Mastering Microsoft Fabric

Unified Data Engineering, Governance, and Artificial Intelligence in the Cloud

Sibaram Prasad panda

DeepScience

Mastering Microsoft Fabric: Unified Data Engineering, Governance, and Artificial Intelligence in the Cloud

Sibaram Prasad panda
Decision Ready solutions

DeepScience

Published, marketed, and distributed by:

Deep Science Publishing, 2025
USA | UK | India | Turkey
Reg. No. MH-33-0523625
www.deepscienceresearch.com
editor@deepscienceresearch.com
WhatsApp: +91 7977171947

ISBN: 978-93-7185-751-2

E-ISBN: 978-93-7185-785-7

https://doi.org/10.70593/978-93-7185-785-7

Citation: Surname, J. (2025). *Mastering Microsoft Fabric: Unified Data Engineering, Governance, and Artificial Intelligence in the Cloud*. Deep Science Publishing. https://doi.org/10.70593/978-93-7185-785-7

Preface

The development of cloud platforms has changed how organizations manage data, implement governance, and incorporate artificial intelligence into business processes. **Microsoft Fabric** combines data engineering, governance, real-time analytics, and AI into a single, scalable ecosystem.

This book, *Mastering Microsoft Fabric: Unified Data Engineering, Governance, and AI in the Cloud*, is designed for professionals, researchers, and architects interested in Microsoft Fabric. It covers real-world use cases, architectural patterns, and practical implementations, this guide explores how to build modern, governed, and intelligent data systems that meet the demands of today's dynamic digital environments.

Drawing on extensive experience in databases, cybersecurity, and AI, I have written this book to address the divide between theoretical concept and practical implementation. This work focuses on role- and rule-based access control, multi-tenant data governance, AI integration, and secure data pipelines, all critical pillars in modern enterprise architecture.

This book functions as both a technical guide and a strategic reference, outlining how Microsoft Fabric is influencing cloud-native data engineering and decision-making. It aims to inform readers about compliance focused architectures and servers as a resource for professionals working within cloud-first and AI-driven environments.

Sibaram Prasad Panda

Table of Contents

Deep Science Publishing, 2025
https://doi.org/

DeepScience
Open Access Books

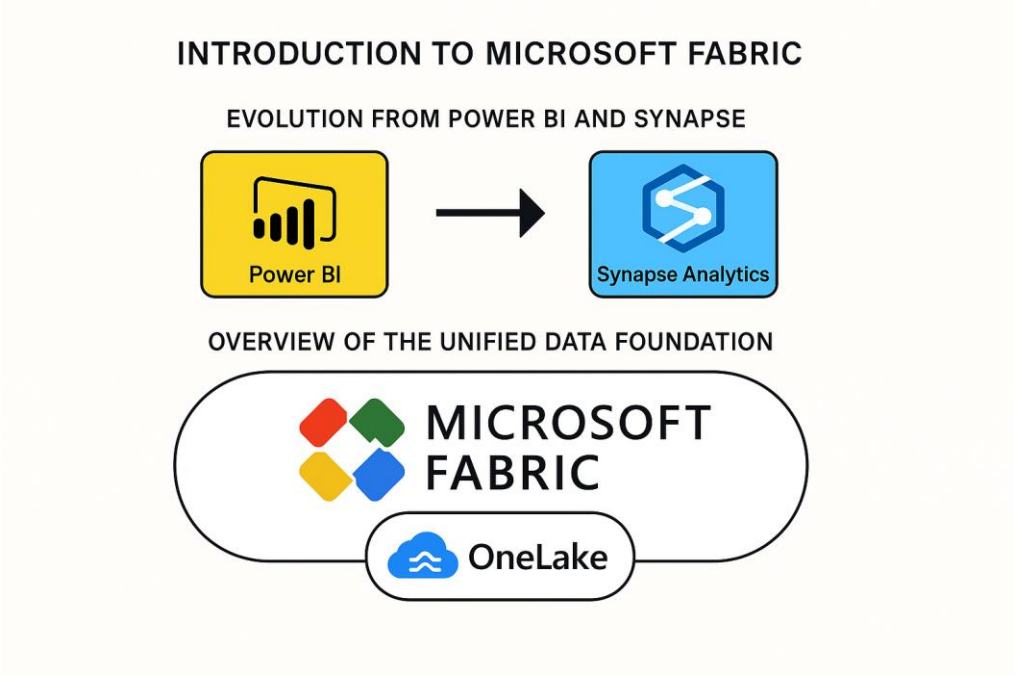
Chapter 1: Introduction to Microsoft Fabric

1. Introduction

Microsoft Fabric is an evolution of two technologies: Power BI and Azure Synapse Analytics. Microsoft Fabric empowers every organization to be a data-driven organization, integrating a broad set of product capabilities into one unified platform with the same experience across analytics personas in the organization.

Power BI has historically delivered a widely accessible and easy to use modern self-service reporting and analytics capability that answers business questions but does not enable data engineers or data scientists to sufficiently explore the underlying data, a capability of the professional BI solution. Visualizations and dashboards connect users with insights but do not allow users to connect with the process of insight development. Synapse Analytics addressed the needs of the professional BI developer, building enterprise-grade BI solutions for business or industry use cases. Despite the capabilities brought to enterprise analytics, most Synapse customers do not invest significantly in the enterprise BI capabilities, causing confusion as to which tool to use when.

The goal of Microsoft Fabric is straightforward. Innovate in the world of analytics in a way that makes it easy for anybody in the enterprise to use, that is the mission. Easy-to-use tools at every level, from self-service through professional development, with confidence in the accuracy and robustness of the data, and the insights being delivered. With Fabric, business users will no longer be in a world where they must go through the actual process of developing a model and creating a report without input or oversight. Connect and comment capabilities allow reports and dashboards to become more collaborative.



2. The Evolution of Power BI

Although Power BI as we know today was announced in 2013, its roots go back to a project called "Project Crescent", which entered preview mode way back in 2006. This original product was called SQL Server 2012 PowerPivot or "Power Pivot", and was focused on power users, allowing the creation of data models and publishing them to the Microsoft cloud model for BI: SharePoint. With the arrival of Office 365, the Microsoft cloud product became cheap and popular for affairs, and Microsoft started to change Power Pivot from a business users' tool to a capital users' tool. The goal of the new project, called Project Crescent, was to create a tool that allowed business users to create data visualizations easily, using natural gestures such as drag and drop.

In July 2013, Microsoft finally announced the general availability of Power BI, which included a workspace for data analysts, a business users visualization tool, the Capsione model server, and a cloud tool for self-service users using different data sources or data sources published by data analysts. When in the first week the Microsoft cloud BI tool hurt in Azure for more than 8 million dollars, the company realized about the major initial interest and decided to proceed with the tool, releasing a new version of Office 365 called Office 365 ProPlus where it roots Power BI. Power BI expanded to Azure with a second Content Pack, and

after several months Microsoft changed the URI structure of Power BI. Over the coming years, Microsoft invested in resources into Power BI and announced opportunities into Power BI, until to the point in 2019 when they decided to integrate Power BI and Power Point into Office.

2.1. History of Power BI

It all started in 2006, with a dedicated group in Microsoft SQL Server that set out to develop a new project, called Project "Gibraltar," which consisted of some standalone applications and a data visualization component that was based on Excel. In 2009 Excel 2010 was released, and the integrated version of the component was made part of the application, hence the name “PowerPivot.” In December 2010, a public beta of Power BI for the Office 365 – a cloud-enabled version of Microsoft Office – was released, but the tools were not integrated into a single product. In 2011 the RTM version of PowerPivot for SharePoint was released, and Microsoft also began to release PowerPivot applications and components for SQL Server. The tool had its first major update in 2012, with the launch of Power BI for Office 365 Gateway, and in March of the same year, Microsoft updated Excel to support 64-bit architectures. In 2013, Excel 2013 finally released Power Map (the 3D tool, formerly known as GeoFlow), and Microsoft changed the name of PowerPivot to Power Query – with the new version of the tool available as an add-in – but both tools remained separate from Power BI. The new solution was actually presented as a tool for business analysts. On July 2015, Microsoft merged all applications into the solution known as Power BI, allowing easy self-publishing and sharing by business analysts without any IT involvement.

2.2. Key Features of Power BI

Power BI is one of the leading BI tools today because of its powerful features, and its easy integration with Microsoft products [1-3]. Power BI has multiple flavors which can be distinguished based on the pricing and the deployment architecture. The Power BI desktop is available free for use and provides options to publish the report to the Power BI service. Power BI service is the cloud-based software as a service offering. The Power BI Pro allows support for scheduling refresh and sharing of reports and dashboards, and hence organizes collaboration over Power BI reports and dashboards. The data remains secured in Microsoft data centers and a role-based criteria can be used to allow access to dashboards and reports. The Power BI Premium contributes to better performance by providing dedicated resources for larger data processing and automates refreshes with faster processing times. It also allows reports and dashboards to be shared with users who do not have Power BI Pro license. The Power BI mobile apps

facilitate easy access to reports and dashboards on the go. IT administrators can use Power BI Report Server to store organizational BI reports on-premise to meet specific compliance requirements.

Using Power BI desktop, the end user can connect to multiple data sources, create attractive dashboards, and share them with others. The Power Query tool provides the ETL functionality for getting data, transforming the data into the right structure, and pushing it into the Power BI data model. Power BI utilizes the in-memory data model to increase query performance. The data model is built on the tabular model which allows easier processing using relationships as well as the DAX functions. Also, being a Microsoft product, Power BI can easily integrate with SQL Server Analysis Services. The composite modeling feature allows the user to use Direct Query as well as the import model in the same PBIX file.

2.3. Integration with Other Microsoft Products

Microsoft Power BI is an integrated part of Microsoft's family of software products. Office 365 is the home to various productivity suites and tools that customers use on a day-to-day basis, from Word to Excel to Dynamics 365 to SharePoint for business process collaboration. Power BI for Office 365 is the integration of Power BI with Office 365. This means that companies can experience a true live connection between their Office productivity data and their Power BI business intelligence environment. The SharePoint integration allows users to take advantage of SharePoint to distribute Power BI reports in the same manner as they do any other document stored in SharePoint. Within SharePoint, users can either take advantage of the basic Power BI web part or the more advanced Office 365 video integration to create a safe place for sharing and discussing reports with other users. Excel workbooks that are stored in SharePoint and that contain Power Map visualizations can be played back through SharePoint, allowing users to explore the data contained in the workbook in a fully immersive way. Furthermore, SharePoint also contains a business activity report to help report on SharePoint usage, so all the Power Maps that are already available can be used to visualize that data and place it on the Office site. Besides Power BI's integration with Office 365, the Power BI Publisher for Excel allows Excel users to pin Excel ranges as visuals in the Power BI environment. Microsoft Azure is the layer on which Power BI is built. Azure hosts the Power BI web services. Azure also provides machine learning capabilities through services that Power BI can leverage through connectors. The Power BI Data Management Gateway allows users to schedule data refreshes for on-premises SQL Server Analysis Services models and for imports from relational databases.

The gateway can also be used to push data into the Power BI service from on-premises data sources.

3. The Development of Azure Synapse

Prakat has been an Azure Synapse field development partner since 2020. We are working on our 10th client implementing Azure Synapse. Working with Microsoft has its perks, including having some of the best minds in the Azure and data spheres as your advocates and mentors as you explore the new capabilities and features. That advantage has allowed us to see the application of some of these new elements before they are released and help to shape the direction of Azure Synapse. This is the eight in a series of blogs that describe the current state of Azure Synapse and where it came from on its journey.

Since the well-received growth of Azure Data Lake Insights and Azure Data Warehouse, Microsoft had added a few widely used capabilities. This included the introduction of Power BI Data Flows, the increasing enterprise-ready maturity of Azure Data Lake, the general availability of Azure Purview, the release of Azure Data Factory, and the success of Azure Data Lakehouse with Lakehouse connectors.

While growing to this newfound clientele of corporate users, complex ETL, reporting, advanced analytics, and deeper data governance was the conversation moving inward. With literally thousands of features released in Azure Data Lake, Azure Data Factory, Microsoft Purview and Power BI; the conversation began moving outward. A few flagship customers said they wanted a central management and development environment. People wanted to see in one place what had underlying dependencies across multiple data products. All these demands and others, lead toward the announcement of Azure Synapse.

A key tenet was to support the entire journey for enterprises and not get locked into one approach or technology such as ETL, or reporting or previous investment in a given technology like SQL. With this understanding, Azure Synapse was built to serve technical people on projects and help all they supported working on reporting, ad-hoc analysis, business partnerships get azure analytics into the Azure cloud.

3.1. Overview of Azure Synapse

The move from data engineering to data analytics has indeed come a long way. In early days businesses relied on an army of analysts, maybe hundreds in a large

financial services company, who ingested data into the many spreadsheets to make some sense of what the data is trying to say. With customer data growing at an explosive rate, businesses needed more than just Excel – they needed a comprehensive data program that could pull data from disparate sources, tousle them together, clean them, stitch them into a consolidated dataset and deliver true analytical insights. Enter the enterprise data warehouse. After 20 years of improvement, solutions were able to load data much faster, redo on demand what it used to take months to do and deliver the requested insights in hours and days.

Welcome to 2023. This time businesses are inundated by data from the latest data deluge – business applications now generate their own data in large volumes, 24 x 7. Supply chain, security and reporting need live insights as opposed to the point in time reports that Data Warehouses have delivered. Digital Leaders in Retail and Travel & Hospitality have had to build Analytical APIs-enabled Data Lakes. Unfortunately most Analytical Data Lakes today are, well, lakes, and quite simply do not have the capabilities to deliver data products faster. Enter Azure Synapse. It starts by merging the SQL-based Data Warehouse with the Analytics Toolset of Dataflow-DataLake-Data Science- Business Intelligence into one unified analytic workspace from where different personas can tap into the same datasets and build all kinds of data products more quickly.

3.2. Key Features and Capabilities

Introduced in November 2019 as Azure Synapse Analytics, the service is considered Microsoft's Cloud Data Platform, providing advanced analytics capabilities to process and present data available in different Azure services. Azure Synapse offers enterprise capabilities that originally belonged to two distinct products: Azure SQL Data Warehouse and Azure Data Lake Analytics. The service provides the full data lifecycle management that SQL Data Warehouse offered from a data configuration, structure, and query point of view, while the control for data storage and costs, as well as the ability to run different data processing activities for mass and microscale consumption, are present with the Data Lake integration from Azure Data Lake Storage Gen2. Typically, initial access to Azure Synapse is done through the dedicated integrated web interface: the Azure Synapse Studio. The Studio provides user-friendly environments for multiple collaboration use cases in Enterprise Data Warehousing or in Data Science workflows. From the Studio, analysts and data engineers can collaborate on building dashboards and reports based on their self-service and data exploration needs, while data engineers, data scientists and ML engineers can share resources for developing pipelines or notebooks targeting Hybrid Data Management and Machine Learning workloads. Through the Notebooks

interface, the Azure Synapse Integrated Development Environment provides collaborative interactive data processing with support for multiple language modes: SQL, Python, Scala, and C#.

3.3. Use Cases and Applications

Azure Synapse Analytics has a rich set of use cases that span various industries. Its flexible solutions involving data integration, preparation, management, and exploration make it an attractive option for enterprises creating and scaling modern analytics solutions. Some common application areas of Azure Synapse include Data Warehousing, Big Data Analytics, Augmented Analytics, Advanced Analytics, Data Integration and Orchestration, Familial Exploring, and Business Intelligence. These application domains are briefly summarized below.

Data Warehouse – It allows data engineers to streamline the creation of enterprise-grade data warehouses through Synapse Studio. Data engineers can ingest data from Azure Data Lake into a server less SQL pool.

Big Data Analytics – Azure Synapse supports a server less, on-demand model that allows customers to run ad-hoc interactive analytics on petabyte-scale data lakes without having to configure the infrastructure manually. Customers can create multi-cloud analytics solutions using Spark pools of Azure Synapse.

Augmented Analytics – Azure Synapse incorporates and provides experiences created for more advanced analytics, machine learning, and data science projects. Customers can perform automated machine learning and deploy updated models using Azure Synapse.

Advanced Analytics – Azure Synapse allows businesses to analyze their big data both at rest and in motion. Using Azure Synapse, customers can correlate batch and streaming data analyzed using the same query language or using notebooks.

Data Integration and Orchestration – Azure Synapse allows customers to integrate, clean up, and modify their data using dynamic transformation scripts within data pipelines.

4. Transition to Microsoft Fabric

The transition of Microsoft Fabric from Power BI and Synapse is a natural evolution of the foundational cloud technologies that keep being told by Microsoft for many future years, enabling more corporate analytics scenarios. The spectacular success of Power BI and its user-friendly creation of analytical

data assets in the cloud pushed in many cases the definitions of the corporate analytical architecture back to the departmental level. Power BI utilized Dataflows to declare data preparation tools that could be used by everyone. Still, they had to use the adhoc Direct Query connections to the Corporate Data Warehouses and Data Lakes to utilize the analytical data required for visualizations that reflect the corporate status and projections. Data Report servers took care of some of these limitations but were not a true answer.

The Microsoft Fabric answer to this dilemma is to have one cloud-based analytical solution to come to data preparation and storage with the Azure constellation and truly keep both Power BI-client side and Synapse-SQL Client side usage in sync. The created tasks, the defined refresh-of-the-data-asset schedule, and the assets themselves look and also feel like a unified environment. It is one platform, one enterprise solution where each of the available engines has its appealing capabilities, thus supporting the workload of data engineers, data administrators, and data consumers defined in its shared framework. This alignment and defined workload enable all solutions to have a concurrent change capability through Change Data Capture and scheduling of this capability. Composite change join scenarios are thus supported and enable business capabilities for various disciplines to be created and enhanced in a more time and effort-efficient way.

4.1. Rationale Behind the Transition

The rationale behind the transition from Power BI and Synapse into Microsoft Fabric is their well-known and existing integration and complementarity, alongside additional ambitions of Microsoft inertia to strengthen data support and services from edge to decision-making, conceiving and transforming data into actionable insights. Microsoft acknowledges that Azure has become the most data-mature cloud and that companies are increasingly investing in analytics, data estate, growth, and modernization. Embedded data analytics is set to grow significantly, and unified analytics from data to insights is also expected to increase. The enterprise agreement renewal and cloud tax encourage organizations to migrate unused services into credits. Helping companies effectively plan and execute their data strategy is Microsoft's mission. Power BI is not only available on Azure but also across service applications that help companies increase productivity, such as Office 365, Dynamics 365, and Microsoft Teams and Places. Those organizations need to invest in data quality and consolidation efforts, including semantics and dictionary knowledge management – from Edge to decision-making.

Embedded BI experiences are crucial to democratize data analytics and insights creation and consumption beyond BI and analytics teams. Microsoft has significantly accelerated Power BI embedded capabilities and analytics democratization from Business Application, strengthening the embedded Power BI offering for developers, customers, and partners. Cloud scale data lakes have emerged as the solution to scale BI and analytics services from specialized business units across the entire organization. More-mature companies are leveraging Microservices Back-End-Based Services from Collaboration Ecosystem Partners connected to their data lakes more than dedicated solutions with third-party integration services. Data Lake Storage has become the data reservoir from which organizations gain value-added analytics.

4.2. Key Innovations in Microsoft Fabric

To enable this vision, Fabric includes several key innovations:

Integrated Experiences. Fabric consolidates both the analytics and data pipeline studio experiences into a single Fabric interface. Instead of switching back and forth between services, customers can now easily find and work with relevant data from within any stage of the analytic development lifecycle, no matter which Fabric product they are using. For customers who do not require this level of integration, Fabric implements this less integrated experience as well. For instance, the Fabric Data Factory is a service without a dedicated interface. It is a Pipelines tab on the Data Hub page of Fabric.

Unified Data Management. Fabric makes it easy to discover and understand the data that organization has in Fabric, while also providing intelligent recommendations so that organizations can discover the data used in their projects automatically. Fabric provides centralized management of data access control policies and data preparation using Data Lake and Dataflow. Customers using Dataverse, Data Lake and Dataflow can share data across the Fabric ecosystem without needing to copy it. In Fabric, customers only copy relevant data files, saving on storage costs. This unified data management makes it easy for data engineers and data scientists to collaborate more easily with data analysts who are users of Power BI.

Boosting Development with Templates. Organizations can significantly accelerate project development using templates. Templates contain everything needed to recreate either a complete analytic and reporting solution based on a set of business objectives, or a specific component, such as a report dashboard, SQL query, or dataflow. Templates are authored by consultants and certified for

use in organizations and shared with organizations directly from the Fabric interface!

4.3. Comparison with Power BI and Synapse

Power BI is widely regarded as the best self-service BI tool on the market. Over its long evolution, it has solved innumerable pain points for business users and is integrated with hundreds of data sources. However, as enterprises have scaled out with more data, and as the types of analytics required have matured towards advanced analytics and AI-enriched analytics experiences, there have been examples of Power BI not meeting business needs, including hard ceilings on enterprise scale and adoption or suboptimal performance and experience when dealing with suboptimal models. Synapse provides more sophisticated enterprise features such as larger model sizes, hybrid connectivity, more complex data sources and task orchestration, and an analytics engine built for trained data professionals who can leverage a deeper set of tools and code-based workflows.

