WHITE PAPER

Shortening Time to Value by Building Data Products in a Lakehouse

By Mike Ferguson
Intelligent Business Strategies
October 2022

Research sponsored by:

Table of Contents

Copyright © Intelligent Business Strategies Limited, 2022, All Rights Reserved

WHAT IS DATA MESH?

Data Mesh is a decentralised approach to producing trusted, reusable datasets know as data products

One of the hottest topics in data architecture at the moment is Data Mesh. This is a data architecture approach originally defined in an article[1] published in 2019. Since then, Data Mesh has grown in popularity to the point where there is global interest with many companies now implementing it. The Data Mesh concept is a decentralised business domain-oriented approach to data engineering that results in the production of trusted, reusable datasets. These datasets are referred to as "data products" that can be shared and consumed across the enterprise and beyond. The main purpose is to use these data products in multiple analytical workloads such as in data science and in data warehouses.

There are four major principles defined in Data Mesh. These are:

- Domain oriented decentralised data ownership and architecture
- Data as a product
- Self-serve data infrastructure as a platform
- Federated computational data governance

Business domain subject matter experts use self-service tools to create pipelines that produce data products

The main objective behind Data Mesh is to enable people in different business domains, who work with specific data every day, to use self-service infrastructure software to create data pipelines that produce data products to share and reuse across the enterprise. The intention is that business professionals, who work with specific domain application data every day, take responsibility for creating data pipelines to capture data from domain-specific data sources and make it fit for purpose for use in various kinds of analyses both within and outside their business domain. This is shown in Figure 1.

Each domain creates and owns data products to make available for sharing and consumption elsewhere in the business

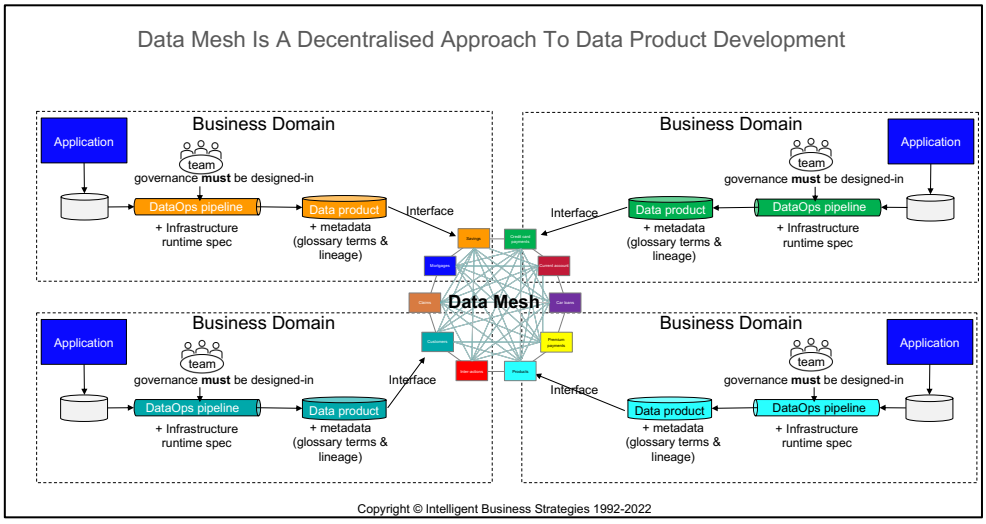


Figure 1

The intention of the Data Mesh approach is to accelerate the creation of high-quality, compliant data to share by upskilling business professionals to produce this data rather than relying on a centralised IT team who are unable to keep pace with business demand. Expert IT professionals can be embedded in business domains to help make this possible.

[1] How to Move Beyond a Monolithic Data Lake to a Distributed Data Mesh, Zhamak Dehghani, May 2019

Copyright © Intelligent Business Strategies Limited, 2022, All Rights Reserved

WHAT PROBLEM IS DATA MESH TRYING TO ADDRESS?

You may be wondering why Data Mesh is needed. What is wrong with existing approaches to data engineering? Exactly what problem is Data Mesh trying to address? To explain this, we need to understand what is happening with data.

Over the last ten years, many new data sources have emerged both inside and outside the enterprise that companies now want to analyse. This includes human generated data such as web chat, inbound emails, voice data, images, video, and social network data. In addition, machine generated data is in demand such as on-line clickstream data generated as people browse your website, IoT data and infrastructure log data. So, it has gone well beyond traditional structured data in transaction databases which is also heavily used.

All of this data is being created and ingested into multiple different data stores that are both on-premises and in multiple clouds. Data is also stored in software-as-a-service (SaaS) transaction processing applications hosted in various application vendor sites and streaming in from IoT devices at the edge. All this new data has resulted in many companies now running multiple different analytical workloads analysing overlapping subsets of this data on different analytical systems both on premises and in the cloud (see Figure 2).

Many companies now have different analytical workloads analysing overlapping subsets of data running on different centralised analytical systems

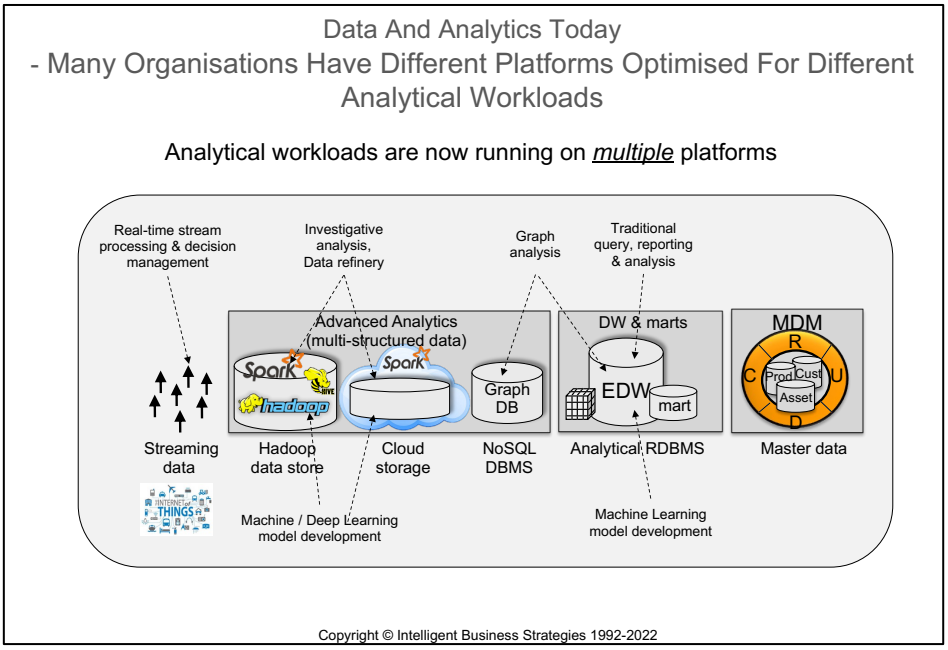


Figure 2

Centralised IT-based data engineers are becoming a bottleneck as the demand to analyse new data grows

Most of these analytical systems require data to be cleaned, transformed, integrated, by central teams of data engineers. However, as the number of new data sources grows, centralised data engineers, whether they be IT professionals or data scientists, are already being viewed as a bottleneck and often have limited knowledge of business domain data sources.

