	SNAPPY
Iceberg	ORC	GZIP

4.1.6 Benchmark Queries

For our benchmarking, the objective was to federate the 22 standard TPC-H queries, and to measure and compare the performance of these queries by the different platforms. This is achieved by sourcing data sitting in different tables, table formats, and across different systems.

The table below represents the 22 queries that was run for this benchmarking exercise.

Queries	Data source/table combination
q01	Data source/table combination
q02	Part, supplier, and partsupp in Delta Lake Lake. Nation and region is PostgreSQL
q03	All tables in Iceberg
q04	All tables in Delta Lake
q05	Customer, orders, lineitem, supplier in Delta Lake. Nation and region in PostgreSQL
q06	Delta Lake
q07	Lineitem, orders, customer in Delta Lake. Supplier and nation in PostgreSQL
q08	Part, supplier, lineitem, orders, customer in Delta Lake. Nation and region in ElasticSearch
q09	All Iceberg
q010	Lineitem, orders, customer in TextFile. Nation in ElasticSearch
q011	Partsupp in Iceberg. Supplier and nation in PostgreSQL
q012	All Delta Lake
q013	All Delta Lake
q014	All TextFile
q015	Lineitem in Delta Lake. Supplier in PostgreSQL
q016	Partsupp, part in TextFile. Supplier in PostgreSQL
q017	Part in Delta Lake. Lineitem in TextFile
q018	Lineitem in Delta Lake. Orders in TextFile. Customer in Iceberg
q019	All Iceberg
q020	Lineitem in TextFile. Partsupp in Iceberg. Part in Delta Lake. Supplier and nation in PostgreSQL
q021	Lineitem, orders in Delta Lake. Supplier and nation in PostgreSQL
q022	All Delta Lake

4.2 Benchmark Outcomes

All 22 federated standard TPC-H queries were run consecutively from the various platforms 3 times each using JMeter.

The table below shows the result averages for each query taken over three (3) runs.

	Queries completed	Average Time	Minimum Time	Maximum Time
Ecosystem X	**22**	**042.41s**	05.04s	**142.76s**
Ecosystem A	21	092.63s	07.73s	450.06s
Ecosystem B	16	066.19s	**04.02s**	242.00s
Ecosystem C	8	173.65s	27.75s	551.34s

Ecosystem X: Starburst was the only one that successfully completed all of 22 queries, whereas in other ecosystems, the successful completion of queries was checkered.

Total compute cost calculations

Benchmark Execution Cost Formula: $\dfrac{(\textit{Compute Cost/hr})}{(\textit{60 mins/hr})} \times \textit{Total Runtime (mins)}$

Average cost per query divided the benchmark execution cost by the # queries completed.

	Cost per Query	Cost per Hour	Execution Cost
Ecosystem X	**$ 1.17**	$ 32.96	**$ 25.63**
Ecosystem A	$ 4.63	$ 41.90	$ 67.92
Ecosystem B	$ 2.35	$ 42.56	$ 37.56
Ecosystem C	$ 3.23	**$ 32.00**	$ 37.05

4.3 Recommendation

Analyzing the price/performance results of these ecosystems and evaluating for the most effective implementation of the data-mesh design pattern, it was evident that the MPP non-persistent design of Trino in general and Starburst is the most viable, performant, and cost-effective solution for a distributed and federated data estate.

5 Data as a Driver for AI

Many enterprises are engaged in democratizing data to imbue the organization with data-centric and data-aware operating model, that creates a pedestal for scalable, ethical and transformative AI for doing business in the modern decade. Adopting the design pattern for Data Products, leveraging a federated Data-Mesh Architecture, as discussed in this thesis, enterprises can accelerate time to market Data and AI solutions, at lower cost, and with no degradation of performance when compared to other solution patterns.

As new interaction models evolve between data producers and data consumers (whether human or machine) it is vital that data architectures provide maximal flexibility, and the ability to rapidly develop Data Products responsive to new needs.

About the Authors

GARY BHATTACHARJEE,

DATA AND AI STRATEGY LEADER, INFOSYS.

DANIEL BRAULT,

DIRECTOR PRODUCT MARKETING, STARBURST.

Infosys Topaz is an AI-first set of services, solutions and platforms using generative AI technologies. It amplifies the potential of humans, enterprises and communities to create value. With 12,000+ AI use cases, 150+ pre-trained AI models, 10+ AI platforms steered by AI-first specialists and data strategists, and a 'responsible by design' approach, Infosys Topaz helps enterprises accelerate growth, unlock efficiencies at scale and build connected ecosystems.

For more information, contact askus@infosys.com

Infosys® Navigate your next