However, over the years, Synapse also has accumulated some technical debt and it mixed a lot of disparate things – tech for different analytics personas including data engineers, analysts, and data scientists, different analytics project lifecycles from early stages to production, and different native query technologies from T-SQL to Python – and as a result it was a more complex experience from a tools and UI standpoint. Power BI and Synapse developers have worked closely over the years, leveraging the fact that they query a common service based on a common architecture. Team members on both development teams have also collaborated a lot on usability and back-and-forth capabilities. Fabric's unified experience enables us to have tighter integration, much deeper collaboration as well as to innovate faster amongst analytics personas! All of the capabilities that reliable and scale-out analytics service must deliver, like quality, latency, freshness, and performance, unify under the core Fabric service, created a single point of ownership enabling a better experience and mutual collaboration between all of these teams.

5. Overview of the Unified Data Foundation

The Unified Data Foundation is the public implementation of a data strategy that enables any organization to drive decision-making through the use of data. Organizations today are struggling to set up the data foundation that can enable the use of data for more informed business decision-making. At the same time, organizations require more sophisticated products and services that use AI, and

they require these services and products to be built more quickly and to evolve over time, with AI driving innovation challenges nowhere near done. As data becomes more central to the operation and the product/service set of an organization, the need for a Unified Data Foundation becomes more imperative so that the portfolio of use cases and services can be handled in a simplified and scalable manner, using off-the-shelf, battle-tested capabilities in a coherent manner with common data models, features such as data governance, security, and datasets that can be shared and also provide value for both producers of data and consumers of data. Otherwise, organizations will continue to run the risk of chasing "one-cool-use-case" scenarios that end up being neither cool nor usable over time and will be left with a data swamp.

A Unified Data Foundation is built around a framework that consists of three layers of technology: the data ingestion and data pipeline capabilities to enable data ingestion and movement across datamart from both internal and external sources; the data engineering capabilities that enable data transformation and cleansing; and the storage, query, and governance capabilities that allow for sharing wealth and cleansed data in a governed and secure manner as a shared source of truth for business analyzers and end users alike, or to be fed into production pipelines to enable automation by those businesses and organizations. A critical goal of the Unified Data Foundation is to allow for both batch and stream data handling using the same data solution components.

5.1. Concept of a Unified Data Foundation

The foundation of all analytics is a unified data foundation. Enterprises have traditionally engaged in buying, preparing, and moving data among disjointed systems before importing it into a reporting or analysis artifact. By dragging their data through the swampy lowlands of disparate tools and services, they create experience barriers that frustrate analysts while delaying insights for everyone else. Unfortunately, the results are usually pockmarked with holes and burnt edges. A unified data foundation, contrary to the approach just described, automates and streamlines enterprise data management into a shared experience for all involved. By taking the swamp away and making it a gardener's delight, it represents the future of enterprise analytics.

Unlike traditional enterprise analytics ecosystems, which are characterized by their variety of independently configured components such as data warehouses, data lakes, ETL tools, data prep tools, reporting tools, dashboards, and ad hoc analysis tools, it implements an experience-first approach. Fully integrated, collaborative experiences are paramount within this framework, and individual experiences are wrapped within a robust data management platform. The

platform consists of Data Factory, Azure Synapse Data Engineering, Data Science notebooks, OneLake, the Data Warehouse, lakehouse storage, and the Business Intelligence tool. Cohesively, these components make up the unified data foundation for all enterprises on a shared data storage layer called OneLake.

5.2. Components of the Unified Data Foundation

In this chapter, we discuss the different components of the Unified Data Foundation, including Data Engineering, Data Warehousing, Data Products, Data Science, Data Integration, Data Application Development, Data Insights, and Business Modeling. The term "foundation" implies that the user may or may not want to utilize all these components in their Data Solution. For an enterprise customer, the foundation provides the different building blocks enabling different Data Solutions, and the unified experience makes it much easier to accomplish common tasks across disparate components of the Foundation.

Data Warehousing provides a place to land and transform enterprise data from different sources in a unified schema, building on top of the capabilities offered by Data Integration and Data Engineering. Data serves its customers through Tables, Views, and Reports, and external customers through Datasets and Dashboards.

Data App Development and Business Model provide a low code / no code Data Domain Development and Application Development experience in a unified and integrated manner along with a business modeling layer on top of Data Insights without requiring any uplift effort. Enterprise tools provide power to advanced users enabling DevOps, Automation, and Governance. Data Science enables its customers through Analytical Models and external run-time through integrated Services via naming conventions and APIs.

5.3. Benefits of a Unified Approach

The Unified Data Foundation is a concept that emerged as a solution for enterprises suffering the pains of increase in complexity over modern data architectures. Over the last two decades, enterprises have built their data and analytics ecosystems by stitching together a plethora of cloud services, on-premises systems, and other third-party solutions. Although this approach to the data estate may have been effective at first, given the speed at which cloud technologies have emerged and industrialized over the last decade, many companies now find themselves with an overcomplicated and brittle architecture that underdelivers when it comes to meeting the business's analytical needs.

The principle behind the Unified Data Foundation is to offer a single end-to-end solution for all data and AI needs such that the need for call APIs, copying data around, sending various data and ML services signals at different points in the data processing workflow are reduced to a minimum. In providing a unified solution for all data requirements, from data ingestion, storage, model training, and serving, companies avoid the need for disparate data and AI solutions. A unified architecture abstracts away complexity and increases the speed at which business requirements can be met. It allows citizen developers to innovate with self-service and get to the insights they need faster, while at the same time enables professional developers to put their skills to good use in helping build enterprise applications. With the Unified Data Foundation, enterprises can be sure that tasks such as data governance, security standards, and compliance mechanisms are taken care of.

6. Case Studies

We will now present how some customers implemented Microsoft Fabric into their existing way of working in Power BI and Azure Synapse, and how they benefited from doing so. We use this opportunity to thank all of them for their collaboration, trust, and the kind words related to the implementation.

6.1. Case Study 1: Implementation in a Large Enterprise

A large telecommunications enterprise used to monitor Customer Experience across multiple interaction points through a dedicated IT Team using a combination of predictive analysis models, scripts, and refresh flows now migrated to Power BI integrated with Machine Learning for reports and alerts on their customers XP. The problem was that data predicted indicators were very hard to monitor and were not integrated with Marketing actions. Now, everything is monitored in real-time with Power BI dashboards and alerts through event-driven Data Pipelines using models for all their data, not only historical but also predictive, with no need of the IT Team intervention. All Marketing actions are monitored using Power BI and integrated experience, connecting different data sources.

This enterprise managed to include different types of data sources, including batch, and data stored locally and in the cloud by leveraging Pipelines, Dataflows, and Dedicated SQL Pool into their dashboards by integrating real-time alerts with Marketing team inputs. This combination allowed these enterprises to reduce the

lost customers and increase their XP, closing the loop with all enterprise areas involved.

Enterprise Entity A is a European company that owns airlines, hotels, travel agencies, and other entities in tourism and air transportation related industries. Company A has implemented Fashion Technology in a large number of brands and developed a center of excellence for analytics and technology. Company A is using Microsoft Fabric as the main technology for data management and analytics.

Implementation at Company A started in the fourth quarter of 2022 mainly for corporate vertical brand data, customer, sales, and marketing related dashboards, scorecards, and reports. In February 2023, Enterprise Entity A and Vendor B finalized a planned enterprise-wide rollout of 1,000+ reports and dashboards to brand executive management, marketing, and sales teams, as well as an industry-wide marketing spend benchmarking tool. The teams at Company A had been struggling with the data discovery process, report preparation, and subsequent/ongoing funnel engagement. The program at Company A, to progressively rid the team of such manual, spreadsheet-driven workflows relied heavily on the strength of outbound marketing mechanisms and personalized deal recommendations, working closely with additional customer data obtained from Enterprise Entity A. Modeling and visualizing historical data in a simple, confident, and easy way, in addition to forms of integration needed with back-end planning processes, helped generate and communicate very clear marketing analytics and investment visibility.

Company Entity A is using Microsoft Fabric's integration of Power BI, Azure Data Factory, Azure Synapse, and Azure Machine Learning to bridge its integration of real-time data and combine historical data on customer interactions, source data on travel and airline industry indicators, spend/purchase data across travel seasons and industry periods, as well as promotional deal-specific performance and patterns.

6.2. Case Study 2: Benefits in Small to Medium Businesses

A small chain of supermarkets included in its privacy policies related to customer order statistics accelerating the advertising-adjacent selection process to direct offers using dynamic pricing to the supermarkets. Traditionally, the stored procedures appeared to be slow and highly maintained.

That created multiple dependencies on the suppliers' IT teams. After a year, users, Data Pipelines, and Power BI dashboards did accelerate the selected pricing offer

recommendations getting feedback from Marketing and finally incorporating the final decision into Power BI.

One impactful case study involved an SMB, a property management company with approximately 50 employees. Over the years, this company had amassed a range of data sources, from their domain system with an SQL database hosting property transactions, visits, and maintenance activities, to their domain email hosting managing contacts, communications, and documentation related to sales and rentals. Despite using the domain website to publish the properties for rent and for sale, the most important journey for renters and buyers happened in external listing sites. This company needed to reduce the pressure that a declining economy had imposed on operational costs while seeking better management of revenues.

After conducting a thorough diagnosis of the main issues raised by stakeholders, we proposed the modeling of a myriad of dashboards using Power BI in the hosting of their internal business data in a Workspace dedicated to that purpose. Those dashboards would work on top of some on-premises or in-cloud connections being refreshed 3 to 4 times daily according to the load of parallel threadings. Although some business areas had already stipulated their Key Performance Indicators pilots using Excel, the adopted solution was to abandon this tool for reporting and monitoring, replacing it with a more secure scenario, the dashboards about which we were conducting and extending workshops directly with the solution end users, and consulting with them about new feature requests.

This solution was already in place, with some dashboards temporarily undergoing minor maintenance operations, when the company had to assign the processing and validation of all data necessary for the construction of those dashboards. However, due to their intuitive interfaces, facilitated Q&A features, and a one-month period of intensive workshops, we achieved user engagement and the fast achievement of business insight reports. With that, the company had by then their first major analysis of business KPIs.

7. Challenges and Considerations

Microsoft Fabric is a new solution that grows out of Power BI and Synapse, and strengthens their closer integration released earlier in 2022. It offers a simplified model that abstracts complexities that arise when using components in isolation, or needing to take data in and out of different components. In principle Fabric

offers a more integrated and seamless experience, which improves productivity while increasing the chance that users stick with Microsoft for the full end-to-end data engineering and analytics flow. Templates and features make analytical wisdom accessible across organizations. More advanced users are given enough power to avoid going down a path of least resistance that produces unmanageable solutions subject to the limitations of the simpler offering.

However, a lot of data solutions in organizations have been implemented with Synapse and/or Power BI. Organizations have spent considerable time and effort creating complex solutions across the two services, and don't naturally gravitate to moving to Fabric. New features in Power BI or Synapse will continue to be desirable. Neither of the two solutions will be going away any time soon. User functions and needs will dictate whether organizations choose to stay or go. Compatibility will be a huge consideration in moving to Fabric, particularly as organizations need to consider their existing investments and the learning curve.

An early implementation approach could take the form of a hybrid strategy – that is, provisioning and using a Fabric workspace, but continuing to use Power BI and Synapse in parallel for functions and features that are not currently being migrated. These organizations could take a careful approach, moving components over to Fabric as they are able to, but still use Power BI and Synapse to stay on top of timely needs. Eventually, a better feature set will dictate users onboarding to Fabric entirely, and organizations will move to retire any remaining components left in the other services.

7.1. Technical Challenges

The merger between Power BI and Synapse is one that is riddled with technical challenges. Power BI, being a "Report Only" product based on an aggregation-only storage engine, caches every result and never attempts to merge any results. This makes the product extremely performant for analytical queries against small models, capable of automatically partitioning and caching any size model, and able to automatically include an aggregation-only object in any SQL rewrite through a dataflow. Such capabilities are not scaled. The tiny design cost is the high composite TCO of hosting unique models to be used as a gateway into enormous datasets located in systems, such as tables and materialized views. However, for team building the product, it means the dimensional model is either re-executed or cited every single time without any sharing or between-query optimization. The costs only go up with a complex User Defined Function library. Objects are either unique or considered as "objects of power", incompatible with the capabilities of a multitenant service and the engine. It

would be incredibly financially and technically attractive to provide Power BI experience against any table in a trusted way without making the object unique.

Building a multitenant product on a scale means one must make several sacrifices with performance and functionalities for maximum value. For such system-wide interoperability to work, the system would likely have to drop support for DirectQuery on certain data sources. DataflowLink, which enables dataflows to reference datasets, allows dataflows to work for any data source. In theory, a surface like that is a neutral layer that could allow Synapse to scale up while Power BI scales down with its restrictions. Such linking would have to support SQL operations however without making objects unique.

7.2. Adoption Barriers

Before describing the technical challenges affecting the adoption of this multi-product, integrated analytics platform, it is relevant to introduce possible adoption barriers. Interest in any new cloud feature or product is related to its potential of being highlighted in marketing campaigns. Companies consider adoption of the platform before committing to use its advanced features when existing analytics and data science solutions do not solve data analysis needs or are not considered fully capable of answering the increasing ones. For most organizations, existing solutions already meet their business intelligence needs, and other analytics tools are usually seen as more advanced partners for problem solutions involving large amounts of data. Adoption of the platform by these companies should be a natural process of the evolution of existing solutions when both products evolve towards the support of more business analytics needs, but with these organizations only committing their teams to use all advanced features if the performance of their current analytics solutions do not meet their growing demands.

For most companies that are solving their business intelligence and operational reporting needs using existing solutions, daily, executive, and management reporting dashboards are published but no detailed data analysis are done using how often data are analyzed and what type of analytical stories are being told. The outlined top-down approach using these two products is being followed, respecting the unusual role of Data Engineer in most organizations. What can be learned – and monitored using available capabilities – is data cascading use by employees, teams, business departments, and business units tasked with the execution of the companies' day-to-day activities. Would it be possible to manage this data access cascade through a set of business rules defined and deployed during the designing of the cloud analytics platform used by the

company if the platform was used instead? Or added analytical tools would still be needed for more local analytics?

7.3. Future Considerations

The scale of the cloud services platform can bring current and possible future challenges and considerations into play. The governance need by the IT department and the onboarding effort needed for the end-user are both factors that a BI ecosystem implemented to support self-service and IT governance must expect to deal with. A hub-and-spoke architecture is a logical but not the only solution that might be deployed to structure the environment. Active support and activation, in addition to guidance, weight significantly in reducing self-service risk. These concepts must be enabled at the organization decisional level, but also in practice by Local Data Officers whose task is to facilitate citizen processes and to guarantee the IT-enforced compliance in the supported projects, where collaboration between business and IT will be encouraged. While restricting users' capabilities and limiting the self-service tools choice might reduce risks, these two concepts can be positively challenging when designing a self-service BI environment. Education and business support during report development sessions, as well as having the right tools in place, will facilitate the leverage of self-service with success, allowing them to loosely control other processes that would hinder the approach and defeat the purpose.

Having internal best practices is important, but it isn't enough if not accompanied by training. Employees must be taught the know-how. Companies should invest in the right people to put at the forefront. Aiming to become a centre of Excellence, the goal must be encouraging citizen developers to work on projects by supporting, mentoring, and developing the right skills rather than just governing them. Only then, with the support of knowledgeable professionals, would organizations be able to develop a business-driven IT-governed self-service ecosystem.

8. Conclusion

This paper presents our ongoing work to identify the evolution of a unified analytics product focused on the democratization of the entire analytics lifecycle. While it shares features with competitor solutions, we show that it evolved from distinct products serving mostly different user personas. We also show that this evolution was rooted in specific user motivations and that the company is tackling the challenges that the transition brings.

Our history is important not only because of what the product is today, but also to understand where it is likely to take it soon. The company changed the paradigm by which the relationship between business-oriented and expert-oriented analytics users' interface with one another. Previously, one would model the data, and then there would be some transfer – with semantics likely lost in translation – to the business user who would replace the numbers in the report models with the data validated and segmented according to the logic of the data modeler, along with the rules for visibility of such data for specific subgroups. By contrast, this division of labour assumes that there are yet more automated capabilities to alleviate such burden on expert users. Be that as it may, given that the connection area based on this paradigm is a vertically integrated product, the existence of such disparate, yet connected, solutions was bound to deal with issues of added costs in terms of labour and/or money. Addressing this optimization was the main motivation behind the building of the product. While it addresses the need to unify in one single product the ability to source, govern, engineer, prepare, manage, analyse, and share analytics artifacts, the drivers of the market will continue to challenge the company to deliver updates to address the balancing of a solution that improves efficiency both for expert and non-expert users.

References:

[1] Shivadekar, Samit. "Secure Multi-Tenant Architectures in Microsoft Fabric: A Zero-Trust Perspective." (2025).

[2] Bai, Haishi. *Programming Microsoft Azure Service Fabric*. Microsoft Press, 2018.

[3] S. P. Panda, Relational, NoSQL, and Artificial Intelligence-Integrated Database Architectures: Foundations, Cloud Platforms, and Regulatory-Compliant Systems. Deep Science Publishing, 2025. doi: 10.70593/978-93-7185-129-9.

Deep Science Publishing, 2025
https://doi.org/

DeepScience
Open Access Books

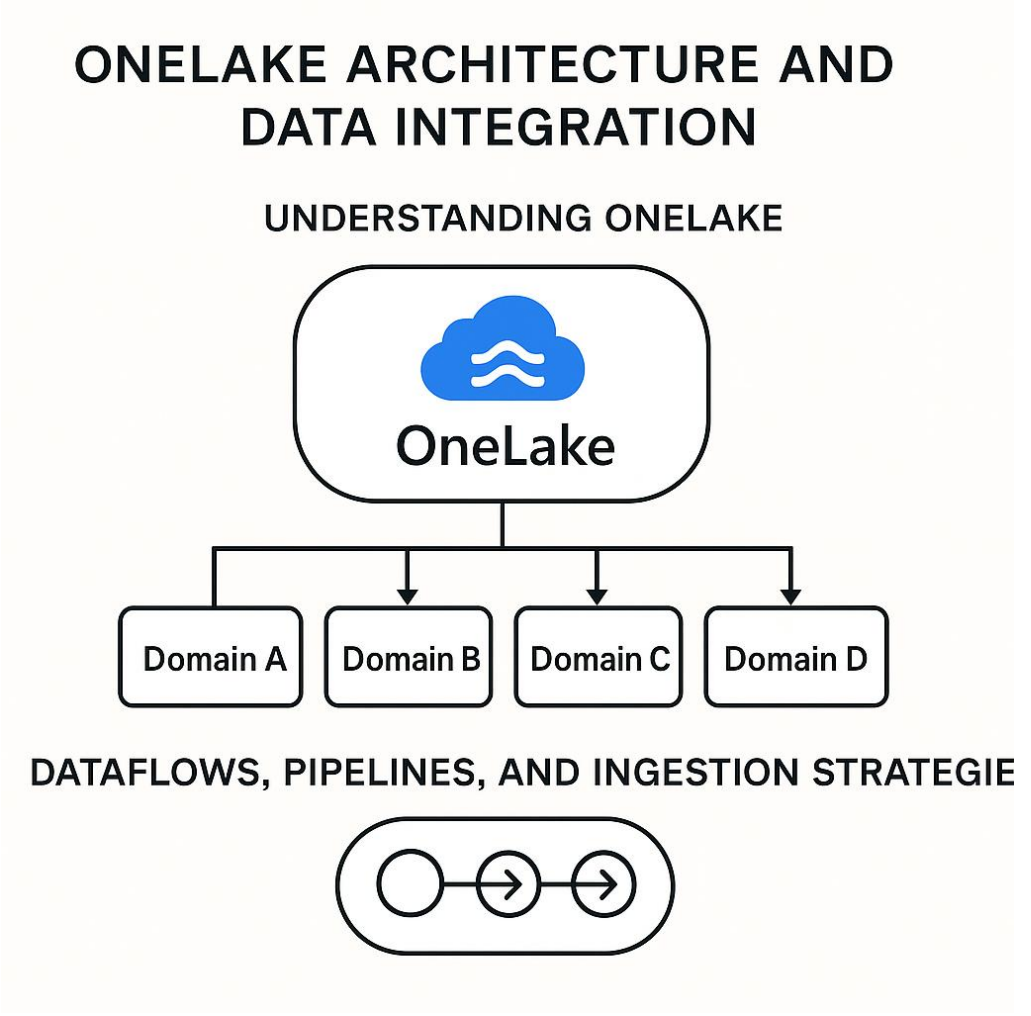
Chapter 2: OneLake Architecture and Data Integration

1. Introduction to OneLake

The OneLake paradigm proposed by the data fabric serves as the sole repository of all enterprise data, its repository being a familiar data format such as a cloud object storage, which acts as the shared context for accessing enterprise data. Enterprise data in OneLake is organized into a logical tree whose structure is dictated by the usage patterns of enterprise data, especially by the patterns of data integrations. One Lake architecture anticipates that numerous companies deploying the OneLake architecture already have a substantial investment in data integration between enterprise data that gives them business value. The investment may take the form of business applications built on data integration but need not be limited to that. OneLake provides the infrastructure necessary to let enterprises reap the business value of these integrations, and of the integrations managed in the OneLake metadata catalog, to the fullest extent possible. At the same time, companies that do not already have such an investment are also able to deploy the OneLake architecture and take advantage of what it can do. The efficiency measures incorporated into OneLake allow such companies to have a positive Return On Investment very shortly after their deployment of OneLake architecture.