In addition, the analytical systems shown in Figure 2 are siloed. Therefore, they typically operate independently with different data engineers using different data integration tools to capture, clean and integrate data for loading into each siloed analytical system in support of different analytical workloads (Figure 3).

Copyright © Intelligent Business Strategies Limited, 2022, All Rights Reserved

Each different analytical system is operating as an independent silo

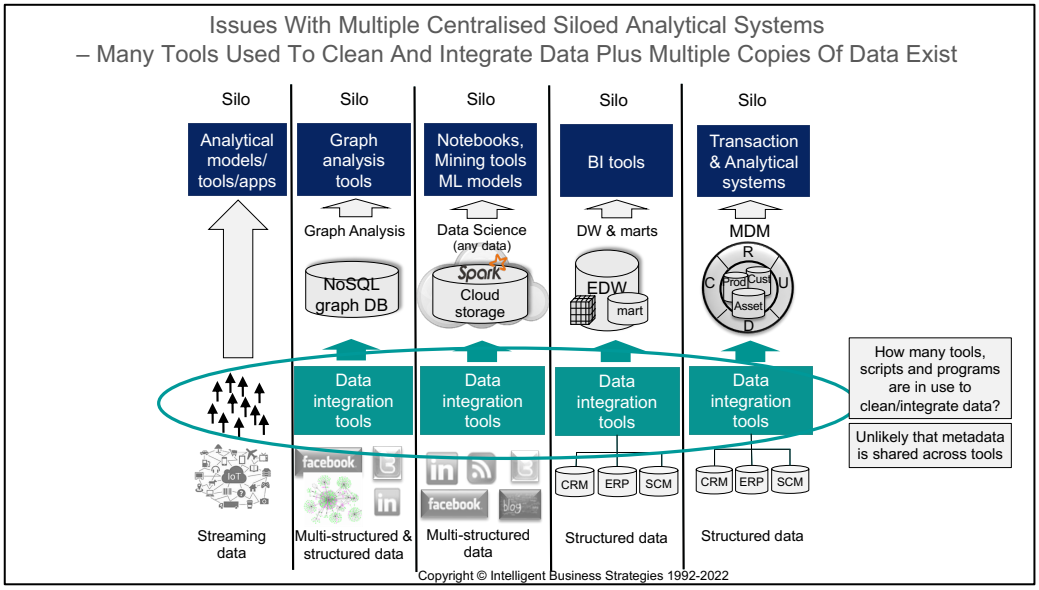


Figure 3

Siloed analytical systems increase data integration costs and cause reinvention which can lead to inconsistent data

In addition, the same data is often repeatedly extracted, cleaned, transformed, and integrated for different analytical systems as shown in the insurance example in Figure 4.

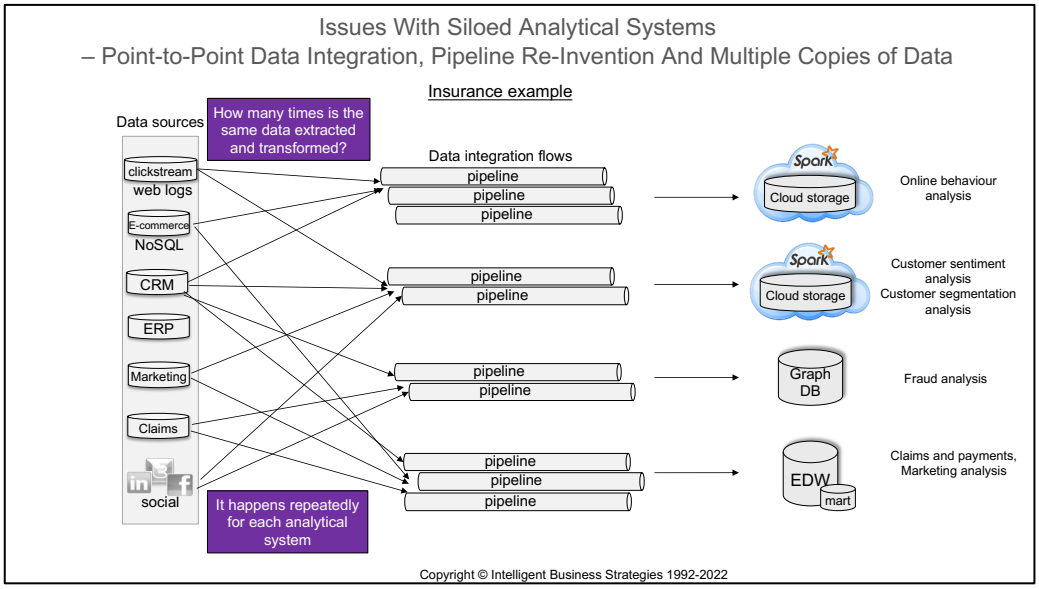


Figure 4

The same data is often repeatedly extracted transformed and integrated for different analytical workloads in different analytical systems

It would be faster and less costly to do it once and create reusable 'data products' for reuse in support of different analytical workloads

A stronger approach would be to create trusted, compliant "data products" once and reuse them in different analytical workloads rather than repeatedly re-inventing data integration pipelines to create that same data for each different analytical system over and over again.

Data Mesh is intended to solve this problem while also speeding up the rate at which data products can be created by enabling different teams in different business domains to produce historical, compliant data for consumption in different analytical workloads being implemented in other parts of the enterprise.

Also, the same analytical system could support multiple different analytical workloads

But could we not go further? For example, why can't data products be created once and be utilised in the *same* analytical system to support *multiple* analytical workloads (e.g., BI and data science) without having to create multiple instances of data products in different downstream analytical systems?

Copyright © Intelligent Business Strategies Limited, 2022, All Rights Reserved

WHAT ARE DATA PRODUCTS?

Data products are reusable data sets that can be consumed for use in different analytical workloads

At the centre of the Data Mesh approach are data products. These are typically based on business data concepts. Figure 5 shows an example of data products in Insurance which could include Customers, Products, Quotes, Agreements (policies), Premium Payments, Claims etc. The idea is that data integration pipelines are created which, when executed will produce these kinds of data products using data from one or more data sources. Data sources can include structured, semi-structured and unstructured data. It is also possible to execute machine learning (ML) models in pipelines to derive structured data from unstructured sources to include as part of a data product. Once data products are produced, they are then published in a data marketplace[2] to make them available for consumption by different analytical teams for use in different types of analytical workloads. This includes for use in machine learning model development in data science, graph analysis (e.g., for insurance fraud), and in data warehouse tables supporting business analysts using business intelligence tools. The objective is to build data products once and reuse them everywhere.

Build data products once and reuse them everywhere

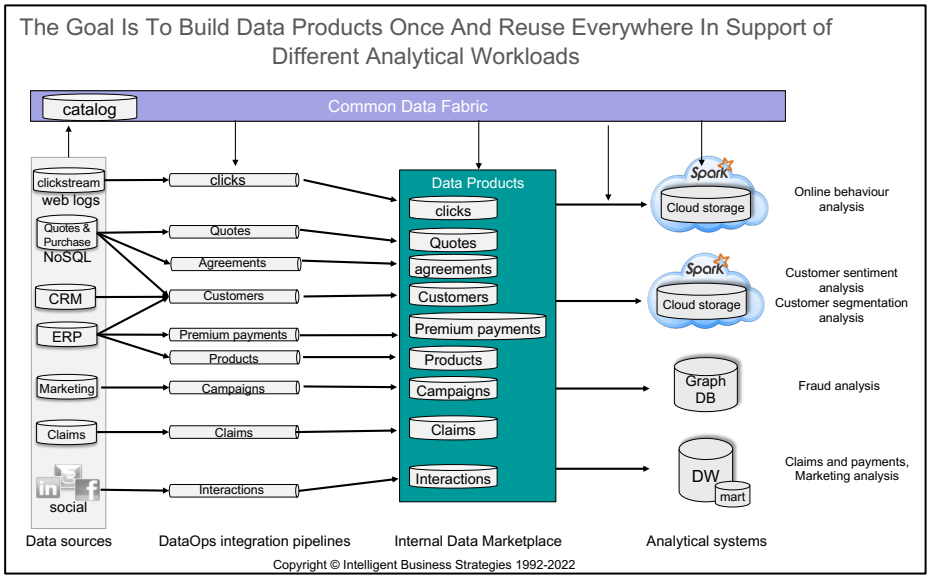


Figure 5

Data products should be high quality, compliant, and secure and be published in a data marketplace with metadata and APIs to make them easy to find, consume and use

Data products are defined[3] as having the following characteristics:

- Discoverable
- Addressable
- Trustworthy
- Self-describing
- Interoperable
- Secure

In addition, the data product includes the data itself, the pipeline (rules to clean and integrate data), the runtime specification to execute the pipeline and APIs.

---

[2] A data marketplace is a catalog application that provides a place where people can 'shop for data'

[3] How to Move Beyond a Monolithic Data Lake to a Distributed Data Mesh, Zhamak Dehghani, May 2019

Copyright © Intelligent Business Strategies Limited, 2022, All Rights Reserved

TYPES OF DATA PRODUCT IN A DATA MESH

Different types of data products can exist in a Data Mesh including:

- Physical data products
- Virtual data products
- Stored queries

Physical data products are persisted datasets that have been produced, stored, and published in a data marketplace to make them available for consumption.

Data products can be persisted, virtual or stored queries

Virtual data products are virtual views that integrate data from one or more underlying data sources (including ready-made physical data products) on-demand, on a timer driven basis or on a continuous basis. They can also be materialised for performance. Again, they can be published in a data marketplace for people to discover, query and use.

Stored queries are typically SQL queries that can be published as services and invoked on demand e.g., via a REST or GraphQL API. When executed, the stored query will then produce a data product and serve it. Stored queries can integrate data from one or more data sources including other data products.

CONSUMING DATA PRODUCTS TO CREATE NEW ONES

Time to insights should get progressively shorter as more data products become available for reuse

Once created, it is possible for people to consume data products in another pipeline and integrate this data with other data to produce a new data product. This new data product can itself then be added to the Data Mesh. In that sense Data Mesh is a kind of bootstrapping capability where more and more data products are incrementally built over time and made available for others to consume and use. Therefore, as more data products become available, the time taken to deliver insights should get shorter because each new project doesn't have to start from scratch to create data.

WHAT ARE ANALYTICAL PRODUCTS?

Analytical products such as BI reports, dashboards, predictive and prescriptive models, are built using data from consumed data products

Some might argue that there are many other types of data products. What about BI reports, dashboards, stories that string together multiple dashboards? What about predictive machine learning models, prescriptive machine learning models (e.g., recommenders, alerters), plans and AI-automation bots? Are they not also data products? I would prefer to call these types of artefacts *analytical products* that are built to analyse data consumed from one or more data products. Once built, these analytical products can also be published in the data marketplace as analytical services available for consumption and use by other users and applications. Much like data products, the objective again is to encourage reuse to shorten time to value.

ORGANISATIONAL IMPACT OF DOMAIN BASED DATA PRODUCT DEVELOPMENT

Data Mesh has a major impact on IT organisation

A federated set-up with a program office, a centre of excellence and experts embedded in domain data engineering teams is good practice

Domain-based data product development has a major impact on IT organisation. This is because IT needs to move away from a central team of data engineers to a federated organisation with a CDO controlled program office and a centre of excellence that embeds expert data engineers into domain-based citizen data engineering teams to help upskill them and to ensure they adopt best practices. The program office helps co-ordinate data product development across the business to avoid chaos and reinvention cause by people having no knowledge of who is producing what data and for what purpose.

Copyright © Intelligent Business Strategies Limited, 2022, All Rights Reserved

IMPLEMENTING A DATA MESH USING A LAKEHOUSE

A Data Mesh can be built on a lakehouse to reduce development costs and to integrate analytical workloads

Having understood what Data Mesh is and its purpose, the next question is, how can you build one to produce high quality compliant data more rapidly while avoiding the problems of siloed analytical systems depicted earlier in Figures 2, 3 and 4? One way to achieve this is to implement a Data Mesh on a Data Lakehouse to remove the need to copy data to multiple analytical systems and integrate multiple analytical workloads. To understand how to do this first requires us to define what a lakehouse is and what it needs to support to build a data mesh. Then we can look at how to build a Data Mesh to support multiple analytical workloads all on the same platform.

LAKEHOUSE PLATFORM REQUIREMENTS TO SUPPORT A DATA MESH

Building a data mesh on a lakehouse requires support for multiple capabilities

A data lakehouse is a single or multi-tenant platform that aims to offer the best of both a data warehouse and a data lake to support the integration of analytical workloads like data science and business intelligence (BI) on one set of data. The data in a lakehouse is typically persisted in cloud storage.

To build a data mesh on a lakehouse, requires a lakehouse platform to support:

- Connectivity to multiple data sources on-premises and in the cloud
- Structured, semi-structured and unstructured data types

Ability to support stages ETL processing on any type of data

- The ability to create tables to organise and support staged ELT processing using data pipelines to enable creation of data products
- A data catalog to understand source data and govern all data being ingested, processed, and made available in the lakehouse for sharing, query processing and analysis
- Automated schema validation and governance to ensure data being produced, consumed, and accessed matches the published schema

Schema validation, version control and data integrity

- Automated schema version control to enable new versions of data products to be created
- Guaranteed transaction properties[4] to ensure consistent views of data even if it is being inserted / updated in real-time

Automated scalability when processing data

- Change data capture from data sources
- Scalable ETL and ML processing on any type of data at any size

Data quality on tables that support both streaming and batch

- Support for policies and rules to guarantee high quality data in tables
- Unification of streaming and batch on tables so any table can continuously receive new data in real-time to keep it as fresh as possible while also enabling historical analysis
- Store and process data in an open, columnar, file format (e.g., Parquet) for performance and to enable use by multiple technologies
- Automated support for historical analysis back to and from specific points in time