The OneLake approach to enterprise data is a bottom-up one. It is designed to help enterprises who are on the digital transformation journey execute that journey on a smooth road, a journey which starts with the development of business intelligence applications, the application of machine learning to enterprise data, and generally all use cases wherein many people in an enterprise consume enterprise data, before going on to develop business applications, orchestrate enterprise workflows, and utilize enterprise data on a day-to-day

basis. The accessibility of democratized data for complex feature engineering, which is designed by ML engineers responsible for a high-level view of enterprise data, uses OneLake extensively.



2. Understanding OneLake

OneLake is a data management service that brings together data from various locations into a single location to simplify access and reduce duplication. OneLake is a single logical namespace that provides a detail view of a given data stored across multiple physical stores, including Azure Data Lake Storage Gen1 and Gen2, as well as OneLake. It provides a simple way to manage what data is included in OneLake using a single, unified set of controls, and to ensure that

only the right people and services have access to that data. It also removes the need to duplicate large amounts of file-based data within OneLake. OneLake allows data engineers and data scientists to catalog and register files from external stores, make them available as views inside OneLake, and refer to them in notebooks and pipelines without needing to make actual copies, and without compromising performance.

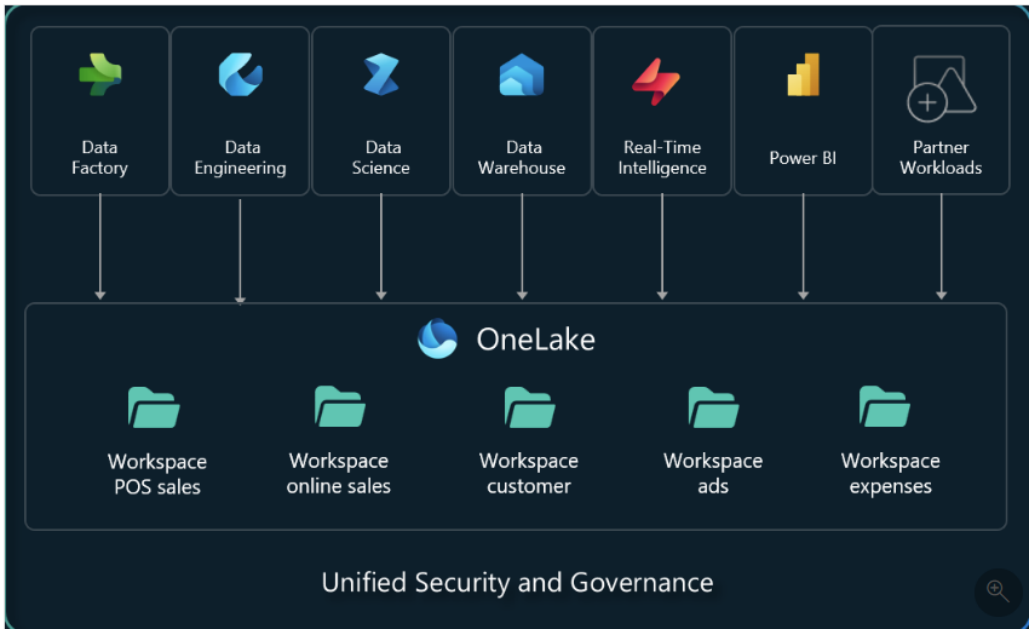
The best way to think of OneLake is as a catalog that sits on top of the actual data you may have in other stores, registering the path for those files, but serving as the interface to the rest of the services for file-based ingestion, transformation, data science, and analytics. This offers real advantages in multi-workspace scenarios. Beyond serving as the catalog for both user-created and auto-detected files from Spark, Data Factory, and Copy Data service pipelines, OneLake communicates directly with both OneLake Analytics and OneLake Data Warehousing services via integrated inter-service communications. This means that users can directly interact with files registered in OneLake anywhere in the ecosystem without needing to worry about the details of approvals and correct access permissions related to the underlying storage.

2.1. Overview of OneLake Architecture

OneLake simplifies data integration into Microsoft's cloud by minimizing the need for multiple data repositories. A single data hub, OneLake, which serves as the common repository to which many data teams load their data at different atomic aggregation levels, consolidates computed results for different use cases. OneLake achieves simplified data integration via advanced capabilities powered by a combination of unique technologies: an architecture that places OneLake at the convergence of OneLake's lakehouse storage system, data integration orchestration system, and analytics tools that use OneLake for scores of near real-time analytics. The OneLake architecture also leverages and utilizes closely related data pipelines, storage management, and applications. The OneLake architecture's consolidated data integration capabilities deliver a vision of an end-to-end analytics system.

Data integration into OneLake is driven by metadata maintained in the system. The OneLake metadata catalog enables data solutions to track all assets, programs, and pipelines associated with data integration into OneLake. As various services read or write data to OneLake, the integration catalog stores information about the data structure, format, content, location, and other attributes, making it easy to discover and use the data. The catalog also keeps track of how data has changed over time by storing metadata keys that allow data solutions to facilitate data versioning and discovery. When OneLake is

configured with a unique namespace to provide data structure organization via a shared folder structure within OneLake storage, the metadata catalog dramatically simplifies OneLake data management and discoverability capabilities, bringing strong advantages over one without the shared folder structure.



2.2. Key Components of OneLake

At its core, OneLake is designed to work alongside your existing lake, warehouse, and traditional data environments. It sits at the intersection of multiple services in Azure, including Azure Blob Storage, Azure Data Factory, Azure Synapse Analytics, Azure Databricks, Cosmos DB, Microsoft Power BI, and others. By adding a new unified management layer that sits on top of these existing services, OneLake provides an easier, more familiar way to manage your files – and provides a more integrated, richer data experience across all of your services.

You can think of OneLake as a special Azure Data Lake Storage account with automatic multi-region replication, integrated security and data governance, built-in connectors for a wide variety of data sources, and optimized for the richest data experience in Azure services. OneLake is your single, easy-to-use, and policy-enforced data source so you can begin to streamline the fragmented data experience that currently exists. OneLake utilizes components from other Azure services like hierarchical namespace, data ingestion capabilities,

integrated data security and governance, and federated query capabilities to give you these new and powerful ways to manage and discover your files and structured data. The OneLake experience will be available in services like Azure Synapse, Azure Data Factory, Azure Databricks, and Microsoft Power BI and will give you a centrally managed view of all your data, whether it's external – located in Azure Data Lake Storage accounts or data sources not controlled by your organization – or in OneLake.

2.3. Benefits of Using OneLake

OneLake brings cloud-conscious, IT-proven data management best practices out of the data warehouse into the world of data lakehouses. Data administrators can leverage the technologies they already use through an integrated experience, to plan, provision, and operate secure data lakehouse environments. Security is first-class. By default, OneLake supports the use of various security capabilities and auditing capabilities for security, and information protection for encryption. In addition, large cyber security and regulated commercial customers can use these services for additional advanced security capabilities.

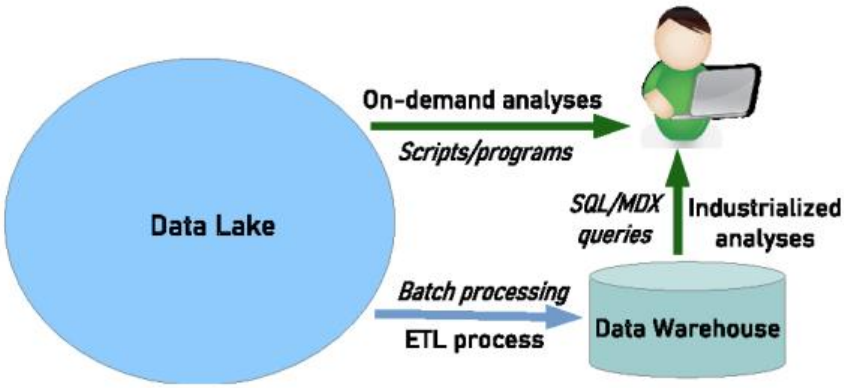


Fig. 2 Data lake and data warehouse architecture

OneLake's API surface is the same as it is for other storage solutions, plus centralized creation, provisioning, and monitoring. This surface is the same as customers already use and rely on. OneLake use cases cover the vast majority of the storage use cases for optimization which will become increasingly more important as real-time insight capabilities mature. Because keeping Delta tables optimized is expensive, there is an intention to invest in these APIs to enable OneLake in the most cost-effective ways possible. For example, OneLake can automatically implement table retention operations when certain analysis tasks occur and tap into the incremental data access guarantees that Delta provides.

All our other services are deployed in the same region as OneLake. And the rapid and low-latency native connectivity of these hybrid capabilities means that we can serve our small and diverse set of customers better.

3. Connecting Data Across Domains

The rise of multiple cloud services and domain-specific systems, such as data warehouses, operational databases, data lakes, and analytics engines, having thousands of different datasets, often a dataset or a source being available from more than one designated location, calls for a means for connecting the datasets and the datasets' sources across domains to enable data integration for analytics, ML, and other use cases. Proper and timely connectivity across data domains has thus a pronounced positive effect on data utilization and business productivity.

3.1. Data Integration Strategies

There are multiple data integration strategies available. The most straightforward method of data integration is dataset replication. Attributes of a target dataset are replicated into a source dataset, allowing both datasets to store the same attributes. Such a strong connection enables fast data blending and federated data queries but also needs fast data redundancy resolution. An alternative is to store linked datasets and their attributes at different conceptual locations while creating a connectivity mechanism enabling the dynamic resolution of interrelated datasets. This compromise between data propagation and data localization formulates classic data sharing and data connection trade-off. No matter which data architecture style is preferred, it does hinge on the existence of Metadata and the libraries that exploit metadata and the underlying technology.

The two basic approaches to enterprise data integration are content-based and schema-based. Content-based integrations include enterprise search, natural language processing strategies, and other AI-based data discovery technologies. Schema-based integration applies semantic mapping, transformation, and other strategies that relate to ETL technology. Search-based tactics can leverage metadata, dashboard definitions, or more advanced AI-assisted discovery strategies. Schema-based techniques can include both physical storage related to the traditional ETL and semantic alignment based on business semantics or governance lifecycles, such as data lineage and cataloging.

For basic schema or file-based data integrations that are more ad-hoc, allowing a little more flexibility, CSV or other delimited files might suffice. As the need for

standardization grows – for example in larger environments for more formalized data loading or for automated ingestion of data pipelines or datasets, file-based standards grow to the various formats supported by the Semantic Web stack; more specifically RDF and its serialization formats like Turtle, N-Triples or JSON-LD or an equivalent binary encoding like RDF/XML or RDF Binary. These RDF serialized standards provide firm encoding for overall schema-level formal definitions due to the industry's support for ontologies and their common formats in OWL or RDFS.

When to Follow Each Data Integration Strategy: Organizations must be very practical when deciding on which strategy to integrate each set of enterprise datasets. While the content-based techniques generally allow for more loose coupling among disparate enterprise datasets, they are typically slower, less reliable, with lower levels of data completeness and comprehensibility.

3.2. Cross-Domain Data Connectivity

Existing as well as new cloud-native systems are designed model and establish metadata for data movement, dataset correlation, and dataset connection resulting in reusable relationships between datasets. Popular data connection strategies include Foreign Data Wrappers, Dataset Linx in ELT, and federated dataset query execution. New initiatives push even further. With synthetic datasets, new foundational models are auto completed based entirely on AI while linking multiple heterogeneous datasets from disparate sources in celebration of dataset similarities and differences. This offers an entirely new definition of dataset UX but also creates a pressing necessity for combating fake news driving user frustration.

Cross-domain data integration enables organizations to leverage data from insight silos to create new value in ways that would not be possible otherwise. For those working within a single domain like finance, HR, or marketing, this concept is purely hypothetical. But for the organizations that need to synchronize data across domains or need to create something new that relies on understanding insights from multiple domains, cross-domain data connectivity delivers. Creating this cross-domain value requires understanding the internal domain data needs and managing the timing to provide the necessary data to the integrated cross-domain process. Cross-domain business process integration requires data sources with which to share data. An integrated cross-domain business process is only as good as the data it receives and how timely and accurately it receives the data. But creating cross-domain data sources is not as easy as it seems. For domain-centric data sources, data models are created and optimized for that single domain. To provide cross-domain data, these domain-centric source data

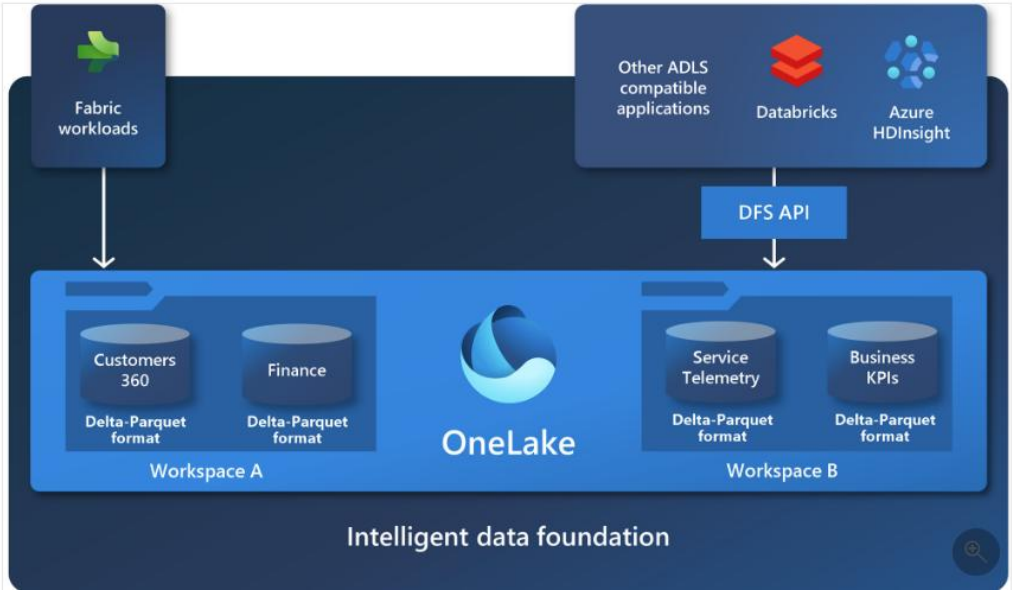
must be transformed into cross-domain data in a shared data model appropriate for cross-domain processes. Business Application technology enables organizations to get domain-centric source data into the shared cross-domain data model via BI technology, but this still only means creating a snapshot of that source domain at the time of data extraction. There can be multiple other business processes occurring within the source domain that could render that snapshot less useful, or unusable altogether, by the time the data arrives at its destination. Once the cross-domain data is created and rendered available to the destination process, a new value can be created. The opportunity to create or eliminate a relationship along that domain interface between domains occurs, necessitating different forms of cross-domain data that initiate that change, modify the terms of the relationship, or that react to and support any changed state of the new or eliminated relationship.

3.3. Challenges in Data Integration

OneLake eliminates the high barriers to self-service modern cloud data integrations with muting effects—so business practitioners can focus on improving decision quality and enabling better operational processes with 1-2 clicks on no-code or low-code pipelines by managing connections around business contexts and domain terms in domain models. These self-service data pipelines, in no-code or low-code manner, are easy enough to use that, in self-service fashion, internal data users are more empowered to build and extend concrete or implicit integrations themselves around the business scenarios they care about. IT compliance policies are enforced through monitoring, automatic observability, and inherent reusability of these pipelines to ensure low IT risks. It is easy enough to pull and push data in-and-out of external data domains, as well as across internal data domains, on-demand and at regular schedules around a project or a business lifecycle. Self-service data integration is complementary to traditional data integration. Manual data integration is still required when there are malicious intentions, persistent needs, high frequency, mission critical business scenarios, or when organizations want to extract the maximum value out of their data through more sophisticated operations or business domain acumen and brainpower. The business users or the data engineers still need to learn the syntax of the operations, understand how to convert the domain know-how, and build and maintain the complex data pipelines through code. Today existing services are either difficult-to-use or too expensive to enable enterprise-scale no-code data integration. No-code and low-code capabilities are built-in OneLake natively, so even non-technical users can drag-and-drop or write simple business-friendly scripts to extract the right data, integrate them correctly, transform and enrich them as needed.

4. Dataflows in OneLake

Dataflows allow users to extract, transform and clean data from a large variety of sources at scale and push it into OneLake so they have all needed data ready for data analytics. Because OneLake is built specifically for analytics workloads and Dataflows are built specifically for ETL workloads, these tasks become faster and users don't incur unwanted costs by pushing data through a working data warehouse or data lakehouse. Dataflows are built into Power BI workspace experience, and users can warehouse data into OneLake from their dataflows with just a few clicks. The transformation capabilities behind dataflows are also compatible with Integration Factory, so users who want to retrieve dataflows from their production environments can do so.



4.1. Defining Dataflows

Dataflows are a way for organizations to ingest and transform data at scale without having to build a large, complex data engineering pipeline architecture. Instead, they can get the power of a highly scalable data preparation pipeline, a computer which is hosted by the data tool. Dataflows built with integration service are completely integrated with dataflows inside Power BI, so no matter what tool a customer adopts for operationalization of the ingestion pipeline, their analysts always have a logical, single source of the truth. The advantages that cause companies to choose this approach are quick time to build and easy ongoing maintenance, including the reusability of the data preparation steps, the

access of users who are not technical to the data, be it for queries or the use of pre-computed tables, as well as the exploitation of the large amount of data that Power BI desktop and Power BI online run.

Dataflows in OneLake change the notion of what a dataflow is for most users. In most tools, a dataflow is the glue that holds a collection of extract, transform, and load (ETL) operations into a group. As such, dataflows are usually internally focused. They use glue code that is newly written, new code that interacts with the raw data in a source data lake and puts it in the correct format to then be published to a data warehouse or semantic model. Because of this, most ETL tools require a higher level of coding and logical skills than most reports or dashboards. However, using Power BI Desktop with OneLake, any user can assemble these ETL steps into one logical flow called a dataflow. This logical grouping is simply a hierarchical display of the relationship meta data that is automatically generated based on your actions in the tool.

In Power BI Desktop, dataflows are defined by exploring the OneLake folder structure, the same way you look at folders on your own machine. As you navigate through folders, dataflow definitions are automatically created as you explore the directory and discover new files. Relationships are automatically inferred as you reach files within folders with the same name. As data consumer files are processed by the tool, they are auto-discovered and used to hint about other related files in the lake. Additionally, dataflows support incremental refresh modes so that increasingly large files don't need to be entirely refreshed at every interval, and dependency tracking so that some files at the base of the dependency chain can be made out of bounds.

4.2. Dataflow Creation Process

A dataflow is created in the workspace from a set of existing files or an external source with a selected destination. To create a dataflow, the user first selects a dataflow template with metadata that the Dataflow Service provides. Template selection is based on how the dataflow is going to be used. Currently, we provide four types of templates for the four product areas: data preparation, data engineering, data integration, and batch activities. Each of the selected templates provides default settings appropriate to the dataflow's expected use. The user can further edit all of the dataflow properties of the workspace, including those provided by the template.

Next, the user is taken through a guided process to select the dataset. This is a required and the most critical step in creating a dataflow. It affects everything else about the dataflow: its speed, scale, behavior, reliability, and performance.

In addition to other attributes, the dataset specifies the source location and the schema definition of the files. The source location would be a datafile referring to a single file, or a parquet (or CSV) file referring to a mapping file for the entire source location. When the source location is external, it has to be a SQL table or view, and an external source dataset must have a schema definition. During this guided selection process, a set of other dataset types are auto-created when required. Example dataset types are a mapping file pointing to a folder, a parquet dataset pointing to a parquet file, and a linked service pointing to the external source. Once a dataflow is created, the user can edit the properties and schedules of those datasets.

4.3. Best Practices for Dataflows

A few best practices can help you while creating such dataflows. If you're creating a Delta Lake in OneLake, consider defining your target dataflow as a one-time run, defining a single logical copy of the data, so that others can use that data without excessive configuration or worry about the state of the data. If you're defining a multiple run, either delete or rename the existing location, or have the dataflow delete the contents on some cadence. Delta Lakes in general can automatically cleanup old transactions, so not all such practices are necessary.

Make sure you have as little data movement and transformation as necessary. Consider defining incremental data copies instead of complete copies if you narrow the date range as much as possible. If you intend to copy an entire table from the source, consider "raw" copies without transformation, because there is no further overhead. For many sources, this is the fastest and most efficient way.

When scheduling different dataflows, consider the order in which they're executed. Avoid scheduling different dataflows to write to the same physical location at the same time. If you're expecting certain data in certain physical locations, consider monitoring those paths to alleviate execution failures caused by data dependencies. Dataflows do background checks to ensure the execution conditions are satisfied, but it's not instantaneous, and execution may fail due to insufficiently updated paths.

Last but not least, monitoring dataflows to notify you of potentially incorrect data operations helps you be sure your data assets are correct and ready for use. Some sources have interesting default behaviors that you may want to adjust. For example, the Copy from SQL Server source chooses on-demand copying of data unless you specify otherwise. The Copy from File source uses binary copy the first time you run it before intending to read any data.