[4] Often referred to as ACID properties – Atomicity, Consistency, Isolation, Durability

Copyright © Intelligent Business Strategies Limited, 2022, All Rights Reserved

Scalable SQL query processing and support for machine learning

- A high-performance query engine for concurrent SQL access to data by business analysts and data scientists from a range of client tools
- Machine learning lifecycle and integration with scalable engines like Apache Spark for access via other languages such as Python
- Row and preferably attribute level security to govern access to data

Ability to securely share data products and publish them in a marketplace

- Provide a data marketplace to allow data producers to publish new data products in a data mesh and to allow data consumers to easily find and consume business ready data for use in their analyses
- Securely share business ready data across multiple analytical workloads in the lakehouse without the need for data replication
- Securely share business ready analytical products (e.g., machine learning models) to enable them to be consumed and used by other applications and business users

DOMAIN ORIENTED DEVELOPMENT OF DATA PRODUCTS USING A LAKEHOUSE

Having defined lakehouse technology requirements, the steps involved in domain-oriented development on a lakehouse are described as follows.

Lakehouse Setup

The lakehouse needs to be organised to enable multiple teams of data producers to create data products

The Data Lakehouse can be set-up and organised to enable decentralised data product development and data product consumption for use in multiple analytical workloads. Figure 6 shows how a lakehouse can be organised to produce and consume data products using three types of tables. These are bronze, silver and gold tables. The objective of this is to use a staged ETL approach to support the creation of high-quality data products in lakehouse silver tables with gold tables being the data mesh consumption layer. Note that one or more lakehouses could be created this way on the same lakehouse platform in the same organisation.

A staged ETL approach to creating data products in lakehouse silver tables and publishing them in a data marketplace

Data products can be shared and consumed to support data warehouse and data science on one platform

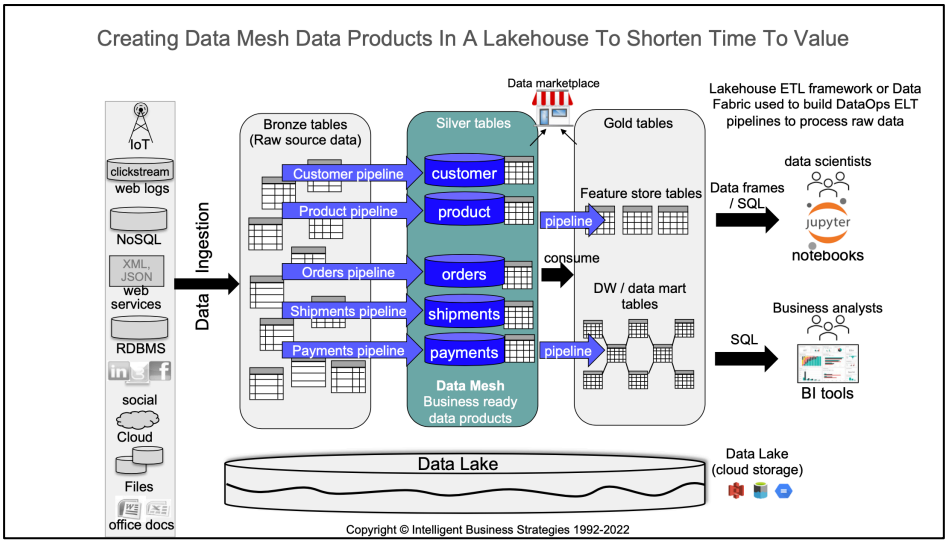


Figure 6

Business alignment

The first step in building a data mesh on a data lakehouse is to determine what data products are needed to support decisions that will help achieve one or more high priority target outcomes which could be associated with customer growth, customer retention, efficiency improvements, cost reduction or reducing risk.

Copyright © Intelligent Business Strategies Limited, 2022, All Rights Reserved

Data product common vocabulary and schema definition

Data product schema should be defined using business data names documented in a business glossary

Having done this, data product owners can be appointed, and appropriate teams of domain-oriented subject matter experts formed to define common business terms and data definitions for data in each identified data product. The common data names for data products are defined in a business glossary within a data catalog. Once this has been done, each data product owner can work with their domain-oriented team to define the schema of their data product using data names already defined in the business glossary.

Data product development

Discover, classify, and catalog source data to enable data producers to see what data is available

Having defined common business terms and a schema for a data product each domain-oriented team of data producers can utilise a lakehouse to build data products for sharing. To do this involves using a lakehouse or third-party data catalog to register and scan one or more data sources to discover what data is available. Data sources could be live data streams on scalable messaging systems like Kafka or AWS Kinesis, relational DBMSs, NoSQL DBMSs, files on cloud storage, SaaS applications, external data provider data, and more.

Once scanned, data product producers then use the data catalog to find the source data they need to create the data product. They then create a pipeline to ingest this source data, convert it to Parquet format, store it in lakehouse cloud storage and create bronze tables on that data as shown in Figure 6.

Create ETL pipelines to ingest, clean, transform and integrate data to produce data products stored in silver tables

Use ML models in ETL pipelines to extract valuable data from unstructured data sources

Having done this, data producers then create a DataOps ETL pipeline using either a lakehouse built-in ETL capability or third-party data fabric software to clean, transform and integrate data to create the required data product. For unstructured source data, it should be possible to invoke ML models in the pipeline to extract structured data from an unstructured source. For example, structured data such as people's names, product names, locations, dates, monetary amounts etc., can be extracted from text using natural language processing. This data can then be added to data from other sources to make a more valuable data product. Validation of the data, data types and schema should be enforced during ETL processing (e.g., using data quality rules) to ensure data products created are of the highest quality. Also, each data product created is stored in a lakehouse silver table as shown in Figure 6 and uses the common business data names defined for it in the business glossary. Data product version control should also be possible.

Publishing data mesh data products in a data marketplace

Publish data mesh data products in a data marketplace so data can be securely shared with all that need it

Finally, all created data products are published along with associated metadata, in a data marketplace to make them available for sharing with all who are authorised to access them. The data marketplace is a data catalog application that shows business ready data products available for sharing and consumption in a data mesh. This includes associated metadata on lineage to show how each data product was created, data product owners, common business terms to describe data meaning, data quality scores, data freshness, social ratings etc.

CONSUMPTION OF DATA PRODUCTS IN A LAKEHOUSE

Data consumers can use the data marketplace to shop for high quality data products and see lineage on how they were created

Having produced data products and stored them in lakehouse silver tables, data consumers can shop for business ready data products in a data mesh using the lakehouse data marketplace. Since the data marketplace typically uses the data catalog, the user has access to metadata associated with each data product including the owner, data meaning as described in the business glossary and lineage to describe how each data product was produced.

Copyright © Intelligent Business Strategies Limited, 2022, All Rights Reserved

Consuming data products into gold tables for use in analytics

Data consumers can build pipelines to consume the ready-made data products they need into data warehouse and feature store gold tables

Authorised consumers can then create lakehouse gold tables to consume data products required in various analytical workloads (see Figure 6). The idea is to 'assemble' the relevant business ready data products you need from the data mesh to support specific business intelligence and data science workloads. So, for example, it is possible to create data warehouse / data mart dimension and fact tables as gold tables in the lakehouse that consume selected dimension and fact data products in silver tables. Equally, it is possible for data scientists to create gold tables in the lakehouse that consume data from data products to provide features as input to the machine learning models that they are building.

Business analysts and data scientists can then query the data to create new data products, BI reports, and machine learning models

Once gold tables have been created, they too can be shared and accessed by BI tools via a SQL interface or by notebooks or data science workbenches via programming APIs (e.g., Python data frames in PySpark notebooks).

Business analysts and data scientists can then build analytical products such as BI reports and machine learning models and publish them in the data marketplace as shown in Figure 7.

BI reports and ML models can be published to a data marketplace and invoked on-demand to provide predictions, alerts, and recommendations

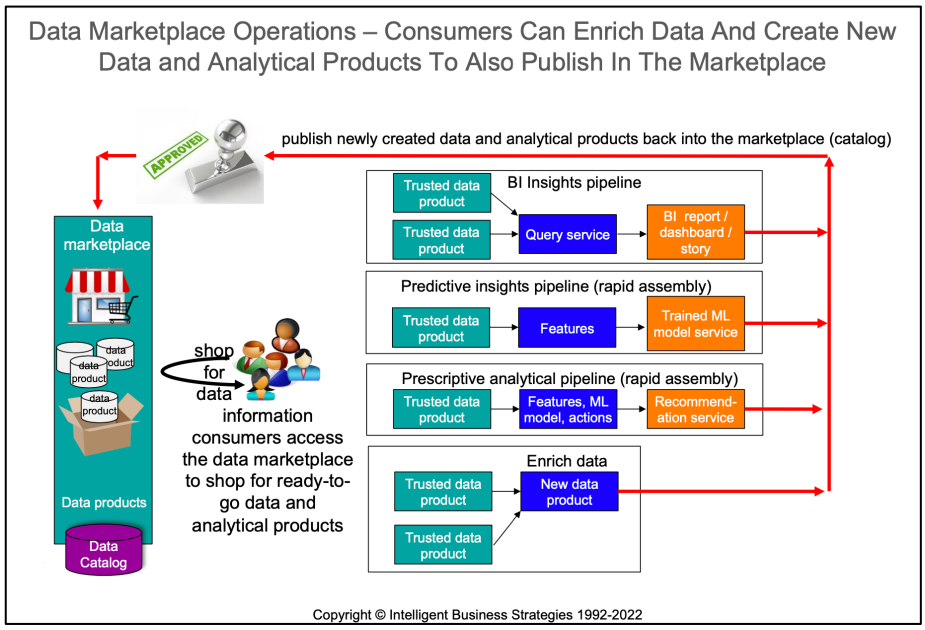


Figure 7

BI reports and ML models can be invoked by applications to create intelligent apps

In addition, analytical products such as machine learning models can also be:

- Included as SQL user defined functions (UDFs)
- Invoked on demand to provide forward looking insights, alerts and recommendations to BI reports, dashboards and to enable intelligent applications

Event-driven automated decisions and actions are also possible

- Triggered on an event driven basis to drive automated actions when business conditions are detected in live streaming data

IMPLEMENTING FEDERATED DATA GOVERNANCE USING A LAKEHOUSE

One of the four key principals of data mesh we have not yet spoken about is federated computation data governance. This is about defining and enforcing both global and local policies to govern data in a data mesh. Governance can of course span everything from data access security, data privacy, data sharing,

Copyright © Intelligent Business Strategies Limited, 2022, All Rights Reserved

Data engineers can enforce privacy policies during pipeline development to mask sensitive data

data retention and data quality. In the context of data products being produced on the lakehouse platform, data governance policies can be implemented both during and after data product development. For example, data engineers can enforce global data privacy policies during pipeline development to mask any sensitive data within a data product being created. This ensures that data products are compliant and can be shared with confidence.

Data sharing can also be governed to ensure access to data is policed and audited

In addition, data product owners can set data access security policies and create data sharing policies on data products stored in lakehouse silver tables to govern who is allowed to access and consume shared data products. Data loss prevention policies also can be set up for data products to prevent this lakehouse data from being accidentally overshared outside the enterprise.

Also, lakehouse auditing capabilities make it possible to monitor and govern access and use of data products in silver tables to ensure policies are being correctly enforced. Gold tables can also be governed in a similar way so that valuable insights remain secure while helping to drive value from the decisions they support.

Copyright © Intelligent Business Strategies Limited, 2022, All Rights Reserved

IMPLEMENTING A DATA MESH ON DATABRICKS LAKEHOUSE PLATFORM

Now that we understand how to implement a Data Mesh using a lakehouse, this section of the paper looks at how you can do this using the Databricks Lakehouse Platform. Databricks was founded in 2013 and has customers all over the world in many different vertical industries.

DATABRICKS LAKEHOUSE PLATFORM

The Databrick Lakehouse Platform runs on AWS, Google Cloud and Microsoft Azure and consists of a number of integrated software components:

The Databricks Lakehouse Platform runs on AWS, Azure or GCP and has a number of component technologies

- Databricks Unity Catalog
- Delta Lake
- Delta Live Tables
- Workflow
- Row level security
- Databricks SQL
- Databricks Machine Learning
- Databricks Marketplace

Databricks Unity Catalog

The Databricks Unity Catalog keeps track of lakehouse tables, views, columns, constraints

The Databricks Unity Catalog knows about and governs access to all data assets in the Databrick Lakehouse Platform. It is available on AWS and Microsoft Azure (with GCP available before year end) and integrates with cloud-based identity management to access lakehouse data. It keeps track of lakehouse tables, views, columns, check constraints, table privileges and more. It is also possible to have several catalogs (e.g., dev / test / production) and manage privileges across these. Unity Catalog can also automatically capture pipeline data lineage down to the table and column level across queries executed in any language as well as track assets such as notebooks, dashboards, and jobs. There is a built-in search capability and a privilege inheritance model to allow administrators to set access policies on whole catalogs or schemas of objects. Also, Unity Catalog comes with a set of pre-defined views describing the catalog that can be queried from popular BI tools or notebooks via Databricks SQL.

It also tracks notebooks, jobs, and automatically captures pipeline lineage

Delta Lake

Delta Lake is an open-source data management layer built on top of cloud storage

Databricks Lakehouse is powered by Delta Lake. Delta Lake is an open-source ACID table storage layer over cloud object stores. It is a data lake based on Delta files and tables. Delta files provide:

- A transaction log compacted into Parquet[5] file format to provide ACID transaction support to tabular data also stored in the file. The transaction log tracks committed inserts and updates, and carries out checkpointing

[5] Parquet is an open, columnar file format with compressed data optimised for query processing

Copyright © Intelligent Business Strategies Limited, 2022, All Rights Reserved

It supports ACID compliant table storage, schema validation, schema evolution and historical analysis

- Schema validation to prevent writes to tables that do not match the table schema e.g., cannot contain any additional columns or have different data types for columns
- Schema evolution, schema version control and historical time travel
- Unified streaming and batch support on tables
- Change data capture that captures row level changes from Merge / Update / Delete operations in separate files to support slowly changing dimensions

Delta Live Tables is a declarative ETL framework that uses workflow, SQL, data quality policies and materialised views to clean, transform and integrate data

Delta Live Tables

Delta Live Tables (DLTs) is a built-in declarative data engineering framework with an optimiser for defining ETL pipelines. DLTs support declarative policies and rules called ***expectations*** that run tests to prevent unwanted and poor-quality data being loaded into tables. DLTs also support streaming data.