5. Pipelines in OneLake

Basically speaking, pipelines allow you to declare how data flows in and out of OneLake — pulling in from and pushing out to data sources. Pipelines can run as one-time copy operations, to quickly pull data in or put data out. They can also run as continuously updating copy jobs, designed to keep data in sync every few minutes, hours, or days. They can also run on a Schedule specified at Pipeline creation time. When you run as a copy job, data ingestion from source and data ejection to target happens once. The practical use cases for a copy job are to get a dataset into OneLake so that it can be queried, analyzed, and shared, or to export a dataset from OneLake to a different storage system to continue processing.

Pipelines allow querying data in OneLake without additional pre-processing once data lands in OneLake — without waiting for an update job to run periodically. OneLake Pipeline functions and OneLake query runtimes take care of parsing and decompressing files, interpreting and managing partitions, transforming encoded objects into structured rowsets, performing additional necessary UNIONs to build unified views when data are partitioned by data type, and preserving any computed columns or semantic metadata during query processing. For example, a OneLake user could submit a SQL query to read files with JSON data, read files with Parquet data, and read files with CSV data and produce a single rowset with additional attributes for the source file type.

5.1. Understanding Pipelines

Pipelines in OneLake are the connectors that take data from a data source, transform that data into a usable model, and write that model to a Synapse workspace or to OneLake. In PeopleSoft, Native Connector users will retrieve PeopleSoft data and stage those data tables in a Synapse workspace. In Oracle, SQL Server, and other Native Connectors users will connect to data sources using a SQL form. The SQL form is how users create and define queries to read data from a source system. The pipeline calls the queries and then processes the results. The pipeline transforms data from its original structure into a tabular data model and then writes them to either a Synapse workspace or to OneLake. This module includes several concepts around pipelines including pipeline groupings, pipeline triggers, pipelines for ad hoc queries, and pipeline data processing capabilities.

A pipeline is a process that takes data from a data source and moves that data to a target location. Typically, pipelines perform the necessary conversions to deliver usable data. This illustrates what pipelines do. Data pipelines perform the

necessary operations to deliver usable data models. Models can write results to Azure Synapse analytics workspaces, which are used to answer business questions, or write results to OneLake for further processing. Pipelines make available several data processing capabilities to move data out of source systems. Those data processing capabilities include the following: Stage data from a data source to a Synapse workspace; Extract and model data from a data source and copy the models to Azure Data Lake; Execute an ad-hoc SQL script against a source system; Execute multiple queries defined in an SQL form against a source; and Create and schedule incremental data loading jobs.

5.2. Pipeline Configuration and Management

Data pipelines enable users to assemble source data into a OneLake data repository after which analytics or machine learning can be performed. Later sections will compare and contrast pipelines with other management and extraction tasks not performed by the OneLake pipeline service will be performed using data factory services. These tasks will include the movement of data between the OneLake and other data stores. They will also include orchestration services that will call OneLake pipeline services to manage data retrieval.

When pipelines are used to retrieve data from a source system, the pipeline invokes a source connector that extracts data from the source system. This connector is specified in the pipeline's configuration. Pipelines support multiple trigger types and connectors. It uses the metadata and attributes to discover the pipeline's data movement needs and route data to the appropriate services for setup and execution of the pipeline task. Pipelines have the capability of batching connector calls, which can abstract some of the connector implementation details. For example, after native connectors extract data from a source system, the data is sent to a OneLake location to allow the native connectors to return responses in a timely fashion. This operation may also include additional data enrichment tasks and additional parsing of the source data to be transferred to OneLake.

A source connector is called by a pipeline to invoke data retrieval. This connector is specified in the pipeline's configuration. Pipeline configuration is done through a user interface that overlays the capability and exposes it in a user-friendly way. It attempts to abstract away some aspects of data movement creation, possibly allowing user-defined tasks, such as data aggregation or filtering. It also allows the view of pipeline activity.

5.3. Optimizing Pipeline Performance

The variety of file formats available and used in OneLake creates different scenarios when integrating data. OneLake supports various file formats such as

CSV, PARQUET, JSON, ORC and AVRO, schema on read and schema on write, as well as different compression algorithms for each file format. Different source and target combinations will lead to different ways of improving data pipelines during the data integration workflow. For example, a CSV file written into a Parquet file might rely on an extremely minimal reading configuration while writing into a compressed Parquet column store with complex configuration might require a three-step approach to validate, analyze and write the data.

Transforming millions of JSON files into a single parquet store with column-based compression might lead to memory overflow if no optimization is applied. OneLake features like Hybrid Data Management, Internal HDFS, Indexing Metadata Management, Native Cross-Account Access, Delta File Format infrastructure, Data Fusion and Iceberg Time Travel powered by Delta tables, Custom Business-Driven JSON splitting and Multi-step Custom ETL as a Service pipelines are available for Data Integration / ETL scenarios and can be combined and/or automated together to increase execution performance and provide business value. Combining these features, Ditan can easily address any ETL/ELT data integration challenge to meet any data management, governance and security needs.

Using Delta tables for data governance and transaction ACID compliance is currently the best use of technology available, while data fusion, hybrid data management, and native Cross-Account Data Access can optimize ETL scripts optimally. However, keeping an eye on Delta transaction logs and file formats is key to a successful OneLake, multi-account data management project, and libraries with proper checks are available to avoid pitfalls.

6. Ingestion Strategies

To provide a cohesive, structured architecture and data integration, ingestion really must be performed in a specific manner from multiple sources into OneLake. The ingestion layer captures a copy of the original source data at a specific point in time, ensuring that enterprise users have a data foundation that connects disparate source systems, aligning across disparate sources, with the understanding that source systems might be highly dynamic, with frequently changing structures or rarely changing structures. Ingestion is performed by a variety of ingestion technologies, many of which are updated for performance, efficiency, or functionality on a real-time basis.

The variety of available technologies and methodologies offer significant capabilities for scalability, enabling mass throughput combined with flexible source scheduling requirements, and decreasing overall time to value for analytical insights. Because enterprise analytic workloads often require both hybrid workloads and frequent request workloads, OneLake ingestion must address both methodologies.

There are two types of data ingestion enterprise analytic processes require: real-time ingestion and batch ingestion. The difference is the time model. In real-time workloads, the data from the original source systems are sent immediately after data is generated, or a transaction occurs to consume or change the data from source systems. This is certainly true for ingestion of telemetry events from sensors, sales event promo codes, and numerous use cases involving Internet-of-Things. Frequent updates are also performed logically instead of using what is often a heavy file copy.

6.1. Types of Data Ingestion

There are three types of data ingestion: Streaming Ingestion, Batch Ingestion, and Cloud Ingestion. Streaming ingestion is intended for time-sensitive data that is continuously generated and needs to be made available for analysis as soon as possible. For example, environmental sensor readings, telemetry data, or business transactions could be ingested within milliseconds, or at least minutes or hours of the actual event. Streaming ingestion is typically supported now by File-based Generalized Data Transfer, Message Queue, and REST-based Ingestion.

Batch ingestion, or at least frequent occasional ingestion, is intended for bulk processing of large data sets that may not be immediately associated with a business event but still is required for analytics, typically driven through periodic ETL jobs. For example, executing a command that extracts customer transactions from an RDBMS read replica and stores them in a Parquet file output for business analysts could result in an isolated workload. Batch ingestion typically supports file-based ingestion using Direct Load, REST-based Ingestion, JDBC, and Managed Copy.

Cloud ingestion uses hyperscale vendor integrations for natively dumping files in different cloud storage formats. These vendor integrations provide a one-click solution that can orchestrate the extraction of data from a source store, ingress, and egress the output from the internal storage staging area, and copy it to the target service or data lake. The output stored in temporary cloud storage is typically being used by data processing and ETL tools and/or for analytics by developers. Examples include Copy and Pipelines: Copy Data.

6.2. Real-Time vs Batch Ingestion

Once the types of ingestion are understood, the next distinction is whether to ingest data in batch or real time. The choice of mode is often driven by the use cases because not all scenarios demand a constantly-updated lake. In addition, there are practical trade-offs that need to be made at the organizational, engineering, and financial levels. Batch is the traditional method. It is simple, inexpensive, and has been the choice of data ingestion for over three decades. A common pattern is to ingest data every night, or even less frequently. This works well for batch analytical workloads and for ingesting data that is not highly volatile, or for which there is no business problem from having several hours of data latency. As the lake becomes a source for increasingly more real-time use cases, people turn on change capture. This allows for near-real-time batch ingestion but incurs higher costs to both the ingesting and the lake processing infrastructure. The data pipeline should be developed with a level of modularity that allows for starting with scheduled batch ingestion and later enabling change capture, or for switching on change capture for short durations to meet business goals without major rewrites of the pipeline.

Real-time ingestion is more complex and expensive. Real-time ingestion must not only collect the data, but also often perform data transformations in the data pipeline itself. It requires more processing resources, including payments for resources that are always available, compared to batch ingestion, where resources can be allocated in a reservation-style manner for the period in which batch jobs are executing, and de-allocated at night. Creating and managing a high-performance end-to-end real-time data pipeline is significantly more complex than batch. Both batch and real-time ingestion have their use cases and have been used by companies worldwide to meet their analytical and operational needs.

6.3. Tools for Data Ingestion

Regardless of how ingested data arrives at OneLake, it is natural for users to want to know what tool or native Azure component will help them do it. This section captures several of the ingestion tools available for use. These tools combine both Azure-native and Azure-partner capabilities to provide complete support for customers' needs. It includes light mention of the categories of ingestion tools available and common uses of each category. There are a few key sources of user requirements that drive the range of options available. It is important for analysts and partners to understand those classic requirements better than anyone else in the industry. This is essential to serving the diverse data sources that customers encounter. The tools span managed service from Microsoft that abstracts away

configuration complexity to industry-standard connectors that might be utilized more efficiently with heavy enterprise deployment.

The simplicity and capability of the tools exposed for ingestion is a prime consumer requirement. In the case of Windows tools, it should be possible to ingest any source from an upload of a set of files on local disk, Azure, or OneLake with a right-click context menu. What makes this easy is that the tool would determine the format and structure of the data based on the contents of the data. When uploading data from a view, most users will prefer to generate a Delta Lake file to OneLake compared to running a query to copy the result back into the tool and then uploading that. Assigning and tracking certain types of metadata are also very helpful for both scenarios.

7. Case Studies

Various organizations in the Microsoft ecosystem are exploring OneLake capabilities and architecture. Organizations have decided to test and develop OneLake as part of an exploration of the best options for managing data more effectively in the cloud using the Microsoft technology ecosystem to help drive more efficiency into their cloud data solutions. A large global ISP designing data analytics for the Network with Dataverse and Power BI was an early tester for OneLake's ability to manage complex data architecture for products like org data that may not be used every day but provide actionable insights to the business on a lower cadence. Many customers are testing OneLake either to add a simple to manage Data Lake with their Cloud Data Warehouse, or to migrate functionality or data from on-premises solutions to a Microsoft Azure Cloud solution that can cheapen and bulk up the structured and unstructured data they have. Lessons learned in these implementations, and their ongoing nature, are useful.

The future of Data Architecture needs new designs to facilitate ease of use and reduce costs of delivery [1-3]. The establishment of OneLake as a core Microsoft cloud element allows enterprises and service organizations to quickly gather data of varying classes into a base that has clear storage efficiency, security and delivery attributes so that all Microsoft capabilities can act on the data. The delta has a clear path from storage readiness, through codifying the environment with Data Governance, to data sharing visualization and preparation using capabilities into an accessibility format for business users. The eventual exploration of formalized Microsoft ML support should be incorporated as a foundry for ML

processes once available – as they will have similar principles and model compatibility, as well as ease of business adaptation once formalized.

7.1. Successful Implementations of OneLake

Data integration has been a focal point of enterprise architecture since roughly 2000, when the term Enterprise Data Warehouse came of age. Over the last two decades, we have seen myriad tools and processes for enterprise data integration come and go. Business pressures for making better and quicker decisions have fueled increasingly sophisticated implementations of data integration, from simple nightly extracts of key operational metrics from transaction systems to near-real-time integration of online systems with predictive analytics. The integration tools used vary widely, from programmed batch loading to enterprise scheduling frameworks to hybrid ETL tools to data virtualization to on-demand data integration tools. Businessly-driven data marts, on-demand data discovery, mobile applications, and AI-powered analytic insights have added complexity to the data ecosystem.

In addition to evolving patterns of data integration, advances in cloud infrastructure and cloud-based services have accelerated the evolution of data integration [2-4]. The need for data security has never been greater, and hybrid multi-cloud security standards have radically changed how enterprises use cloud infrastructure as integration capabilities commonly known as "one lake". OneLake is critical for enterprise data fabric, enterprise-wide business intelligence, and enterprise data domains being called "one source". OneLake is unique. There has been a change in how companies think about OneLake. Enterprise Security requires a change in user access, development, startup, and maintenance of OneLake. Data source connectors need to be used prior to data movement into OneLake, where shared data is discoverable.

7.2. Lessons Learned from Data Integration Projects

Many organizations have invested significant effort in integrating data across disparate systems, often investing millions of dollars into integration technologies and projects. The experience in many of these projects has contributed to the development of a number of lessons learned:

1. IT Alignment: Data integration projects require the full alignment of multiple IT groups. Each group has its motivations, focuses, and priority systems. Any disunity or miscommunication will become painfully clear as the implementation nears completion.

2. Business Myopia and Aversion to Change: Business groups often spurn proposed integration projects, preferring to remain with local solutions that are quicker to build, easier to modify, and less offensive than changing existing implementations which are often heavily used. They often feel that the proposed solutions fail to address their areas of need. Document requirements well and document solutions well.

3. Integration Doesn't Change the Why: The result of integration does not reduce the need for business change. New data may expose business process deficiencies. Give business groups the support and latitude needed to address their processes; they already have your data mess; the last thing you want to do is support their present state. Each integrated data model should be examined and approved by the appropriate business team, with support from the data governance group.

4. Product, Give Me a Product! Business groups want input to the integration process, and they want closure. The last thing you want is business groups bringing modifications requests weeks or months after the data has been released into production.

5. Dare to Dream: Create data models for integrated enterprise data, even at the risk of disagreement and/or criticism. Be bold and supportive as changes are addressed. Integration is critically important to success; without it, there can only be limited visions and limited outputs.

8. Future Directions

In Data Engineering, we see two important trends emerging. The business landscape changes very often: there are M&A, new regulations, and new business ideas that spark high interest. That leads to new data sources coming into the corporate data landscape. At the same time, the cost of technology continues to decline while technology offers more possibilities. When we look at data integration, we see strongly enabled IT, e.g. solutions are popular among business users who can deploy them to solve their issues. Budget constraints lead to cost cutting: everything needs to be cheaper. Classic data integration solutions become costly, so a more agile self-service alternative is desired and needed by many organizations.

Research and vendors alike focus on new enabling technologies. Serverless technologies can make data integration easier and cheaper. Thanks to big data,

we can store anything and everything. Data lakes and data lake houses layer the technology stack. Data virtualization connections allow access to data without preparation. The need of governance leads to a focus on metadata technology. Data preparation for self-service analytics is getting interest from users. All this leads us to the question, where is Architecture going? While it offers a three-cloud-sharing-based solution today, tomorrow's cloud solutions will become highly interoperable. The integration landscape will merge down to four or five true best-of-breed on-demand services, run from any cloud, also for any cloud. The Architecture is a microservice that creates these clouds and virtualizes the other clouds. It becomes a data integration control plane. Data augmentation moves into OneLake: On-demand integration is automated via integrated metadata matching and execution of the tasks is monitored and enforced.

8.1. Emerging Trends in Data Integration

Data Integration is changing. No matter if you select the traditional ETL approach or the ELT approach introduced by Data Lakes, new trends changed the way Data Integration is approached and executed. Data Integration was the result of manual processes distributed across IT and Business Executives. Data Integration is now provided as a service capability with functional flows that can be implemented by Business Users. Cloud Native computing has added scale, speed, and cost-effectiveness to Data Integration processes. Finally, the introduction of increasingly complex analytics algorithms requires increasing Data Integration capabilities.

Self-Service Data Integration allows Business Users to connect their business domains and create the Data Fabric that makes possible the efficient flow and integration of data across a diverse set of applications and data domains. Without this ability, supported by the emerging Data Fabric solutions, Business User processes become fragile. They cannot be operationalized as services as needed by corresponding analytical algorithms running in IT or Business User environments.

The switch from low-code to no-code Data Integration solutions has become a must-have feature of Data Integration engines because Business Users are and will be the primary users of these services. With the Low-Code tools fueled by accelerators created in IT environments, Business Users are able to create and maintain Data Integration tasks. With the No-Code capabilities, most of these tasks become automated and only require the attention of IT when events occur that introduce variations to Data Integration tasks such as Storage performance issues, changes in quality of source data, and changes in query plans and/or algorithms.

8.2. The Future of OneLake Architecture

As for the OneLake architecture, that is still evolving. The simplicity of an endpoint with onboarding workflows that can be built with no or low sophisticated skills is appealing, especially if it is simple enough to democratize data curation. We are working on an easier consumer interface. There are three key functionalities that are being worked out: access control, recommendation and enrichment of existing content through data-native AI models and better workflows to enable quality curation by power users who build and enhance templates for data harmony. The first use cases from customers support the vision of OneLake being a stage placement for external data, prior knowledge embeds in models serve as the important quality gate combined with curation workflows. Customers see benefits in the speed between stage and AI ready data, driving a frictionless experience for the community. Those benefits would naturally drive the pattern into hybrid solutions whereby sourced datasets without any transformation would remain in external stages while data blended and harmonized would move into OneLake.

Implementing cross-cloud, multi-data provider vendor-agnostic patterns allows us to create win-win partnerships with all potential data partners. By moving files from customer's clouds to the low latency global data center we would enable a new hybrid data experience marrying the notion of data space from OneLake with the underlying notion of data mesh coming from the greater data ecosystem. The above ensures retention as a focal point, once modeled, we can guarantee easy data amalgamation through a rich direct connector ecosystem on top of other vendors and clouds. The data-driven organization firmly stands for one version of the truth but be it the landing for any multi-data cloud partnership refers to trusted roles and responsibility along with a low-latency process to ensure data quality. This pushes the democratization of data engineering while preserving the security lines set along the flow at point of entry.

9. Conclusion

In this chapter, we introduced OneLake, Microsoft's content lake. We described the architecture of OneLake, focusing on the simplicity of its design. We argued that the architecture of OneLake has the key dimensions needed of a content lake, including simplicity, searchability, and integration with tools. We provided a deep dive into how OneLake integrates with Microsoft products for building enterprise pipelines and for content management. We showed that thanks to the shared architecture and the integration with Microsoft products, OneLake can

integrate heterogeneous data efficiently and easily. We also described how third-party solutions can be integrated with OneLake.

Looking to the future, companies will have to manage convenient, trustworthy, desirable integrable data lakes for content streaming solutions that their business data scientists, business domain experts, and software engineers can utilize. To achieve this goal, it is crucial to distinguish between content lakes – that use a simple category-key-value data model for direct data storage and management – and content data lakes – that use key-value architecture for data lakes based on a traditional data model like relational schema-on-write, and a second traditional schema-on-read data-validation step for making data discoverable, searchable, trustworthy, and desirable, and for speed-efficient Fast Data, and the near-real-time updating of main-cohort near-real-time user-facing pipelines.

References:

[1] John, Tomcy, and Pankaj Misra. *Data lake for enterprises*. Packt Publishing Ltd, 2017.

[2] Inmon, Bill. *Data Lake Architecture: Designing the Data Lake and avoiding the garbage dump*. Technics Publications, LLC, 2016.

[3] Sawadogo, Pegdwendé, and Jérôme Darmont. "On data lake architectures and metadata management." *Journal of Intelligent Information Systems* 56.1 (2021): 97-120.

[4] Microsoft. "OneLake in Microsoft Fabric." *Microsoft Learn*, https://learn.microsoft.com/en-us/fabric/onelake/. Accessed 7 July 2025.

Deep Science Publishing, 2025
https://doi.org/

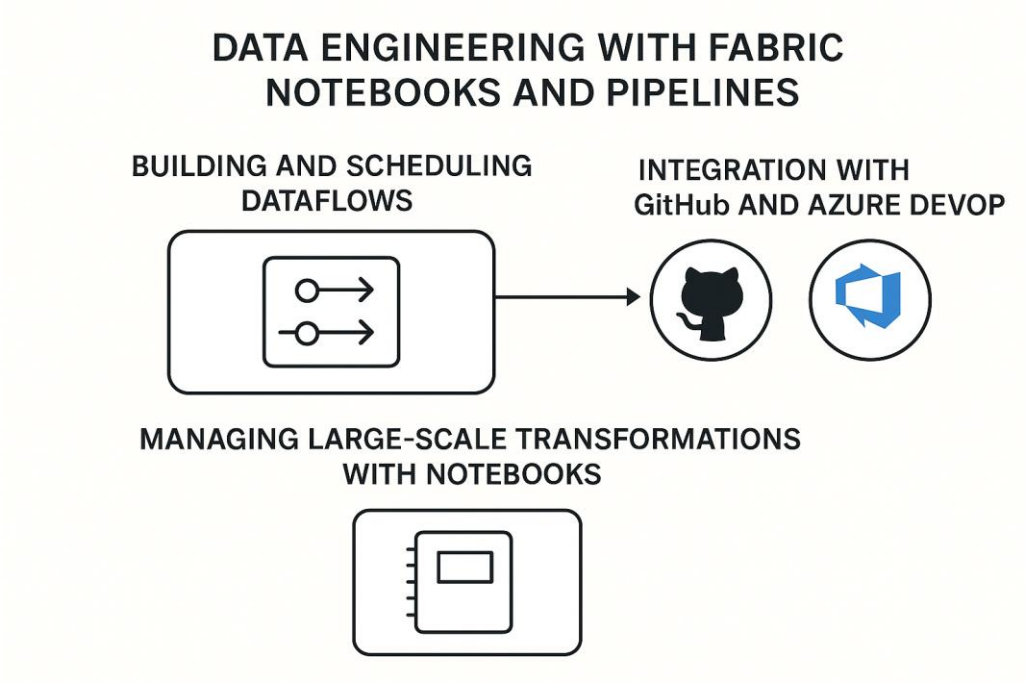
DeepScience
Open Access Books

Chapter 3: Data Engineering with Fabric Notebooks and Pipelines

1. Introduction to Data Engineering

The term data engineering is not widely used in general books about data. From a general point of view, data engineering is seen as a subfield of software engineering or data science. The term seems to have originated from the industry more than a decade ago, and now it is pretty well established among business enterprises, big and small. This discrepancy is, however, not surprising since the development of data engineering technology has been initiated and championed by data technology companies, followed by big enterprises realizing their power and making use of them. Meanwhile, however, a broader range of small enterprises, and eventually even startups, have started to use the data from their customers or business processes, first to build additional value on top of their offerings, and then to leverage their data to monetize it in its own right.

Industry analysts are projecting that the worldwide market size of data engineering, at around 4B in 2021, will continue growing at a compound annual growth rate of around 25 percent to surpass the 20B market size threshold by 2025. In the early 2000s, the term was even used to describe a new class of database management systems that relaxed the OLTP speed principles of relational databases and allowed the modeling of new types of dimensionally based data warehouses for big data and analytics, along with hybrid approaches such as relational and SQL/Cube extensions.



2. Building Dataflows

Microsoft Fabric products simplify the process for building rich data pipelines. Having all the building blocks available in one product works particularly well for dataflows that cover a wide range from ETL to ML.

2.1. Overview of Dataflows

The heart of data engineering is moving data around your organization, making it accessible and usable to your colleagues. You create and schedule dataflows to extract data from a source location, apply whatever mapping and transformation is needed, and return the processed data to a target location in your organization. Fabric provides a unified experience for building dataflows, but the underlying operations and architecture are quite diverse, reflecting the diversity of data engineering use cases. Each type of dataflow represents a unique operation – whether copying data in different formats or managing data automatically – designed to address a specific kind of need. More specifically, there are three types of Dataflow operations that you can schedule for periodic execution, two of which are powered by Spark batch compute: Spark Copy, which is a fast and flexible high-throughput copy; and Spark Dataframe, which represents a custom data transformation and data engineering code that you write and maintain as a

Fabric notebook. The third category of Dataflow operation, Pipelines, provides a low-code, auto-parameterization experience for building and scheduling pipelines for ELT workloads. Pipelines connect pipelines steps, managing orchestration, dependencies, triggering, and scheduling automatically for you. Each pipeline step gets data from a source, queries, filters, and transforms it in some way, and passes the result output data to the next step in the flow. Automating pipelines is crucial for ETL purposes and various other tasks, including data cleansing, validation, and coalescence, loading raw data into physical stores, and propagating and harmonizing changes between digital twins.

2.2. Designing Dataflows with Fabric Notebooks

Just as data scientists can now explore data in notebooks, data engineers can build dataflows in notebooks. This section describes how. Data engineers build a dataflow using notebooks of AI services and a set of pipelines that implement the logic to take input from multiple datasets or trigger-based events, perform some logic that includes machine learning or some data manipulation or data tracing, and produce one or more datasets. A pipeline may produce datasets as its final output or may undertake other tasks like updating a database automatically, ingesting data to a target system, notifying some stakeholders, increasing or reducing resources, etc. You can run pipelines from different originating events or once they get triggered using a periodic schedule.

All feature notebooks and notebook cells that use Spark language. You can use them to load or create tables from input datasets, call pipelines to implement the dataflow logic, and save the resulting datasets for the downstream data consumers. The notebooks also allow you to build interactive visualizations to profile the input datasets used in dataflows to perform data quality checks on the datasets going to the consumers. You can run the notebooks interactively to run code that may require access to external services without setting up triggers. When the notebooks are part of pipelines, they are used to convert the files into final datasets, which are then transformed or loaded into a target system. You can also schedule the pipeline periodically to implement some automation requirements.

2.3. Best Practices for Dataflow Development

A notebook is a great place to prototype and deploy a simple, recursive task that is only responsible for doing a single, atomic action. However, once you want to implement a more complex solution that coordinates the execution of multiple dependent actions, it is better to define a pipeline instead. This is because a

pipeline provides great features like elegant dependency management, automatic scaling, and the ability to schedule and access task executed logs easily.

When creating notebooks and pipelines, there are a few best practices to follow. Since a pipeline job can execute multiple notebooks that process different resources concurrently, it's beneficial to separate the code that processes all your different resources into separate notebooks. After that, you can set a task in a pipeline that orchestrates the execution of each of those notebooks with the parameters necessary for each notebook invocation. Ideally, a notebook should only have a small number of code dependencies, and the less dependent a notebook is on another notebook, the less coordination needed when executing notebooks as part of a pipeline job.

The execution of the notebooks can be made completely independent by specifying the parameters in the notebook task configuration screen. For more advanced scenarios, if you need to pass complex data types between the tasks of a pipeline job, you can specify any of the notebooks to be executed beforehand as a state-saving task, and the output of the corresponding task will be automatically persisted into and read from a TempTable and TempTableRead respectively.

Beyond the above considerations, the more modular and reusable your code is, the more likely that code is going to be neater and easier to maintain. Reusable code can also be easier to debug if errors arise since you can unit test the code within your self-contained modules or functions. A function can also provide a better abstraction level so that any user of the function won't be concerned with the byzantine internals of how the function performs the business logic.

3. Scheduling Dataflows

Let's see how to create a scheduled dataflow using a notebook from the previous chapters. To create your first pipeline, navigate to the Pipelines tab on the left menu. There, you can find two buttons: the "New pipeline" button, to create a new pipeline, and the "New folder" button, to group pipelines into folders. Click on the "New pipeline" button to create your first job. A window will open, prompting you to select a notebook to run. Select one of the notebooks you have created previously. Then, you can specify the pipeline parameters and, importantly, select the trigger. A trigger is what is scheduling the pipeline.

3.1. Introduction to Scheduling

In the last chapter, we have seen how to create interactive notebooks with various kinds of code. However, if we want to ingest data for a data warehouse or data lake for a production deployment, it is usually needed to create a dataflow based on a notebook and then schedule it to run automatically. A possible use case could be a scheduled task that downloads data from an external source on a daily basis, loads it into a database, and then runs a stored procedure to send emails notifying users that the relevant data is ready. Or maybe loading a sentiment score from certain keywords into a table in your data warehouse in order to connect it with the other available data on your business and help decision-making.

In Microsoft Fabric, it is very easy to create such a scheduled dataflow by using Fabric Pipelines, a low-code feature that allows you to schedule and monitor lots of different scheduled jobs. Pipelines allow you to group different types of activities, such as Data Flow activities for designing ETL/ELT processes, Copy activities, Data Flow activities to build custom data transformation processes and store the resulting data into a database, or other notebook activities. Pipelines can simply be a wrapper around scheduled notebooks, or you can combine tasks built with other activities into one single operation. You can then schedule your pipeline to run periodically.

3.2. Using Fabric Pipelines for Scheduling

Despite being spelled "pipeline", the contract we use is declared as a function called “schedule” that decorates a custom function called “flow”. The purpose of a data engineering pipeline is to orchestrate the execution of “flows” that manage data from a data source. This flow is able to transform the data, load it elsewhere, and cross multiple sources of different natures. In addition, it can call several notebooks that do specific tasks in a Data Engineering workflow. For notebook management, the APIs allow you at any time to execute, stop, delete, reset, or change the parameters given to each one. From a software development perspective, we can say that having a dedicated library to schedule is a step forward to apply the DRY software development principle. When the notebook task is repeated in a flow or data, it is better that this task is built as a separate notebook and later on called by the main pipeline.

“Pipelines” are the main entry point to schedule data work. Pipelines bind together multiple notebooks and functions that run any type of workload. Pipelines can run anything from a simple task executed within a single notebook to a complex orchestration consisting of a sequence of data preparation operations performing fast ETL calls to a series of external SQL-based tasks that

are activated on demand based on the incremented behavior of external files. They allow some level of both horizontal and vertical orchestration. They run notebooks, run functions and use waits and condition checks to control the flow. In addition, "Pipelines" can execute in a pre-defined order and/or based on the result of a previous task, and/or based on the end of a previous task.

3.3. Monitoring and Managing Scheduled Dataflows

Scheduled dataflows in Fabric run in the background and monitor the trigger or schedule but they are not like a web service that you monitor constantly nor like a client application that you check after every run. You monitor and manage the scheduled dataflows from the Data Engineering workspace Dashboard. You see the overall status, success and failure metrics on the Dashboard and detailed status, error messages, re-running and deleting operations from the logs panel.

The Dashboard shows the overall status and metrics for the scheduled dataflows on visual cards. You can switch between two Analytics views – the scheduled dataflows view and the connection errors view. The view selectable from the dropdown menu shows the visual cards. The pipelines view shows metrics in two sections – a top section with metrics for all scheduled dataflows and a lower section with the visual cards for the individual scheduled dataflows. The connection errors view shows metrics related to connectivity errors. In the individual scheduled dataflow visual cards section, the background colors help distinguish the status and metrics related to the successful completion of runs, the status of the last run which completed unsuccessfully, or the status of the last run which is executing at the time.

All the serious errors which have led to aborted runs, shown in red squares on the visual card, can be seen from the logs panel. You can choose to rerun an aborted scheduled dataflow either for debugging the errors or for reaping the missing data. Deleting a dataflow means that its pipeline is no longer scheduled to run periodically nor when it has been disabled. The periods and timestamps for which the dataflow has executed and the last invalidating errors for the current period can be seen in the logs panel as well. The dataflow visual card also helps you know if the dataflow has linking scheduled refresh errors.

4. Managing Large-Scale Transformations with Notebooks

Building production data pipelines may require performing data transformations at a large scale. While many client libraries allow you to do that from your local computer–creating Storage Accounts and Data Lake storage containers, uploading some files, and passing them to the Ingest API, for example–that's not how we want to run jobs in production. Python notebooks run data transformation jobs at large scale, taking advantage of Data Lake. Notebooks provide a mechanism for building and running pipelines through cells, code snippets that define a step in the pipeline. This is a crucial point, because traditional big-data frameworks didn't provide solutions for handling mixed pipelines, where some parts were run in a notebook without a specified order–cells do have some dependencies, like the need of writing data to storage if they want to be read–while others were accessed through APIs, such as for inference or online update of machine-learning models.

Notebooks were quite successful, and other systems have adopted their style. Notebooks, for example, are integrated with their jobs API, which allows scheduling. In Fabric, scheduled jobs are Pipelines, that allow you to chain notebooks together, passing parameters for one to the other, and joining them together as directed acyclic graphs (DAGs), creating more complex pipelines.

We have seen how magic commands are used to run queries, as they are run inside a session. This may be inefficient for interactive exploration, where you want to run many queries with different data slices without waiting for a job to finish for every query. However, when running pipelines, it is preferable to use session functions, for when you are running many different queries.

Notebooks, by design, allow interactive exploration and debugging. As they are running code in a shared environment, variables initialized in previous commands can be reused. You can add print statements or use a logging module for monitoring your notebook execution.

4.1. Understanding Large-Scale Transformations

Data refresh cycles for dashboards and reports need not only aggregate and connect to a few tables, but also transform data from external applications to make it easier for business users to understand. Large-scale transformations involve both significant data volume and complex, computation-intensive transformations. Often, macro decisions about products, finance, sales, and

competitors are based on complex dashboards that aggregate billions of rows of data from multiple tables in the data lakehouse and perform business logic calculations using large semi-structured columns. Some transformations require a very large amount of compute resource and run into resource contention problems with other systems during their execution. Examples include high-end models that pull data over external data source connectors in a pipeline.

Using a notebook created with the Markdown editor, data engineers can develop, test, and refine individual business logic components in smaller. Then combine these components into a larger model without having to run through the entire execution path each time. In addition, using a Notebook enables data refresh developers and business users to quickly review and validate business logic task output using cell-by-cell execution and built-in visualization tools. This approach avoids resource contention and management time spent on debugging, while still meeting business user expectations for output recency.

4.2. Optimizing Performance in Notebooks

Don’t use for loops over large datasets. In notebooks, you can write code using iterators and for loops. However, there may be a significant loss in performance if you use these operations on DataFrames involving a for loop. Python is an interpreted language. Each command in the for loop is executed one by one, and it needs to call multiple libraries, which can be expensive. Wherever possible, write your data transformation logic either by applying functions or using native functions provided by the library to improve the performance of your notebooks. In some cases, executing commands or functions is a good alternative.

Use to perform numerical operations on DataFrames. The performance of for DataFrame applications is reasonable; however, when you want to do fast numerical computations, you should consider using functions. Functions are known to be up to 50 times faster than similar operations done by or lists in Python.

Use or for DataFrame libraries if you need to deal with super large datasets. You can use APIs similar to that of , which are useful for parallel computations. When you need to utilize the complete power of your computer cluster and execute your task into multiple cores and machines, go for API. However, if you need to apply your commands to a single machine, use , which uses multiple cores and performs parallelization better than .

Train Machine Learning Models on Sampled Data. Running hyperparameter tuning to get the best accuracy for machine learning is typically a time-consuming task. If the size of your input data is reasonable, it will take hours to

run a modeling task on a full dataset. In these cases, if you plan to use boosting-based models, you may perform the hyperparameter tuning by sampling the data and run the final model training on the full dataset.

4.3. Error Handling and Debugging Techniques

A common concern when performing ETL processing with notebooks is that if the transformation fails, we may not want it to disturb the resulting data because of the commit and rollback semantics of SQL operations. The solution we provide for this is that the transforming function needs to implement the Spark in-memory error handling patterns by properly catching exceptions, writing failed partitions to an error data lake, and keeping track of any successful results. Most of our work is implemented in the NotebookPartitionPipeline class but is kept transparent for the user.

In addition to the action-based error handling, we provide a batch-based method for the data engineer to use during interactive development stages that wish to debug code that uses the Transform-and-Display pattern by being able to execute a batch of many operations of the change result together in order to examine the result and catch any exception raised. This requires not having the need for the original file to stay available and unchanged after operations. The need for needing to re-import the original file after every batch affects the performance of this while analyzing code and functions.

5. Integration with GitHub

Fabric Notebooks provides support for integration with GitHub. It allows shared projects to have version histories and additional collaborators to work on the same Fabric Notebooks project code. Fabric Notebooks directly connects to GitHub and enables you to create GitHub-hosted code repositories without running any CLI commands. This integration is helpful to Data Engineers, Data Scientists, Business analysts, and other data practitioners using Fabric.

To enable GitHub integration for a Fabric Notebooks project, any member of a Fabric workspace organization can authenticate Fabric's GitHub service connection in the Fabric console. Each project in Fabric Notebooks is stored in a Fabric workspace, similar to the notion of an organization in GitHub. Fabric uses the GitHub organization or account name as the Fabric workspace name.

Upon GitHub integration setup, the Fabric Notebooks service connector creates a new GitHub organization with a GitHub-hosted code repository for the

corresponding Fabric workspace organization if it does not already exist. Workspace members whose GitHub accounts are linked to the Fabric workspace organization can choose to authorize Fabric for additional capacity. This allows direct access to Fabric services for Data Engineering and Lifestyle Management using APIs and webhooks. For users who are members of both the Fabric workspace organization and the GitHub organization authorized for hosting, there are Fabric console options for selecting shared repositories and importing code for shared Fabric Notebooks and Pipelines.

After GitHub integration is enabled, users associated with the Fabric workspace organization linked to GitHub accounts can create new Fabric Notebooks projects or open existing Fabric Notebooks projects directly from GitHub-hosted code repositories. All code and structure in these GitHub repositories comes from the Fabric project. When you create a project, the project code repo will be initially populated with base files required for running the notebook, and the repo will continue to receive updates for subsequent commits.

5.1. Setting Up GitHub Integration

The integration with the Git options in the dialog sidebar of Jupyter notebooks allows users to seamlessly push (export) or pull (import) notebooks between Personal Google Storage and GitHub. You must have a Windows machine with Git available. Make sure to create and manage Personal Access Tokens in the User Settings beforehand.

Importantly, to be able to use the Git options in the sidebar when a notebook is opened, you need to add a GIT_TOKEN variable into the Environment section of the Kernel Settings. Replace your_token with the token you created in the previous step.

Then in this section, you will be able to pick the file from the repo you want to pull/push to using the option buttons, and it will just work from there! Notebooks saved to GitHub will be saved in .ipynb format with a .ipynb file suffix while dumping, and rendered into an .html file with a .html suffix upon pulling so that you can view the git history more easily.

These files will be saved in a folder called pkg in the root directory of your GitHub repo. You can set it up to whatever else you want, as long as it is a valid configuration, by modifying the environment variable, GITHUB_DIR. It is highly recommended to keep it as is. The GitHub repo must initially be empty without a README, .gitignore etc. files. If not, it will throw an error on saving. More specifically, this feature is built on top of the API for Pages. Therefore, the chosen dummy file, when pulling, cannot have any content at all, and when

pushing, it can never change or be modified (all commits will refer to the same unchanged dummy file).

5.2. Version Control for Notebooks

Traditional version control relies on storing changes as text. This works very well for code, but Jupyter Notebook files encode code, markdown, and cell state as JSON, and the JSON structure is difficult for humans to decipher. That makes it difficult to use traditional version control to track and manage changes in notebook files.

The Notebook Diffing Tool solves some of these challenges by providing support in Git for diffing, merging, and handling of Jupyter Notebooks. While it doesn't change how versions are stored in Git, it improves how you and your collaborators interact with versioned notebooks, making the experience concise and informative.

The tools help with the common workflows you might encounter while using Git with notebooks:

diffing – If you want to see what changes were made to a notebook between commits, or compare two different notebooks, it provides commands tailored to that use case. *merging* – When multiple contributors change the same notebook, you may encounter merge conflicts. It implements a merge driver to simplify the resolution of those merge conflicts. *Git command-line* – When you run git commands, such as git status or git checkout, you can have it display notebook information in the output of those commands. *notebook storage* – If you want to have git store a more compact or human-readable version of your notebook, it includes a pretty-printer that can be used for that purpose.

5.3. Collaboration Strategies Using GitHub

Collaboration requires communication. This can happen in various ways including task assignment, signaling code changes, with and without asking for help, sharing state information, and other forms of constant feedback. In this section we explore these topics in the context of Fabric support. We describe code changes in Fabric Pipelines and Notebooks, then summarize tools that support group collaboration. We then list different people roles during Notebook Pipeline development, and propose software development group collaboration strategies.

The most common form of code change is a pull request. Any user can create pull requests to a Fabric Repository with a Notebook or Pipeline modification. Fabric includes an option to auto-create pull requests when saving Pipeline or

Notebook modifications. Pull requests can be assigned to one or more collaborators and have comments associated with them. They can be created from commits to different branches, providing a version branch for each task. Pull requests allow an individual developer to submit a proposed code change, and then have that code change evaluated before being merged, either automatically or manually. Useful information on code quality is available in the form of Notebook status pages, Notebook Code Quality Indicators, and Notebook Visual Diagnostics Tools. The code change can then be reviewed, with the reviewer affirming or questioning the change. The pull request is a request for help: help reviewing the code, feedback on its correctness, and assistance if the code needs to be changed or debugged. In some cases, the reviewer can go out of band and test the proposed change as well. The Team Owner can if necessary reject the code change.

A standard Repository can be used as a primary Module Repository. However Formulas cannot be called in custom Notebooks in Notebooks Module Repository because the Integration does not use familiar Modules conventions. This also applies to secondary Formulas Module Repositories or Repositories that are not available over the network.

6. Integration with Azure DevOps

Azure DevOps is the software development service by Microsoft, that was designed to be a complete solution for managing DevOps projects. It is a collection of SaaS tools that helps plan, develop, test, deliver, and monitor software projects. Projects in Azure DevOps can be public or private, and can host open source or proprietary codebases. Public projects can have unlimited collaborators, while Private projects have features available based on the subscription. The features of Azure DevOps include dashboards, teams, pipelines, repository, artifacts, boards, test plans, and wikis. It supports cloud and local deployment models, as well as various integrations with other tools.

In Azure DevOps, a project is a collection of tools and the artifacts represented by those tools. Projects enable user access control, and logical separation of resources. Azure DevOps supports repositories that can be either Git repositories or Team Foundation Version Control. Projects can also contain built pipelines to automate building and testing projects, and release pipelines to automate deploying code and various artifacts.

Continuous Integration/Continuous Delivery (CI/CD) is a software engineering methodology that defines an optimal workflow. CI/CD is a set of modern development practices that combines Continuous Delivery and Continuous Integration which helps software developers and IT operations teams reduce the risk of delivering changes by allowing for automating all steps of software delivery. While the original CI/CD concepts only applied to software engineering, they have been adopted by the Data Engineering and Data Science teams to build Data Engineering Pipelines and ML Deployment Pipelines.