Workflows

To orchestrate pipelines, include custom code for complex transformations and invoke ML models to extract structured data from unstructured sources to provide richer, more valuable data products.

Row and column level security

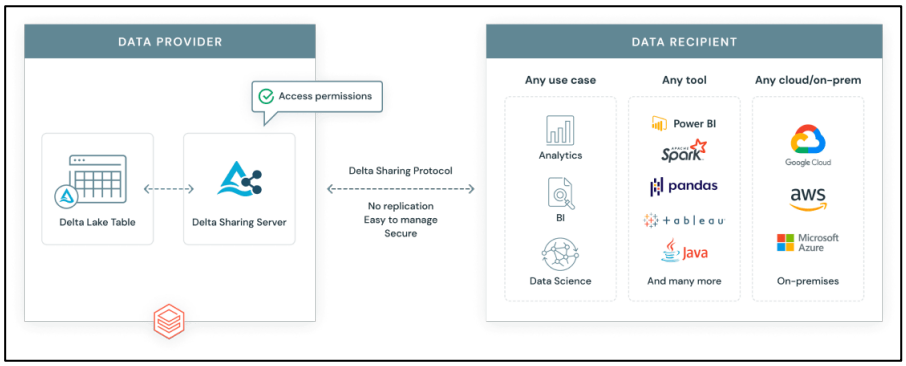
To restrict table access to rows and columns. Column masking is also supported.

Delta Sharing

Delta Sharing enables you to securely share data without data replication

Delta Sharing utilises part of the Linux Delta Lake open source project to provide a built-in, centrally governed, secure data sharing capability that allows you to govern the sharing of data from your lakehouse without data replication. It enables you to audit and track usage of shared data on one platform. Predefined templates, notebooks and dashboards are included to guide consumers through common use cases. This helps shorten time to insights. Delta Sharing enables consumers to run queries and analytical workloads using several languages including SQL, R, Scala, Java, and Python (See Figure 8).

It also enables you to govern data sharing through permissions, auditing and tracking the usage of shared data



Source Databricks

Figure 8

Databricks SQL supports SQL and dataframe concurrent query access to Delta Lake tables

Databricks SQL

Databricks SQL is a data warehouse product that provides massively parallel query processing for SQL and Dataframe API queries needing to access data in the Delta Lake tables. It is Databricks intention to extend this capability to support federated queries for querying remote data sources such as PostgreSQL, MySQL, and AWS Redshift without the need to extract and load data from sources.

Copyright © Intelligent Business Strategies Limited, 2022, All Rights Reserved

Databricks Machine Learning

Databricks Machine Learning supports model development, model deployment and monitoring

Databricks Machine Learning is built on a Delta Lake foundation and includes complete support for the machine learning model lifecycle. It includes:

- Collaborative notebooks that support Python, R, SQL, and Scala
- A machine learning runtime
- A feature store
- AutoML for rapid model development
- Managed MLFlow to track all pipelines and experiments
- Integration with Git for versioning and automated CI/CD workflows
- MLOps – model registry, model serving, model monitoring
- bamboolib - a user interface component that automatically generates Python enabling users to perform no-code data analysis and data transformations from within a Databricks notebook

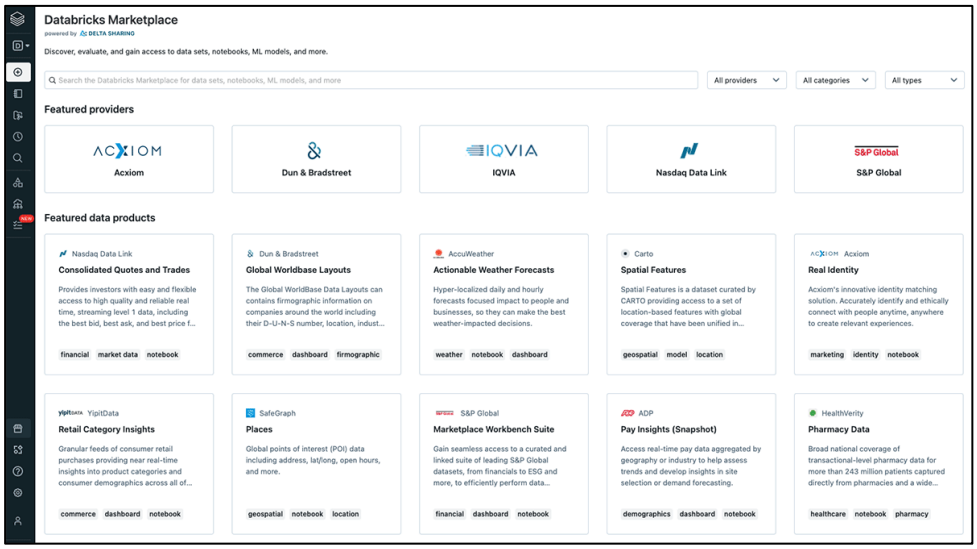
It includes collaborative development, a feature store, AutoML, CI/CD and MLOps

Databricks Machine learning is integrated with Delta Lake which means that you can build pipelines to engineer any type of data and use ML models to process and analyse data in the pipelines.

Databricks Marketplace

Databricks Marketplace is an open marketplace for exchanging datasets, notebooks, dashboards, and machine learning models. It is powered by Delta Sharing and governed by Unity Catalog. It allows you to easily discover, evaluate and gain access to data products and analytical products such as machine learning models, dashboards, and notebooks from anywhere, without the need to be on the Databricks platform. Both internally produced data products as well as data from external data providers can be published in a data marketplace (see Figure 9).

Databricks Marketplace enables internal and external data products, notebooks, machine learning models and more to be shared across the enterprise and beyond



Source Databricks

Figure 9

DATA MESH IMPLEMENTATION USING DATABRICKS LAKEHOUSE SOFTWARE

Figure 10 shows how a data mesh can be implemented on the Databricks Lakehouse Platform using the aforementioned platform technologies.