6.1. Overview of Azure DevOps

Azure DevOps encompasses a suite of Development and IT Deployment Services needed to increase orchestrate and validate Data Ecosystem Development Ops enhancements and fixes through build, release, and orchestrate workflows and pipelines. Azure DevOps consists of Development, Test, Implementation, and Support Services rolled into one easy-to-use integrated web interface. Azure DevOps enables organization teams and projects to collaborate closely, increasing shared understanding of tasks and reducing reconsideration of latency for development task execution timelines.

We use Azure DevOps to establish a tightly integrated Continuous Integration, Continuous Deployment, and Orchestration workflow. These data ops workflows for Data Orchestration Code, Data Clean Data Quality, Data Process Data Engineering, Data Science ML, and Data Fluent Documentations for Data Analytics services enable us to Operate the data ecosystem services and implementations. By bringing together work items planning, peer code review impact for documents and services, feedback, and documentation generation inside a single web-based and API driven collaborative ecosystem. Azure DevOps enables us to build quickly, test intelligently, and release confidently using resources in any combination of on-premises and cloud environments. We rely on Azure DevOps to version control artifacts and establish build, test, document, publish, and deployment pipelines for all Ecosystem Document Artifact processes, including EDA, Data Cleaning, Documentation Generation, NarrativeML, and PKM Services, across all of the various cloud services and on-premises configurations of EDA, Data Cleaning, Document Generation, NarrativeML, and PKM Services.

6.2. CI/CD Pipelines for Data Engineering

CI/CD is a combination of best practices and tooling that make it easy to rapidly and reliably deliver small but iterative enhancements to production applications. CI stands for Continuous Integration, which focuses on build automation and unit

testing. When engineers implement code for the application or data engineering pipeline, as soon as it merges into a code repository, it is automatically built, deployed to a test or staging environment, and unit tested so that code is working. Additionally, the construction of buildable artifacts may also occur.

If CI is about detecting problems as soon as possible, CD or Continuous Deployment is about deploying to production rapidly without human intervention. There are many data preparation pipelines that get production datasets ready for consumption by the enterprise. At this point, the cleaned, transformed, and ingested data are at rest. However, once a data pipeline is wired with a monitoring service that can detect how fresh the ingest is and fire an alert if it is stale, it is possible to convert this data pipeline to a CI/CD pipeline that will run when a production event is triggered. Updating data at higher frequencies is now considered best practice since it gives analysts and data scientists access to newer records.

For critical data at enterprise scale updated frequently, enabling push CI/CD support requires working with the infrastructure and resource management services that schedule the execution of the pipeline [1]. Two commonly used strategies to push enhance pipeline execution frequency for event-driven pipelines are to leverage existing serverless compute or schedule worker nodes to run as needed.

6.3. Automating Workflows with Azure DevOps

Idle Cerberus is an example of an Azure DevOps pipeline for scheduling Notebooks on Fabric. You can read, validate, modify, and schedule Notebooks from Fabric in a CI/CD pipeline without any prerequisites. Any changes to Notebooks and pipelines would be reflected in your Fabric account as a new scheduled or modified pipeline. The Notebooks and the pipeline can be in JSON or binary forms. The Azure DevOps pipeline uses the Azure CLI command for fetching the JSON format and the Azure REST API for fetching the binary format of Notebooks.

The pipeline will have one or more steps that determine what options you want. The options to install the Azure CLI and connect to your Fabric account are mandatory. The rest of the options are optional. The Azure DevOps pipeline will load all the environment variables from the variable group. The pipeline uses these variables at runtime. The task will install the Azure CLI in DevOps first. This is done because the Azure DevOps agent uses a limited version of the Azure CLI. Also, the extension az afabric is not installed. Neither are the necessary environment variables set. Next, the Azure DevOps pipeline connects to your

Fabric account, so the commands az afabric can be executed. The Notebooks and Pipeline tasks are created by simply calling the corresponding task with required parameters. The Parameters are documented in the task document. The options to create an Azure Pipeline schedule with are only the mandatory parameters. The Azure Pipeline Schedule task is illustrated next.

7. Case Studies and Real-World Applications

In general, the end users of data engineering work are other data professionals, and they generally work from notebooks and pipelines that they are happy to share and that form a large part of the metadata that allow data engineers to build and alter the infrastructure. While users do their own development work, often on data stored elsewhere, fabric connections will be used to link the fabric experience with the underlying data, and fabric's capabilities and product integrations will be used to make the initial exploration of the data easier. Additionally, metadata transfer, security management, and other compliance tasks will focus on these data pipelines and notebooks when it is time for production.

In this section, we formalize our experience with Fabric through two anonymized case studies. In our description of the first case study, we describe the task in full detail in order to highlight all of the particulars of the Transformation component. In the second case study, we summarize the task because of its complexity. Product pipelines are often part of a larger complex of tasks, and our descriptions and limited dimensions of the case studies are designed to engage data professionals in a process of self-hypothesis that should lead them to explore Fabric Notebooks and Pipelines in the service of their Data Engineering goals. By offering concrete and interesting examples, we hope to increase the breadth and depth of the exploration.

While our focus is on its transformative aspects, Fabric also allows product teams to schedule multiple queries and scripts in order to offer a web service for hosting Multiple Languages Operations for the business teams responsible for the success of their products. These workflows track proximity sources and targets and include pre-processing and reporting capabilities.

7.1. Case Study 1: Large-Scale Data Transformation

In this section, we present two case studies related to real-world applications of Fabric Pipelines. Our studies show the role full-stack integration plays in making

pipelines easier, faster, reactive, and reliable, helping customers get more value from their data. Our first case study applies to data engineering domains such as anomaly detection, ETL, observability, and computer vision; the latter is a more advanced case study about data mesh, data discovery, and the role of notebooks in easing the deployment process.

A lot of modern enterprise applications need to process large amounts of data into model-ready format. In the past, data engineers connected their tools to the rest of the data app ecosystem through handwritten connectors and code, maintaining stateful, order-dependent scripts that are typically hard to observe, reason about, and fix. More recently, we have seen the emergence of declarative approaches to transforming data. This has happened hand in hand with cloud computing. Whether it is traditional batch job scheduling, micro service design, or observability, as a parallel processing platform, the declarative approach works better and is easier to use than the imperative queue-based one.

Adding cloud components and services take away most of the boilerplate work involved in scaling the ETL function and having the built-in ability to delegate that processing step to back-end services means pencil and paper coding is a lot rarer. But that does not mean data pipelines no longer require any coding. Far from it: creating a pipeline that backfills a data warehouse table and is triggered by webhooks is going to be coded. Reusable and composable functions, though, ease the process by buffering the complexity of individual steps and pushing it back until a complex one-off structure. But if we move towards a world where the act of writing a data pipeline is a one-off job, it frees up time for the data engineer's actual job of adding domain-specific code to these modular pipelines to kick off an end-to-end latency-sensitive function using these triggered pipelines.

7.2. Case Study 2: Scheduling Complex Dataflows

While Ado and Zarrin's pipeline is a typical computation to be fairly standard at many organizations, the scheduling of this pipeline is far from trivial and is not accomplished by a typical data-science or data-engineering notebook. The end result is that while the company does machine-learning on the data they have, they are reliant on very simple pipelines that have very bad data-correctness guarantees. Further, this means that the complexity of these pipelines is bounded, which impairs the depth and breadth of the machine-learning projects they take on. Luckily, the obvious solution is for us to build up more advanced ETL pipelines for them. This case study shows how easy it is to do this when all ETL pipelines, data transformations, and scheduling code exist in the same space.

An ETL pipeline that runs daily and involves datasets as well as company datasets. Datasets are produced by the hourly ingestion of partial-content logs. Those are particularly read-heavy load-balanced data, often with a combined volume of petabytes a day. Conversion logs help convert the data into proper accounts, but because the data is logged at the pixel level, we need to sample-and-split the data to connect the two. This and other questions have long been raised at the company but not until recently during a meeting about the longer-term setting up of a profit-sharing-based cost model for the advertising demand management of the company, which has the marketing manager hugely in favor, discussing the possible upside of doing these things, did we feel it appropriate to commit to a schedule for these projects that hooks into the advertising system.

8. Future Trends in Data Engineering

Data Engineering in 2023 has already witnessed a rapid increase in the availability of innovative and sometimes disruptive technologies that will likely define the future of data engineering for the next decade and beyond. Specifically, what we might call intelligent data engineering will increasingly popularize innovative techniques including large language models, high-level programming interfaces, agent-based data diversification, generative transformers, prompt engineering, and more. With these kinds of new intelligent data capabilities, the data engineering workflow will accelerate beyond the point of rapid development that we have already seen in terms of automation, democratization, and synergy with AI.

Emerging Technologies. For starters, large language models exhibit the ability to do code completion and code generation fantastically well. At the current pace of improvement, tools might soon develop into basic programming interfaces for multiple data and machine learning tasks. With large language models as basic programming interfaces, the next generation of data engineers will spend most of their time validating AI-generated code using code interpreters and unit tests. Such change will lead to the significant automation of bulk repetitive engineering tasks, including data preparation, data cleaning, data ETL, feature engineering, and so on. With large language models, data engineering will rapidly democratize. Increasingly domain-centric business users will be empowered to create and manage their own data engineering pipelines through simple languages like English.

The Role of AI in Data Engineering. Feedback loops embedded in machine learning models will enable the fusion of two worlds—data engineering and AI/ML. As the synergy between data engineering and AI/ML advances, the horizons of both fields will expand. From the data engineering perspective, AI will power intelligent data pipelines, orchestrate the flow of complex data transformations, design automatic feature engineering, and produce best data quality practices with little user input.

8.1. Emerging Technologies

At the time of writing, the field of data engineering is changing rapidly, with new technologies emerging all around, along with a more profound integration of AI and data science with software engineering. In this section, we survey a few such technologies that data engineers will be in contact with in the near future. As such, it will be interesting to combine their tech-savviness with their deep knowledge of the data domain and the data pipeline, to create novel implementations and solutions. Amongst the most significant new technologies that will shape data engineering, we mention the advent of serverless pipelines, decentralized data platforms, increasing adoption of vector-store embeddings, self-service BI platforms, low-code/no-code data pipeline development and management, real-time streaming data analytics, and AI-based data management solutions. A typical data pipeline involves a variety of CPU and memory-intensive transformations of the ingested data. Such data pipelines are common in the data warehouse and data fabric architectures. Moreover, the increasing need for real-time data delivery and the rise of ad-hoc data use cases have pushed for the development of pipelines that can react quickly to any new events, with a focus on low latency. Typically event-driven serverless architectures are commonly employed for such type of operations. It is increasingly common to see Data Pipelines that take advantage of serverless data processing providers.

8.2. The Role of AI in Data Engineering

Over the past several decades, artificial intelligence has been one of the most active fields in computer science and a dominating force behind the rapid advancement of technology in general. AI enables computers to simulate intelligent behavior by perceiving the world, making decisions, and learning through experience. AI primarily includes three high-level categories of methods: optimization, reasoning, and learning. Optimization is often used to define and solve essential problems efficiently, such as route planning, scheduling, network design, assignment, and resource allocation. Reasoning is often used to conduct logic reasoning over knowledge, such as explanation and question answering. Learning is often used to build machine learning models over data, primarily

supervised learning, unsupervised learning, and reinforcement learning. In particular, deep learning is a breakthrough in learning, which unleashes the power of learning to analyze large, unstructured data, such as natural language, image, and video.

In data engineering, AI has become an important part of building intelligent data systems, making data systems more efficient in terms of automation, usability, quality, insight, but also cost-effective. Firstly, in the world of automation, enterprises have been increasingly relying on AI techniques to replace human labor in building data systems, covering a wide spectrum of tasks, such as pipeline design, data processing, schema and data management, ETL, and interaction. Secondly, in the world of usability, the diversity and expansion of users require data systems to provide intelligent assistance. AI has been playing a significant role in enhancing the usability of data systems, in terms of conversational assistance, intelligent recommendation, self-service, and visual exploration, and lowering the barriers for end users to access the systems or data.

9. Conclusion

Our goal in this book was to introduce you to basic data engineering skills while demonstrating practical uses for the powerful tools in Fabric. After providing some theoretical basis and context, we took a hands-on approach — exploring several real-world data projects — to present the capabilities of Fabric and to illustrate how to combine these tools to facilitate your work. The pros and cons of moving your data projects to the cloud (and not adopting a multi-cloud strategy) is a separate discussion, but we believe that using Fabric Notebooks and Pipelines will streamline the execution of your data engineering and science needs. Using Fabric, you can use Microsoft's vast cloud resources for storage, basic data transformations, and data orchestration easily through intuitive and friendly interfaces. You can also extend and complement these resources through your code and knowledge.

Hopefully, you learned some of the possibilities of using Fabric Notebooks, especially the new notebook engine that integrates with Jupyter while extracting the power of Fabric. You also got a taste of Pipelines and Orchestrators without which notebooks could be relegated to prototypes waiting a long time before they are moved into production or abandoned. You saw that Fabric provides some tools to use by anyone in the company and others that need advanced skills and are meant for data engineering professionals. As the integration with traditional

data formats is added to the tools, the use of Fabric will increase, and the hope is that it will keep evolving to be your data projects hub, whether done by a data engineer, a data scientist, or by other non-technical employees.

References:

[1] Ghosh, Debananda. "Real-Time Analytics with Microsoft Fabric." Mastering Microsoft Fabric: SAASification of Analytics. Berkeley, CA: Apress, 2024. 209-241.

Deep Science Publishing, 2025
https://doi.org/

DeepScience
Open Access Books

Chapter 4: Real-Time Analytics with KQL and Event Streams

1. Introduction to KQL in Fabric

Azure Data explorer allows you to perform fast and highly scalable data exploration using KQL and get instant business value by creating stunning dashboards and reports. KQL is the query language used for the majority of Microsoft Azure's analytics services. KQL is powerful, expressional, read-only request syntax that is used to query and ingest data. It is similar to SQL in some aspects, especially when it comes to extracts, transforms, projects, joins, and unions.

KQL is purpose-built to facilitate working with structured and semi-structured data. It extends the practical aspects of using SQL with additional powerful commands, and improves performance for some scenarios through its other capabilities. KQL contains commands for ingestion of both structured and semi-structured data into the Azure Data explorer service. The ingestion commands support multiple sources of data, including Azure Blob storage, Queue storage, Azure Event Hubs, Azure Service Bus, and streaming data from the web.

KQL is used to query streaming and stored data in Azure Data explorer, and is also at the core of many other Microsoft products which provide data analytics features such as Azure Log Analytics, Microsoft Sentinel, and Azure Monitor. Azure Data explorer provides very low latency ingest and query times and can handle massive amounts of data. The Azure Data Explorer managed service is capable of ingesting around 3 TB/minute of unstructured log data. With appropriate best practices, query execution and ingest times can be extremely low for both streaming queries and stored data.

1.1. Overview of KQL

Kusto Query Language (KQL) is a language designed to query, analyze, and transform large amounts of data in real time. KQL is not T-SQL on massive datasets, and KQL is not programming - it does not have types or type declarations, nor does it have loops or complex programming constructs. KQL is highly optimized for the types of queries you commonly issue against streaming data to offer first-class performance while being both user-friendly and very expressive.

A KQL query consists of a list of statements, and each statement can be composed of a comment, a function declaration, or one or more query expressions that are evaluated sequentially from the top down. Each expression can be a single statement expression or a chain of expressions. You can define KQL functions, including user-defined functions and data manipulation functions, for DRY code, reusable code, and sharing between multiple queries. You can also use KQL inline functions to allow code reuse within the scope of a single query.

A key part of this is its operators, functions, expression syntax, type system, and rules. KQL has operators similar to SQL's SELECT-FROM-WHERE and JOIN, data prep functions, and let statements for common table expressions. KQL has most popular SQL constructs. We implemented them where they made sense, but there are significant differences and KQL has a different focus. SQL is designed for database and query performance. KQL is designed for extensive data pipelines and monitoring use cases. KQL also allows the writer to disregard the need to accomplish multi-pass execution and join order optimization to extract maximum execution speed.

1.2. KQL Syntax and Structure

KQL is structured via a modular syntax that makes it easy to read, write, and maintain, even for users who are unfamiliar with KQL. In addition to the keywords, operators, and expressions mentioned previously, KQL scripts are built using a limited number of commands. The core commands are as follows:

Table Variable Declaration – The make-series command creates a new table (called a variable) whose values are the series results. It serves as the assignment statement for a new query variable.

Data Retrieval – All standard commands, such as union, are used within KQL to provide data in tables. Commas separate the statements in sequences. However, you must take care to keep the logical flow sensible.

Result Control – The project and project-away commands transform tables before they are returned in results. The order of results is controlled by the order-by arguments in project.

Builtin Functions – KQL has many builtin functions that can be used in any query.

Any part of a KQL script can be encapsulated in curly braces and used in other parts. For instance, if you very frequently want to summarize data by hour slot, and you want to see that in a color table to quickly see if anything important happened in a time slot, you would want to code that as a function and call it instead of coding it over and over.

KQL also provides shortcuts for commonly used queries. These aliases and shortcuts are not visible in the query results but can be called to see the actual query that will be executed behind the scenes and its structure. However, just like functions, user functions can be defined and used as well.

1. Create Stream

```
CREATE STREAM stream_name (
  column1_name TYPE,
  column2_name TYPE,
  ...
) WITH (
  KAFKA_TOPIC='topic_name',
  VALUE_FORMAT='JSON' | 'AVRO' | 'DELIMITED'
);
```

2. Create Table

```
CREATE TABLE table_name (
  column1_name TYPE PRIMARY KEY,
  column2_name TYPE,
  ...
) WITH (
```

```
  KAFKA_TOPIC='topic_name',
  VALUE_FORMAT='JSON',
  KEY_FORMAT='KAFKA'
);
```

3. Select from Stream/Table

```
SELECT column1, column2
FROM stream_or_table_name
WHERE condition
EMIT CHANGES;
```

4. Create Stream from Existing Stream (Filter/Transform)

```
CREATE STREAM new_stream AS
SELECT column1, UCASE(column2) AS upper_col
FROM original_stream
WHERE column3 > 100
EMIT CHANGES;
```

5. Join Streams

```
CREATE STREAM joined_stream AS
SELECT a.col1, b.col2
FROM stream_a a
JOIN stream_b b
  WITHIN 5 MINUTES
  ON a.key = b.key
EMIT CHANGES;
```

6. Aggregate Stream into Table (Windowed)

```
CREATE TABLE agg_table AS
SELECT key_column, COUNT(*) AS count
```

```
FROM source_stream
WINDOW TUMBLING (SIZE 1 MINUTE)
GROUP BY key_column
EMIT CHANGES;
```

Supported Data Types:

- STRING
- INT
- BIGINT
- DOUBLE
- BOOLEAN
- ARRAY

1.3. Key Functions and Operators

KQL is a powerful query language that makes analyzing machine data easy, fast, and fun. It simplifies our working life, enabling us to write analytics queries quickly, even if we are less experienced with coding. This is particularly important for domain experts such as security analysts or IT support who need fast responses and as few obstacles as possible. KQL is not just a simple filter or a search. It is a command language that helps you break complicated problems into a sequence of easier ones, each yielding meaningful intermediate outputs, leading you on the way toward solving the overall problem.

KQL Explorer, one of many tools powered by KQL, is an easy, powerful, and intuitive way to query and analyze your data. Most importantly, it allows you to extract just the relevant portion of the data as an output of your search operation. The power of KQL unfolds especially when working with large amounts of structured and semi-structured logs or events that originate in various ways. KQL shines when using it to perform ad hoc week or month-on-month diagnostics in an exploratory fashion, or to create dashboards and alerts for established long-term detections. What sets KQL apart from high-performance databases are the built-in security and management capabilities available within certain platforms.

Among the frequently used functions are Makespan, which runs predictive usage optimization based on data from the same workspace, varying resource utilization over time, as well as similar resources within the same region and capabilities, Totals with time and target subsampling, which makes it easy to spot irregularities in selection-based costs, Scamper and Sagetable for resource tagging and custom routing, etc.

1.4. Use Cases of KQL in Fabric

KQL-powered tools are infinitely extensible and can address a wide variety of use cases, sometimes in unexpected ways. On the data operations side, KQL acts as a glue language to unify and accelerate a range of tasks, from deep analysis to low-code orchestration, that increase the value of data assets in Fabric. In this chapter, we'll highlight those data operations capabilities by walking through tasks in Fabric you can accomplish with KQL and then moving on to the data consumption side. In the next chapter, we will give an overview of KQL syntax, and in the two following chapters, we will go deeper into KQL expressions and functions and how to structure your statements inside the tools you'll use. Data pipelines and orchestrations: Use KQL in Azure Data Factory Mapping Data Flows to design scalable data flows that ingest, process, and output data, augmenting the orchestration capabilities of KQL functions with Data Flow and orchestration capabilities in Data Factory. Data flows and orchestrations can connect to any Fabric data other data integration tools can. Low-code Data Factories and easy pipelines: Sort, filter, limit, join, union, append, and elevate lakes, add computed columns, or even do light exports from KQL inside Power Query. The low-code Data Factory and easy pipeline features in Power Query reduce the coding burden on analysts by providing no-code point-and-click alternatives. Then, in teams where coders are present, coders can deploy KQL-based pipelines for the entire organization that Power Query Data Factories can leverage.