Copyright © Intelligent Business Strategies Limited, 2022, All Rights Reserved

It is possible for multiple teams of domain-oriented data engineers to create data products in a data mesh on the Databricks Lakehouse Platform

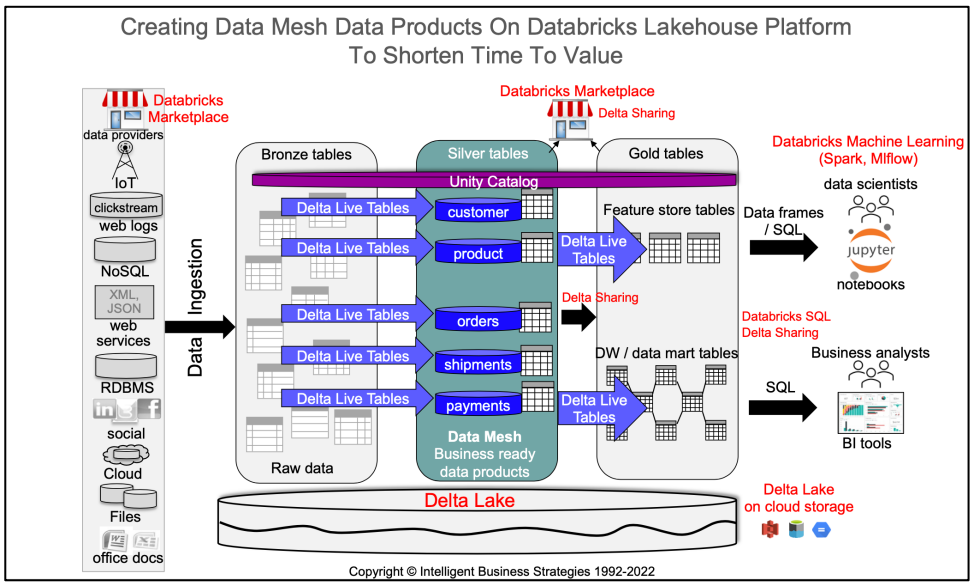


Figure 10

BUILDING ETL PIPELINES TO CREATE DATA PRODUCTS USING DELTA LIVE TABLES AND WORKFLOW

Delta Live Tables can be used to declaratively define ETL pipelines to ingest data into an organised lakehouse and create data products

Multiple teams of domain-oriented data engineers can use Databricks Delta Live Tables (DLTs) declarative ETL framework to create ETL pipelines that ingest and process batch and streaming source data to produce data products in the lakehouse. Delta Live Tables enable you to define SQL queries or Python to build materialised views in a Databricks Lakehouse. They are created and populated using a pipeline workflow that reads and analyses the descriptive pipeline code written into notebook cells to create the tables and transform the data. Data transformations are declared in the DLTs. Databricks then optimises the pipeline to provide the best performance, automates pipeline execution and manages all dependencies. It also automatically captures the lineage associated with ETL pipeline transformations and stores this metadata in the Unity Catalog.

Streaming and batch data can be ingested from many different sources

You can read source data from cloud files, Kafka, AWS Kinesis, DBMSs etc., using normal Spark APIs and also from other DLTs. All dependencies inside of the pipeline are tracked. You simply define a live dependency using live schema. You can also create ***streaming*** live tables which are based on Spark™ structured streaming. Streaming live tables are stateful which means they guarantee that they're going to read each input row only once. This can reduce costs and data latency by only processing new data. An example of using streaming DLTs is shown below. This invokes Databricks Autoloader to ingest data into the lakehouse bronze table without the need for triggers or manual scheduling.

Streaming DLTs process data only once and only process new data

DLTs can be linked together to ingest and process data in a pipeline to create data products

```
CREATE OR REFRESH STREAMING LIVE TABLE customers_bronze
AS SELECT * FROM cloud_files("/databricks-datasets/retail-
org/customers/", "csv")

CREATE OR REFRESH STREAMING LIVE TABLE customers_silver
CONSTRAINT valid customer_id EXPECT (customer_id IS NOT NULL) ON
VIOLATION DROP ROW
CONSTRAINT valid_timestamp EXPECT (timestamp > '2010-01-01') ON
VIOLATION DROP ROW
AS SELECT * FROM STREAM(LIVE.customers_bronze)
```

Source: Databricks

Copyright © Intelligent Business Strategies Limited, 2022, All Rights Reserved

DLT constraints ensure data is cleaned and has a valid schema when creating data products

Note that the silver table takes data from the bronze table and so there is a live dependency in the pipeline that DLT detects so that it can ensure all operations are executed in the right order. Creating a Delta Live Table with *expectations* enables raw data to be cleaned and transformed to create a business ready data product in the silver layer. DLT expectations are declarative constraints that test the quality of the data being loaded into the silver table. Any violations on quality constraints are automatically acted upon (e.g., drop the row) and captured in an event log for detailed monitoring. Note that data producers can use the full power of SQL (e.g., aggregates, joins, Python defined Spark SQL UDFs etc.) to perform complex validations.

DLT pipelines can be configured to auto-scale to handle large data volumes and can be triggered or run on a continuous basis

Pipelines can be triggered periodically or run continuously to keep data fresh. They can also be configured to auto-scale if volumes are large and scale down when completed. In addition, DLT allows the same version of your pipeline to read and write to different isolated environments in support of pipeline versioning and CI/CD. This is done using a '**target**' pipeline parameter and the LIVE schema which tells the DLT framework where to publish the pipeline e.g., development, test, or production environments.

CI/CD is also supported via pipeline parameters to easily move pipelines into production

PUBLISHING DATA PRODUCTS USING UNITY CATALOG AND DELTA SHARING

Data products are stored in Delta Lake silver tables with lineage on how they were created available in Unity Catalog

Once pipelines have run and data products are created in Delta Lake silver tables, they can be made available to consumers both inside and outside the organisation. Data product silver tables are documented in the Unity Catalog along with automatically captured lineage of how they were created. Therefore, the Unity Catalog already knows about the data products in a Data Mesh.

Data products can be shared via Delta Sharing and published in Databricks Marketplace

Data products can also be published in Databricks Marketplace (shown in Figure 9) to allow authorised internal and external consumers to shop for data. This is made possible through Delta Sharing which is an open protocol developed by Databricks for secure data sharing with consumers regardless of which computing platforms they use. Delta Sharing is integrated with Unity Catalog and allows you to easily share live data in the lakehouse without copying it to another system. You can publish data products and analytical products (e.g., ML models, dashboards etc.) to the marketplace.

Delta Sharing works by creating shares, assigning one or more tables to a share, defining recipients, and granting permissions to the recipient to access the data product in the share. An example is shown below.

```
CREATE SHARE orders
ALTER SHARE orders ADD TABLE silver.orders
CREATE RECIPIENT xyz
GRANT SELECT ON SHARE orders TO RECIPIENT xyz
```

CONSUMING DATA PRODUCTS VIA DELTA SHARING AND DELTA LIVE TABLES

Consumers can shop for data products in Databricks Marketplace and create pipelines to consume the ones they need

Consumers can then use the Unity Catalog and/or Data Marketplace to find the data products they need. They can then consume this data by creating declarative consumer ETL pipelines by creating data warehouse star schema DLTs as gold tables where each dimension and fact DLT selects data from the corresponding data product DLT in the silver layer (See Figure 11).

Data products can also be consumed:

- Into gold layer feature store DLTs for use by data scientists in ML model development
- By users of other ETL and BI tools

Copyright © Intelligent Business Strategies Limited, 2022, All Rights Reserved

Data warehouse star schema DLTs can be created as gold tables on Delta Lake to consume data products from corresponding silver tables

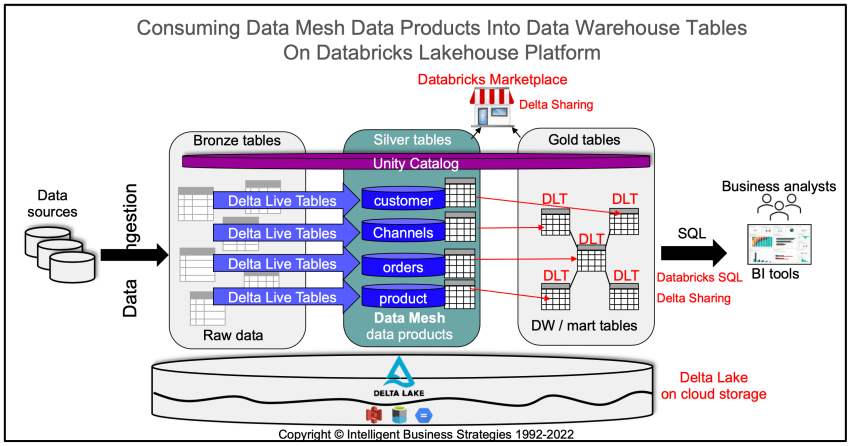


Figure 11

FEDERATED COMPUTATIONAL DATA GOVERNANCE USING UNITY CATALOG AND DATA SHARING

Unity Catalog, Delta Sharing and Databricks Marketplace help govern data product access and usage

The combination of Unity Catalog, Delta Sharing and Databricks Marketplace helps address the requirement of federated computational data governance in a Data Mesh. Domain based data product development teams can implement global data privacy policies to mask and encrypt any sensitive data during data product development. In addition, data product lineage is available in Unity Catalog and data product owners can create shares and set policies on data products stored in the Databricks Lakehouse Platform to govern access to and consumption of shared data products. Also, auditing capabilities make it possible to monitor access and data product usage.

INTEGRATING ANALYTICAL WORKLOADS TO DRIVE MORE VALUE

ML models can be created on the lakehouse and invoked by BI tools and applications to help drive better decisions

Finally, BI reports and ML models can both be built on the Databricks Lakehouse and published in the Databricks Marketplace. ML models can be invoked on-demand to provide forward looking insights, alerts and recommendations to BI reports and applications. It is this integration of analytical workloads that creates new insights to enable better decisions to drive more value.

Copyright © Intelligent Business Strategies Limited, 2022, All Rights Reserved

CONCLUSIONS

Data and analytics are now considered strategic in the boardroom

In the digital economy, almost every CEO wants their business to become data driven. Therefore, it is not surprising that data and analytics is now considered strategic in the boardroom. Every department from marketing to human resources, now wants to analyse integrated, high-quality data to provide timely insights to help them make more effective decisions. As a result, the number of business users wanting to analyse data has also grown rapidly.

However, implementation approaches to date have been siloed, fractured and expensive

However, despite the demand, implementation approaches taken to date have been fractured with multiple siloed analytical systems supporting different analytical workloads and each system has become too dependent on small groups of centralised expert data engineers to clean and integrate data. Also, the same data is often used in different systems. Data engineering across all these systems is not only fractured, but it is too costly, prone to reinvention as opposed to reuse and data engineers can't keep pace with business demand.

Democratisation of data engineering is needed to speed up creation of business ready data

Businesses also want to integrate analytical workloads to create new insights to improve decisions and outcomes

The requirement now is to democratise data engineering and industrialise the production of business ready, high value, data products by enabling multiple domain-oriented teams of data producers across the business to build data products using multiple types of data and make them available for sharing and reuse. The objective is to shorten time to insights. It is this that has spawned the emergence of the Data Mesh approach. However, it is also a requirement to tear down analytical silos and integrate multiple analytical workloads to drive more business value on a richer set of data. For example, the output of data science can become the input to business intelligence. Also publishing reusable analytical products like ML models and BI reports enables rapid development of intelligent applications, automated decisioning, automated actions and timelier, context-aware decisions.

Both can be achieved on the Databricks Lakehouse Platform

Both of these requirements can be achieved by implementing a Data Mesh on the Databricks Lakehouse Platform by ingesting raw streaming and batch data into cloud storage using bronze Delta Live Tables in Delta Lake and creating data products using silver DLTs, exceptions and pipelines to clean and transform the data. Data products and their lineage are automatically documented in the Unity Catalog. They can be shared using Delta Sharing and published in the Databricks Marketplace which can become the user interface to the Data Mesh. These data products can then be consumed by gold DLT data warehouse and feature store tables also on Delta Lake for use by business analysts and data scientists respectively. Business analysts can query, report, and analyse gold star schema tables using BI tools via Databricks SQL query engine while data scientists can develop and deploy ML models using Databricks Machine Learning. Furthermore, ML models can then be deployed, registered in a model registry and invoked by BI tools and applications for future-oriented insights and recommendations. They can also be published in Databricks Marketplace.

A data mesh can be created in Delta Lake silver tables and relevant data products consumed into data warehouse and feature store gold tables for use by business analysts and data scientists

For companies looking to democratise data engineering while shortening time to value, all of this capability makes Databricks Lakehouse Platform well worth considering. However, when implementing a Data Mesh, it is not just technology that matters. It also requires organisation to know who is building what data products, what ML models and what BI reports and how they all fit together to improve business outcomes.

Copyright © Intelligent Business Strategies Limited, 2022, All Rights Reserved

About Intelligent Business Strategies

Intelligent Business Strategies is an independent research, education, and consulting company whose goal is to help companies understand and exploit new developments in business intelligence, machine learning, advanced analytics, data management, big data, and enterprise business integration. Together, these technologies help an organisation become an *intelligent business*.

Author

Mike Ferguson is Managing Director of Intelligent Business Strategies Limited. As an independent IT industry analyst and consultant, he specialises in BI / analytics and data management. With over 40 years of IT experience, Mike has consulted for dozens of companies on BI/Analytics, data and analytics strategy, data architecture, data governance, data warehousing, technology selection, and intelligent applications. Mike is also conference chairman of Big Data LDN, the fastest growing data and analytics conference in Europe. He has spoken at events all over the world and written numerous articles. Formerly he was a principal and co-founder of Codd and Date – the inventors of the Relational Model, a Chief Architect at Teradata on the Teradata DBMS, and European Managing Director of Database Associates. He teaches popular master classes in Data Warehouse Modernisation, Big Data Fundamentals, Centralised Data Governance, of a Distributed Data Landscape, Creating Data Products in a Data Lake, Lakehouse or Data Mesh for Use in Analytics, Machine Learning and Advanced Analytics, Real-time Analytics, and Data Virtualisation.

Telephone: (+44)1625 520700
Internet URL: www.intelligentbusiness.biz
E-Mail: info@intelligentbusiness.biz

Shortening Time to Value by Building Data Products in a Lakehouse
Copyright © 2022, Intelligent Business Strategies
All rights reserved

All screenshots, diagrams and code samples sourced from Databricks and used in this paper remain the copyright and intellectual property of Databricks