2. Event-Driven Architecture

Modern systems are increasingly adopting event-driven architecture. Today's businesses expect their software systems to be responsive, in the reactive pattern, to changing business needs and to handle the increasingly large volumes, velocity, and variability of data being generated for e-commerce, social media, mobile payments, and similar use cases. The early service-oriented architecture, which decomposed monolithic systems into discrete services, while laying a foundation for better scalability and maintainability, was adopted for

performance, resource utilization, and ease of scaling. However, service-oriented architecture often makes it difficult to quickly respond to changes in requirements. More recently, we have seen the rise of Microservice Architecture, where the services have smaller latency and are more responsible for their own states.

Event-driven architecture takes these ideas and enhances them. In event-driven architecture, the individual services are treated as being largely decoupled. Instead of making synchronous requests from one service to another, services emit events and react to events generated by other services. Services generally do not communicate directly. Services could also have different lifecycles and be written in different languages or frameworks.

Event-driven architecture greatly reduces the amount of orchestration needed to get things done. For example, let us consider a user making a purchase on an e-commerce site. After entering payment details and confirming a purchase, the user receives an acknowledgement. At the same time, inventory information is updated, and shipment arrangements are made. In a service-oriented architecture and strict synchronous communications with requests/responses, the services would need to maintain session state.

2.1. Fundamentals of Event-Driven Architecture

Event-driven architecture (EDA) is an approach for enabling software systems to respond quickly, seamlessly, and efficiently to events that may occur while the systems are operating. Event-driven systems are designed so that event detection is separated from the actions performed in response to events, which have internal or external origins. Events are defined as occurrences of expected or unexpected conditions at specific points in time. The detection of unexpected events usually indicates the need for taking a corrective action. For the system to be efficient, such an action should be taken as soon as the events are detected or at least within a specified, small time window. An internal event originates from one of the components of the system itself or the composite system. An external event originates outside the composite system, such as the arrival of a message from an external system. The separate detection of events and the execution of conditional actions implies the use of the observer design pattern.

An event-driven architecture defines an application design model that supports the production, detection, consumption of, and reaction to events, and the dependent changes in data and state. A data-centric event is done best using data that describes the occurrence of the event. An action-centric event is based on the actions that happen to be enabled in response to the event that has occurred. The

latter approach relies on function calls in remote procedure and service calls to be able to address the needs of special domains where specialized functions need to be performed in response to occurrences of events. An event-based approach to event-driven architecture involves the use of events for application design, enabling flexible, high-level, and abstract navigation of data reports.

2.2. Components of Event-Driven Systems

All event-driven systems are composed of distinct yet interconnected elements, which are brokers, event publishers, event consumers, event streams, and events. An EDA uses an event stream broker to route events from event producers to event consumers. These communication points are not generally exposed to each other beyond sending and receiving messages. The event publisher typically enters information about some change in the state of the system. This change of state is captured as an event and sent to the event broker, which can route these events to multiple or single subscribers.

The event broker then determines which event consumers are interested in consuming events from which streams and then sends those events to the consumers. The event consumers subscribe to certain types of events from specific event streams and process incoming events based on defined business logic. After capturing the change of state in the event, there are several possible use cases for event consumers. This change of state is processed for analytics, reporting, auditing, compliance, enrichment, feeding databases for queries, triggering some action, and triggering workflows. This processing of data-in-motion is the foundation of an event-driven architecture.

2.3. Benefits of Event-Driven Architecture

Despite a challenging implementation, in practice, event-driven architectures provide many benefits over traditional ones, including more flexible cooperation among different business domains within an enterprise, improved performance, lower latency, improved resiliency, and easier support for modern development paradigms. Because an event-driven system consists of many separate services that communicate with each other using events, it can be designed so that different services are implemented by different teams, using different technologies. This means that differently, but related and cooperating business processes can be done with interestingly different approaches. Different services can be independent or be implemented using different application paradigms, such as microservices, serverless functions, or containers. When a business process can be split into multiple services owned by multiple teams, developed independently, and released for production at different schedules, events provide

an important management guideline. Each event represents a state change, with business meaning attached to it, which all teams can understand. The event level becomes a major aspect of the business logic and serves as a boundary in the application architecture. The event level acts as a contract between the teams for each releasable version of that event. If one business process changes and does not affect the events generated or received by another, then only one team must make changes, be tested, and be put into production. By moving event creation closer to the service that changes the state of a business entity, latency can be reduced. More generally, in a classical architectural style, one request must wait until a response is returned by the service responsible for handling that request. In contrast, in an event-driven style, the event-processing application can receive events asynchronously, without requiring a request/response protocol. This event-driven approach often provides lower latency and closer to real-time application behavior.

2.4. Challenges in Implementing Event-Driven Systems

Event-driven systems bring many advantages to organizations, but also have associated challenges, such as their distributed nature. Integrated event-driven systems grow to be complex, asynchronous, distributed systems. Building, deploying, and/or managing either the overall system or individual event-driven services is more complicated than with simpler organization patterns. Debugging issues as they arise or understanding service behavior during execution are generally more difficult. More services and a more rapid deployment cadence can lead to more failures in production and may increase the latency involved in recovering from them.

Infrastructure dependence is a first-class concern in event-driven systems. Developers can write code that executes business logic but they have to rely on messaging or event infrastructure to handle the service invocation, failure, and retry behavior that is generally done for them when they implement simpler synchronous transactions. Failure conditions primarily relate to missing messages, duplicate messages, and versioning issues. Operations teams have to make sure that the service is executed and the latency characteristics fall within acceptable bounds. Because of the dependence on the underlying infrastructure, developers and operations have to coordinate more closely and many teams share both responsibilities.

Right People and Assets: Most organizations have not spent their resources in the quantity or diversity required to build a rich set of reusable components for event-driven systems. The challenge is not in producing those assets, but that many

teams underestimate the effort required and/or overestimate what they can accomplish using them.

3. Monitoring and Streaming Analytics

Monitoring the health status and error behaviors of large-scale information systems are crucial for service reliability and system security. With the rapid deployment of large-scale and distributed systems, traditional static models cannot easily keep track of the characteristics of the system. This is largely because of the complex interactions among services, semi-randomized nature of the traffic and dynamic patterns in resource consumption. Therefore, it is necessary to build up an automated online monitoring infrastructure to support various stakeholders during the deployments and operations of large-scale systems. In the context of large-scale web applications, we discuss the monitoring problems from two different perspectives: service-level monitoring and system-level monitoring. From the service side, we need predictive models to track the service completion time in order to maintain service level objectives. From the system side, we travel the inverse direction and construct service request models based on monitoring the system resource consumption. To support these two different problems, we develop techniques to capture both the short burst of disturbances and long-term variations of resource utilization from the internal perspective of the system itself, and service completion time distribution from the external perspective of the service itself. We also build multiple demo panels and illustrate the monitoring analytics in live server clusters, distributed message queues, and large-scale services deployed in cloud.

3.1. Importance of Monitoring in Real-Time Systems

A very popular definition of a real-time system expresses that a real-time system is a system in which the correctness of the behavior depends not only on the logical result of the computation, but also on the time at which the results are produced. The distinctive characteristic of real-time systems is therefore the time constraint associated with the computation of the results. Real-time systems can be divided into soft or hard real-time systems depending on how important it is to reply within the time constraints. At the one end of the spectrum are hard real-time systems in which missing a deadline produces catastrophic results. In the middle of the spectrum are soft real-time systems where the system goals are harder or softer depending on how far from the timing constraints computations are produced. In this latter case we can also have a relative monotonicity constraining how the timely responses are produced over time. In this case, a

system implementation is more desirable if it is more monotonic. The opposite end of the spectrum from hard real-time systems are just systems with timing constraints where the deadlines can be missed, but doing so has a significant impact on the quality of the system.

Real-time systems manage resources that interact and compute activities which are time-critical. A possible definition of a time critical activity is that they are activities where some events that act as triggers of the computation need to be processed quickly and within defined upper bounds. Computing events quickly is the main goal of a real-time monitoring system. For example, consider an enterprise that sells products according to some schedule and has invested significant number of resources to obtain alarms about events that require immediate responses, such as a possible error in the demand forecast that defines the sales of a product so that a timely correction can be made.

3.2. Techniques for Streaming Analytics

Streaming query processing is meant to provide a near real-time response. In doing so, it sacrifices certain data model and functionality by imposing a more restricted model for processing queries over streaming data than the batch processing model. Realizing and predicting what such restrictions should be requires a deeper understanding of the behavior of data over the specific monitoring domain under question. The latter consideration ties into the broader goals of the monitoring system in the first place. A monitoring framework lays out such a taxonomy of goals of general monitoring systems in order to help analyze the domain at hand, and port the results onto guidelines and design specifications for a monitoring system focused on streaming analytics.

Query results in traditional data processing tasks are generally only overwritten when a query is re-evaluated with new input data brought in by newly completed data ingestion cycles. Optimization techniques such as caching previous results for interval queries, that benefit from similar data access patterns, provenance queries for lineage results can locate their affected input data-based sources for value change, have made efficient incremental query result updating an academic specialty area for batch processing-based frameworks. By contrast, advanced query processing techniques in streaming data engines dig into the features of the specific query data type dependency on time to maximize efficiency. The underlying idea is the time correlation in query data that results in phases with similar values of the query within specific time intervals. Stream engines expose these granular data “phases” to queries, providing APIs for both monitoring over the granularized stream and query task scheduling oriented at them.

3.3. Tools and Technologies for Monitoring

Defining targeted monitoring for different enterprise assets may be a complex task, and there are many available tools and technologies dedicated to monitoring. With the class of events that can be modeled in predefined templates, such as security-related events, network events of different layers, and many other layers of the technology stack, classic tools of monitoring are still quite popular in the field. However, with more modernization of enterprise technology stacks and adopting agile/scrum methodologies and DevOps practices, including increased use of microservices and cloud computing, these services that require high uptime are often not directly reachable with classic monitoring probes anymore. More services are pushed into the cloud, and the composition of complex applications with continuous rolling updates of containers and services in private or public clouds is making them even harder to monitor.

Infrastructure and platform hosting layers need for monitoring are traditionally provided by cloud vendors, collecting some of the events from built-in APIs in the provided infrastructure and platform services. In addition to abstractions of layers below the application, many log aggregators provide agent services installed inside containers or virtual machines that gather logs and events and help detect monitoring anomalies. AI/ML-based alerting and notification services can also be pushed on top of these aggregators to detect abnormal behaviour in the logs.

3.4. Case Studies of Streaming Analytics

Large-scale monitoring systems must match the variety and amount of events originating from the monitored distributed systems. Moreover, they need to offer adequate latency bounds — alarms should pop up before the problem is solved. Tools have been developed to address these twin challenges: achieving horizontal scalability and offering adequate latency bounds. Thanks to its framework based on a variation of Data Stream Processing, impressive throughput is achieved and it is able to guarantee, in bounded memory, guaranteed output. The system is able to accomplish both goals by exploiting parallelism extensively. It uses multiple workers to evaluate partitions of a lightweight version of the system composed of micro-aggregators; the system organizes these multipath flows into a workflow encompassing both accumulator/estimated-next-points and window processors dealing with segmented-microsegment flows.

4. Integration of KQL with Event Streams

Enterprises collect tons of event data every day to gain insights into operations and to detect, investigate, and respond to security threats. Traditionally, this data is stored in fast, resilient, and inexpensive data stores. Data in these stores is leveraged to generate insights and detect anomalies using Kusto Query Language (KQL) statements. However, building a data lifecycle management metaphor that extracts and feeds the most important records back to the queryable store incurs latency in production. It requires data processing that is not real-time.

Integration of KQL with streaming technologies and specialized transport protocols—it's a great combination that enables event data to be analyzed in real time using your favorite query and analytics language. Think of sending events from Kafka or Event Hubs into a KQL query and having it run as a search and reporting service. It eliminates the need to run a projection to a data warehouse or data mart afterward. Further, it allows to apply a KQL query on live data every single time you want to generate insights. You don't wait for batch jobs to run. You can write a KQL query just as you do today for data that is past, and have it run as an on-demand service. You get the power of KQL with the low latency of streaming technologies.

Azure Data Explorer's integration with event streams supports processing the data in real time to gain immediate insights into operations and business functions. Event data sent into the Azure Data Explorer Traffic Management component over the Kafka or Event Hubs transports will run your KQL queries as real-time analytics; returning results immediately.

4.1. Connecting KQL to Event Streams

KQL can be a powerful tool for filtering, manipulating, and analyzing event data that flows through a streaming system like Apache Kafka. However, KQL typically only operates on data that is ingested and stored in a Kusto database. In this section, we show how you can build a custom KQL data connection to allow KQL to access event data in real time, while it is still flowing through the stream.

KQL uses the concept of a "data connection" to allow it to connect to any data source, such as a Kusto database, or an external data source to query data, augment Kusto data with external sources, and then save it into a Kusto database. In this section, we will show you how to build a KQL data connection for Apache Kafka. This KQL data connection extends KQL capabilities to query a Kafka

topic directly as a direct-partitioned storage, write data to Kafka, and append Kafka data to Kusto tables. This allows deep analysis of event data using the rich set of KQL functions. For example, you might want to use the KQL "sentiment" function to determine the sentiment behind Twitter posts, and then update a Kusto table periodically with this sentiment to create sentiment scores over time.

The KQL Kafka data connection is based on a Kafka connector for Kusto. The connection is implemented in a KQL plugin and allows KQL to execute queries against the Kafka streaming system using the same Kusto connection interfaces as connecting to Kusto data. Data is written back to Kusto using the same interfaces as ingesting external data into Kusto without the custom load API needed for doing it via an external connection.

4.2. Real-Time Data Processing with KQL

Real-time analytics are queries that run on data that is still ingesting from one or across multiple data sources. The classic example is a query of web server logs that show how many users are currently active on a website. Typically this data is queried statefully, showing the current value of the result. Because of the high volume of data that is still ingesting during the query, the resources needed are high. Special features allow for mitigation, such as limiting the time window of data being queried.

Currently, KQL is used in two main ways in event streams that are still ingesting: counter-based queries that allow the user to determine current state in real-time streams; and result-based queries that produce the result states of the query and generate events corresponding to that state for the event stream at a lower frequency. Within KQL, this is made possible because KQL is a query specification language with which end-users can easily aggregate data on diverse types of data with ease, merging and joining diverse types of data sources within the query.

The current demand for getting real-time answers to diverse questions about ingesting and constantly changing data comes from all kinds of users in all types of organizations. Data scientists use it for monitoring business-related metrics, tracking changes as KPI values rapidly evolve. Business people use it for receiving immediate alerts, because some business-relevant metric has changed, for example, sales have dipped below a certain threshold or some social media data trend has taken place. Operations users send it for monitoring events of their day-to-day operations, such as changes in a service's performance. Security users invest money in high-throughput data monitoring systems because businesses are deeply concerned about service availability and reputational risk.

4.3. Analyzing Event Data using KQL

KQL offers event viewers with immediate access to event streams and real-time search and analysis capabilities—without requiring any prior configuration or setup, other than connecting KQL to a signal broker. Event participants can quickly change the filters applied to both what is being viewed and how it is displayed, in order to meet operational needs. In addition, all filtering effects are easily recovered. Real-time troubleshooting of a fault or anomaly condition often requires good knowledge of the signaled system and its operation, and may involve multiple participants with overlapping but not identical expertise. After identifying an operation or condition of interest, the next step is to understand the details of its action and relation to other events or signals—in particular, those that were observed during other conditions. KQL supports these tasks by allowing filtering argument values to be added or modified while viewing active event data as well as recorded data.

KQL provides a simple interface for defining multiple filtering types. By using a mouse right-click on a signal rendered on the display, the first KQL Filtering Properties window will open. The selection allows display and modification of the currently selected filtering category, and the exploration of the data to input new filters or modify existing filters. The signal categories assigned to all rendered signals and the events available for filtering are dynamically explored according to the rendering definition. Multiple filters can be created based on the event type, value, source, or purpose of its signal. The second window allows modifying the currently defined filtering sets.

5. Future Trends in Real-Time Analytics

The future of real-time analytics will be shaped by a number of emerging technologies and trends: intelligent and contextualized event-based analytics; automation of the ETL process and query composition; embedded analytics and democratization of analytics development; event-redaction services; low-latency streaming databases; event stream buses and distributed ingestion; real-time predictive analytics; and vendor adoption of the event-centric business model. Each technology is a building block for the next generation of event-driven data analytics and analytics services. This chapter describes these technologies and trends and predicts the implications for users and vendors.

Analytics cannot answer all data business questions all the time, but it has to be able to turn an increasing amount of events into insights. In this chapter, we

present our predictions about the near-future of real-time analytics and analytics services. We outline the major analytics technology building blocks that are or will be objects of significant innovation and only briefly describe the architecture that focuses on event streams as a new type of data source. We make the case that the future belongs to event-centric analytics and the associated data services. This is a very narrow, event-centric perspective on analytics. Technologies and techniques that are mentioned, such as adversarial predictive analytics, interactive machine learning, search and analytics, augmented analytics, and analytics synthesis, are generally applicable to many types and classes of data and analytics systems, but our belief is they will become first-class components of event-centric, event-driven systems and services in a not-so-distant future.

5.1. Emerging Technologies and Tools

Innovations in real-time analytics tools and techniques are proceeding at an accelerated pace [1]. An interesting new open-source project to mention is a distributed publish-subscriber message broker famous for its high throughput and low latency. An alternate open-source project offers similar features. Both projects bring enterprise capabilities to the event-driven application. These lower-latency, higher-throughput systems allow event feeding to a larger number of event-driven applications, including those that desire only simple transformations. Consider a news-oriented event feed generated by a number of feeds and filtered by one or more keywords, along with a service triggering notification whenever one of the keywords feeds something interesting.

Once the event stream system is in place, it is possible to build stand-alone event response services that can trigger directly from the event stream in the same way that we can continuously monitor a database for relevant commands. These event response services will become more complex and have a number of similarities to database services that have been implemented. These services will allow NOTIFY on keywords, keyword pairs, or simple keyword predicates. These services will also allow subscribe on keyword pairs or simple keyword predicates. Cloud computing is becoming ubiquitous and affordable. Adding a low-cost virtual machine dynamically and allowing it to run for a few hours is not going to test the limit for cloud services, and the access and programming allow adding and testing new algorithms. An exciting example of this innovation is a program that has been initiated, allowing anybody to upload new models and algorithms for testing against a sovereign data set.

5.2. Predictions for Event-Driven Analytics

The main event-streaming technology was first released in 2011. The number of its deployments has been growing rapidly. It is not just another random technology: it quickly became the de facto industry standard for stream processing. There is nothing like it when it comes to the number of mainstream adoptions, vendor and community support, first-hire migration pathways, user experiences and shared knowhow. And many new tools, frameworks and libraries commonly used in data analytics have been developed with support in mind, so their features can often be utilized in event-driven analytics as well.

As the number of analytics tools directly dependent on streaming technologies' long-term success is large, these platforms should have no problem adapting, augmenting and delivering support for widely demanded features or growing use cases. The architecture of such streaming platforms inherently allows solutions to rapidly accommodate additions and improvements, for example, retroactively serving previously published data in a data-lake-like mode or supporting more complex functions on or over event streams in addition to the usual flow functions. More things are generally better. The popularization of diverse, specific analytic solutions will create demand pipelines for these additional-function requirements, as the basic functions become so ingrained in organizations' analytic processes that their data analysts cannot help but be on the lookout for additional productivity improvements or pain-stake removals.

6. Conclusion

Designing and building a reliable, scalable data processing foundation is no easy task. It’s impossible to anticipate every eventuality. But many scenarios may justify using some form of persistent message queue or event log. Adding useful functionality to saved events strongly reduces the maintenance burden of constantly reprocessing data to answer oneself or others’ questions about “what happened when?” By combining easy-to-use semi complicated query languages and data streaming technologies, it’s possible to create a data access and processing framework that will serve the new data-driven reality for years to come.

This book started off as an account of newer technologies. Distributed systems and related concepts have always been a forte. This book, however, isn’t just about specific technologies. Rather, it has a straightforward, logical goal that all data-oriented books should have: to explain complex underlying concepts in an

accessible way, regardless of who the vendor is. With clear presentation and focus on what's truly useful, any person who's responsible for maintaining a technological equilibrium for themselves or their business can be comfortably reassured. Listings are normalized, anecdotal instances are irreducibly condensed, and simple strictures for dealing with all sorts of events help bolster the certainty that things will stay normal. Do you think writing books is worthless, because all people do is read them, and then go on to screw up the same way they had before? Writing is an act of defiance against the futility of human existence. Be it a hundred times, or a hundred thousand, a new book about the challenges of knowledge persistence is a sure way for people to absolve themselves from misunderstanding how they did it the last time.

References:

[1] Tommasini, Riccardo, et al. "Declarative Languages for Big Streaming Data." *EDBT*. 2020.

Deep Science Publishing, 2025
https://doi.org/

DeepScience
Open Access Books

Chapter 5: AI and Machine Learning in Microsoft Fabric

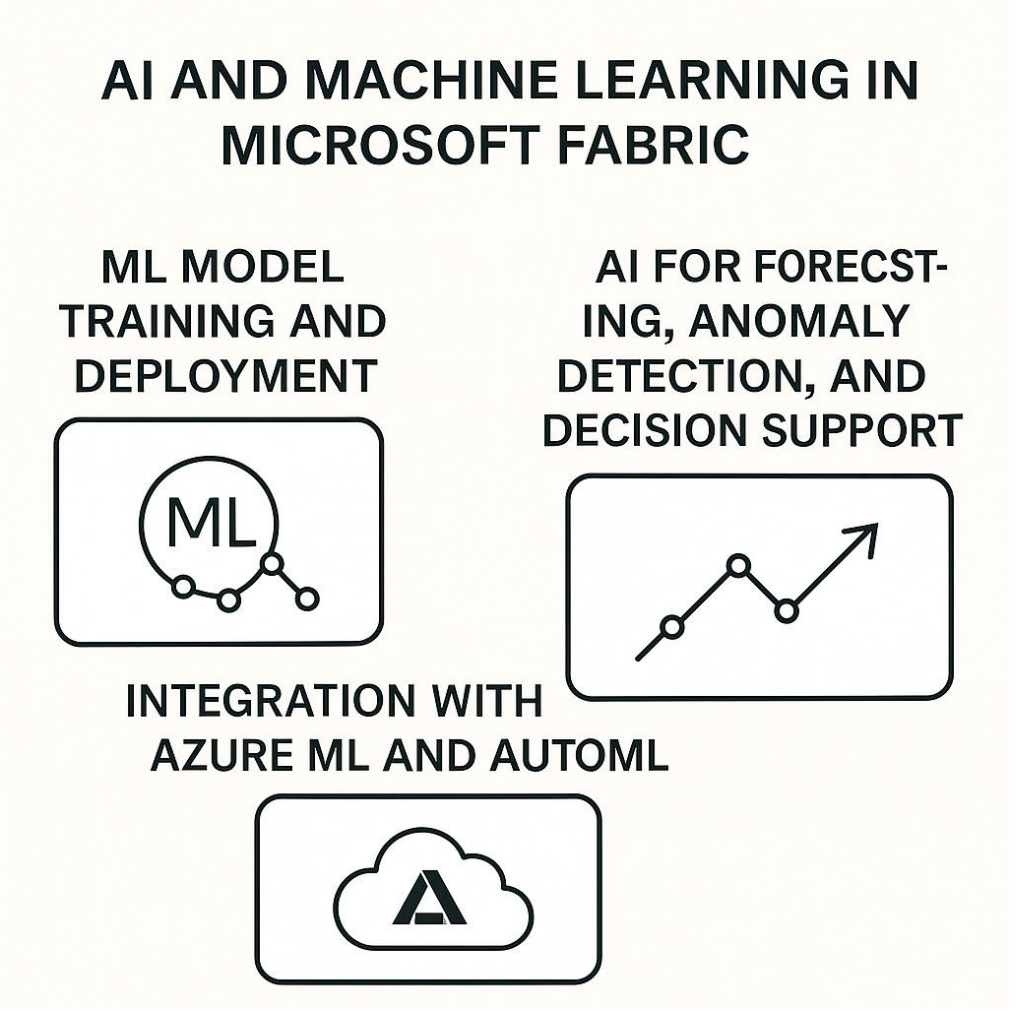
1. Introduction to Microsoft Fabric and AI

Microsoft Fabric is an integrated analytics platform that helps us get more from our data and AI investments. As a fully managed cloud service, it enables anyone to do analytics by automating the complex tasks of working with data. Microsoft Fabric includes a range of analytics tools that integrate seamlessly, reduce time-to-insight, and can be adapted to every skill level.

Why another platform? Because we have to cater to people with different skillsets and different needs. Microsoft Fabric integrates into a single workspace what people want to do: query, model, visualize, integrate, and orchestrate. It hides much of the complex environment setup so that people can jump straight in at the level they want. And it provides an architecture that can be used for everything from a personal report or pipeline, through workloads with thousands of integrated connections, to data lakes and warehouses that organize and store huge volumes of enterprise data. It also has a level of orchestration that adds reliability and ensures context, so that end users don't wake up to broken reports, or worse, broken automation jobs that are only encountered days later.

Microsoft Fabric brings together all four components of a successful data strategy: Data engineering, Data warehousing, Business intelligence, Real-time analytics. Microsoft Fabric does this in a way that allows users of varying skillsets to work on the same pipelines, report layouts, and models without having to resort to exports and imports. Want a business analyst to set up a pipeline to copy a few tables into a data lake? Don't write tedious templates; just let them create a pipeline job from Dataflows. They can build a connection to the data lake, then design the copy job and specify a schedule for it just like any other

dataflow. Want to simplify a feed, or complexify a report model? No longer do you have to export to OneDrive and reopen using the other tool; instead you can open the pipeline right from the report, and navigate between the source tables in the dataflow and the data model in the report.



2. Overview of Machine Learning Model Training

Machine learning involves creating algorithms that learn from a set of features and a label set. The goal of the ML model is to optimize the set of weights by maximizing the accuracy. To achieve this goal, we split the dataset into train set, dev set, and test set. The ML model is trained using the train set. After training the model, we use the dev set to tune hyperparameters that cannot be learned

from the training set but can help improve the model prediction accuracy. Finally, we use the test set to evaluate the accuracy of the tuned model before deploying it for prediction.

Machine learning problems can generally be broken down into three phases: data processing and feature engineering, model architecture design and model training, and model deployment and monitoring. The first phase is crucial to the performance of the model. In fact, a model may achieve higher accuracy if it has been engineered with the right set of features. The second phase focuses on creating a model architecture that best describes the underlying relation between the features and targets. Research has shown that different model families have different biases that affect the final accuracy of the model. The trade-off is that a complicated model architecture that is highly expressive and learning capacity could easily lead to overfitting if not monitored carefully.

Several services provide the ability to build and deploy ML models easily. In this chapter, we will go through some of the core ML functionalities. This includes: 1) creating and processing data pipelines, 2) designing and training ML models, 3) deploying model inference and automated retraining, and 4) monitoring and evaluating model performance.

3. Data Preparation for Machine Learning

Model training is the crucial step where the model learns the patterns present in the data. Almost all computer vision, image processing, natural language processing, and reinforcement learning techniques utilize deep learning to train models that have achieved state-of-the-art results in numerous tasks. Machine learning has also achieved unprecedented success in applications across various domains. Creating a model that works perfectly and is robust enough to work without any issues takes time, effort, and various techniques. But how do we achieve that success?

Before performing model training, certain steps must be executed on the dataset to make it suitable for training. The processes of data cleaning, processing, and reduction, together referred to as data preparation, are now widely recognized as important to model accuracy, but they are often overlooked. In this section, I will explore the various strategies to accomplish data preparation.

Data Cleaning Techniques-Data cleaning is a complex step. It is concerned with detecting and correcting errors in the dataset. As with many other computer

science disciplines, automated systems and techniques are being developed that aim to accomplish this complex task. No single mechanism is adequate to accomplish data cleaning. Users are encouraged to explore the many tools available and refine them with hand-coded techniques. Even the best systems available are not guaranteed to clean the dataset with 100% accuracy. For many AI and machine learning systems, a large percentage of the dataset is not checked manually, and small errors that persist can have a significant impact on the performance of the model.

3.1. Data Cleaning Techniques

Data cleaning is an important first step in any machine learning project. Contamination of the dataset can lead to all sorts of problems, such as biased estimates, incorrect models, and the sufferer of more than a few bad days. Therefore, it is essential to check the quality of the data. Some commonly faced issues in your dataset can be too many zero values, too many missing values, or too many outliers. Also, checking for duplicate records is necessary. Cleaning out the errors manually or specifying a particular filter can help. Cleaning and preparing a dataset are usually as important as the model and features used. There are two common types of problems or errors usually present in datasets, namely categorical data problems and numerical data problems.

Usually, datasets with categorical values such as country, gender, etc. may contain errors. Categorical value datasets can be a huge source of problems if not dealt with properly. Checking for duplicates in this type of dataset is a must. If there are more than one records with identical values for unique 2 or more fields, those records should be removed. Further, you should check for typos or syntax issues within the data. Spelling mistakes, capitalization problems, spaces, and incorrect data types are possible issues. Combining similar categories is possible but requires a clear intention, knowledge of the data, and domain.

Another important task is data imputation. Data imputation refers to the process of replacing missing values with the original values. If there are little amounts of missing values in a dataset, normal filling strategies such as forward-filling, backward-filling, and filling with mode can be applied. However, if the dataset has huge amounts of missing values, you should be concerned and use more sophisticated techniques to overcome the issue. These sophisticated techniques can be K-Nearest Neighbours for imputation or Multivariate Imputation by Chained Equations.

3.2. Feature Engineering Strategies

Feature engineering is among the most important stages of a data science project. In this stage, we transform the variables available in the dataset into new variables to improve the performance of the machine learning algorithms. The most common applications of feature engineering consist of grouping, transforming, scaling, or creating new explanatory variables from the existing features in the dataset. In the following sections, we go through the most common feature engineering strategies.

Data Distribution Correction

Data distribution correction is important before running different data science algorithms because some of them assume some data distributions to estimate statistical significance. The most important algorithms that assume data distributions are the Gaussian Naïve Bayes algorithm, which assumes that the features follow a Gaussian distribution, logistic regression, linear regression, and linear discriminant analysis, which assume that the features follow a Gaussian distribution for each class, and the t-Student pairwise analysis and the ANOVA hypothesis tests, which assume that the features follow a Gaussian distribution for each class and that the classes have equal variance. If the features do not meet these conditions, the results may lead to an inaccurate conclusion. In these cases, it is a good practice to correct the distribution of the variables. In this case, we can mitigate this issue by transforming the features so that they are more similar to a Gaussian distribution and check them using statistical tests. Some of the techniques often used to correct the data distribution include feature normalization through the Z-score method, the log transformation for right-skewed variables, and the Box-Cox power transformation.

4. Model Selection and Evaluation

The process of model selection is both an art and a science. First, it requires an understanding of the various learning algorithms and the type of problems for which they are well suited. There are many supervised learning algorithms, ranging from simple linear regression to complex ensemble techniques. In each case, there are tunable parameters, and the combination of algorithm and parameters is often referred to as a "model". Furthermore, a particular algorithm and parameter setting may work better for one type of input dataset than for another, sometimes dramatically better. For that reason, testing many algorithms on a problem is a critical part of practice. The results from an exhaustive search

for the best algorithm and parameterization are often used to perform a coarse test to gauge how well any model might perform and provide a reference point for assessing how well more complex models perform. The different flavors of machine learning are suitable for handling different classes of prediction tasks and from slightly different perspectives.

Given that any learning algorithm optimizes its own objective function, using a strategy that resides in the same space as that function can often yield the best results. However, each algorithm differs in terms of how flexible it is when it comes to modeling the data. Less flexible models such as linear and logistic regression rely on the representation of the data to provide additional information and guidance. By default, models induce from training examples that belong to the same class that they are attempting to predict. Flexible models, on the other hand, such as neural networks, decision trees, or support vector machines are less reliant on input representation because they can learn complex decision boundaries and representations through the learning process.

Aside from prediction time, which may be difficult to compute based on model choice, one of the more important performance metrics that affect the trade-off between model choices is the accuracy or loss function. In the supervised learning setting, the model evaluates the quality of its predictions based on the cumulative loss that results from predicting on each of the training examples.

4.1. Choosing the Right Algorithm

The words artificial intelligence have been around for a long time, but recent years have seen a resurgence in interest in this area, and now it is definitely one of the most popular buzzwords in technology as seen by the growing number of research papers, patents, conferences, corporate research centers, and dedicated efforts in AI. Modern techniques have been applied to problems including spam filtering, search engines, computer vision, and natural language processing. Among these techniques, machine learning has certainly gained a vast amount of attention, and it has even been said that AI is just another name for machine learning. Machine learning algorithms are based on the principle of learning from the examples provided by the user and generalizing on the underlying function.

As the machine learning landscape continues to expand, there will naturally be a growing number of machine learning packages, frameworks, environments, and libraries. Too much choice can lead to confusion. Researchers want to make reliable comparisons between algorithms to allow them to make appropriate algorithm choices. Therefore, a natural question arises: Given a new problem and a collection of different algorithms, how does one go about selecting an algorithm

that will perform best on that specific problem? Algorithm selection is clearly a key problem to be solved in order to leverage the power and promise of machine learning in practical settings. Algorithm selection involves given a data set about a problem and a collection of available prediction algorithms, selecting an algorithm that is likely to yield high prediction performance on the new problem. Repeating this process for various datasets, these algorithms are trained and evaluated. There are different criteria for the final selection of an algorithm among the pool of competing alternatives.

4.2. Performance Metrics for Evaluation

Introduction is one of the most challenging parts of the machine learning problem. But to do this correctly, we need to define the problem correctly. Then, we could explore many of the selected existing methods and test them in the available data. The true challenge is to perform a real implementation and test with unseen data.

Sure, we could calibrate our model with random selection. If our model performs poorly, it will immediately appear. But if the model has good performance, we would need a better way to evaluate. Because it could overfit our data or it could say the same data distribution we want to use in the predict. In short, the selection of one of the evaluation methods should take into consideration the objective and the type of model.

Once we decided the model, it is usually evaluated using a specific metric depending on the task. For regression tasks, the selected metrics could be R2, Mean Squared Error, Mean Absolute Error, and Relative Squared Error. R2 is a relative measure that gives us an idea of how well the model explains the variation of the output label. Mean Squared Error is the average of the square of the errors. Mean Absolute Error is the average of the absolute value of the errors. Relative Squared Error is basically Mean Squared Error but represented relative to the label mean.

For classification tasks, the selected metrics could be Accuracy, Recall, Precision, F1 score, and Area Under the ROC Curve. Accuracy is the ratio between correct predictions and total predictions. Recall is a measure indicating the ratio of true positives to the sum of true positives and false negatives. Precision is a measure indicating the ratio of true positives to the sum of true positives and false positives. F1 is a harmonic mean involving Recall and Precision. Area Under the ROC Curve is a simplification of the Receiver Operating Characteristic defining the area by selecting many points within the

graph of plotting Recall by selecting different thresholds with Precision selected as one or zero.

5. Integration with Azure Machine Learning

5.1. Connecting Microsoft Fabric to Azure ML

Microsoft Fabric allows interaction with Azure Machine Learning resources by using the Azure account service. Not only that, but there are great Azure Machine Learning components you can add to your Fabric workspace by following these steps: Open the Microsoft Fabric portal. Click on Add resources, on the top left of the page. Select Azure ML Component Catalog. Select a workspace or create a new one. Choose an Azure region. Click on Create.

Once the component is created, you can explore all the models in Azure Machine Learning, and see if they are compatible with the type of data that will flow through your pipeline. You can also create pipelines that connect Azure Machine Learning models and bring the data back into Fabric.

5.2. Using Azure ML for Model Management

When it comes to monitoring and managing those models, you usually have to do that using Azure Machine Learning Studio. This is not ideal, of course. You might prefer to do everything that is available in Fabric. You could connect Azure Machine Learning pipelines with the refresh of Fabric items but that is not ideal either.

This integration opens a few possibilities. First, if you use Azure Machine Learning pipelines to implement and manage the module lifecycle, especially if you use Fabric to implement or manage the data prep or the monitoring tasks, you have a unified infrastructure that can provide value for your organization. Second, with this integration, for a file of supported models, you can easily use AI features such as AutoML or Hyperparameter Tuning from Azure Machine Learning Designer, then use Fabric for BI reporting, paginated reports, or dashboards, which are great for visualization.

5.1. Connecting Microsoft Fabric to Azure ML

One of the most important features of Microsoft Fabric is its ability to connect to Azure Machine Learning and retrieve models that were developed there. The integration between Microsoft Fabric and Azure ML enables data engineers using the Data Factory app to call Azure ML models during data processing. Azure

Data Engineering was built with this integration in mind, and Microsoft Fabric allows data engineers to call Azure ML models directly in the pipeline without calling an HTTP endpoint. Using native connectivity, the model is called, and the output is automatically converted into a data frame so that data engineers can inspect it within the pipeline. The main use case for this capability, particularly in the Data Factory part of Microsoft Fabric, is data preparation and data quality tracking.

Machine Learning is a new feature in Microsoft Fabric, which lets data scientists and business analysts find and utilize ready-made AI capabilities that were created using Azure Machine Learning. It enables a one-click workflow to deploy models to Power BI for business users to use in reports or dashboards. It also allows for easy collaboration. Business analysts using Microsoft Fabric Machine Learning can create a simple report with the AI capabilities from a model, and then share it with data scientists who can improve it by modifying the model parameters.

There are 2 pre-requisites you need to meet to connect Microsoft Fabric to Azure ML. First, you need to have Azure ML workspace. Second, you need to have an OAuth 2.0 application registered in Azure AD and have permission to access this app. This app allows Microsoft Fabric to use service-to-service communication to connect to Azure ML.

5.2. Using Azure ML for Model Management

Just like a baby is born only after a lengthy turmoil inside the mother's womb, a model is born only after painstaking efforts of scientists and engineers amidst layer after layer of confusion. And a model is born only as a collective, with inputs from many supporting partners. Plus, a lot of engineering goes into wrapping a model with APIs and software automations, so that the wider set of users can use it with minimal skillset required. The platform manages the creation of your models, particularly machine learning models, by data scientists, the deployment of machine learning model inference services, by backend engineers, and the actual execution of the inference code with wrapper APIs, by software engineers.

Various types of models and assets can be stored in the platform's tab: ML pipelines, ML jobs, models, datasets, deployments, images, workspaces, notebooks, environment definitions, spaces, and datastores. Further, the required resource can be created directly from the platform if they don't already exist or imported to the platform if they already exist, by simple clicks. In fact, when you run a new ML pipeline or a job in the platform, under the covers it is completely

managing the backend machinery for you even if the service is not visible in the data factory. Make a remark about the resource selection. The platform handles where the processing happens if it is selected to run using private compute or using a remote workspace. You can choose to have your ML job run in either of them or just execute the job in the service. Users can even visualize their ML jobs in the studio page, where they can view the ML pipeline structures, jobs' logs and even pickle the output data for a job run.

6. AutoML Capabilities in Microsoft Fabric

Machine learning activities like data cleansing, feature engineering, dealing with missing values, and hyperparameter tuning require a significant amount of human time and effort not only to simulate the activities, but also to carefully design the algorithms that handle the work accurately on a project basis. Automating these activities can save a significant amount of development time and help experts focus on higher-level tasks like model selection or feature engineering. Autotuning is not a recent idea; automating model selection and parameter tuning has been an area of active research and the work on proprietary algorithms is now complemented by offerings from many proprietary and open-source frameworks.

Supported in some capacity by many Automated ML frameworks, the aim in the coming years is to create a full platform of AutoML settings to cover most needs of users interested in Machine Learning but without extensive experience on the field, or experts wishing to speed up their development. Aiming to bring AutoML capabilities integrated in the pipelines environment to such potential users, users can deploy AutoML capabilities immediately to generate predictive models at lower cost. Model training can be done immediately with AutoML capabilities. No additional costs are implied for model training; using AutoML capabilities is an additional option of Machine Learning.

6.1. Introduction to AutoML

The complexity of machine learning implementation has traditionally limited its accessibility, particularly for business professionals such as strategists, analysts, general managers, or marketing experts, who have little to no experience in writing code. This presents a missed opportunity, as the vast set of decisions that define a machine learning model's performance are optimally decided by the business person for the domain and field of the prediction task. As the predictors drawn from these models become more integrated into the fabric of daily

transactions and activities of a business, proper selection and tuning of model types is essential.

The introduction of automation into model selection and tuning - known as AutoML - is rapidly changing that. The latest frontier in AutoML presents end-users with as straightforward interface as uploading the appropriate data and specifying a prediction task without requiring them to consider model type selection, coding, or even the interpretation of model outputs. Further advances in AutoML aim to incorporate far more sophisticated user defined pipelines than the base model image augmentation, hyperparameter tuning, or evaluation stage parallelization. These pipelines could include complex transfer learning capabilities for popular domains, some even allowing end users to dynamically customize the architecture of neural network and the underlying philosophy of running neural architecture search - on which architecture to focus on selecting and tuning - presented as stress tests run on small held-out validation sets.

The capabilities available in Microsoft Fabric will focus on the input workflows and ease of use for the general manager, analyst, etc., without diving into much of the implementation detail. This decision is made not only for the sake of readability but also is reflective of what is the intention of AutoML itself - to significantly decrease the barrier of entry to machine learning techniques in daily business decisions.

6.2. Automating Model Selection and Tuning

In this section, we explain how users can automate model selection and tuning using AutoML capabilities in a low-code manner. AutoML enables you to automate the task of selecting algorithms and tuning hyperparameters to build the best possible model. It provides AutoML capabilities that allow you to get started quickly to build and deploy models with high predictive accuracy with little knowledge of machine learning. It automates several algorithm selection and hyperparameter tuning tasks, saving data scientists time spent on tedious low-level tasks and streamlining the model creation and tuning process.

Two primary reasons make AutoML valuable for data scientists and organizations. First, the demand for machine learning is outpacing the supply of skilled machine learning data scientists. Many domains require machine learning, but organizations do not have data scientists with deep domain expertise coupled with experience in the nuances of ML. Thus, many organizations are looking for tools that allow their employees to apply machine learning to solve problems without requiring deep expertise in the field. Secondly, even experienced data scientists tend to rely on tools that automate these tasks. As machine learning

models become increasingly important in business decision-making, organizations need to automate tasks that are often tedious or low-level for experienced data scientists.

7. AI for Forecasting Applications

Forecasting refers to the predictive analytics area that estimates values of future data based on the knowledge of past data. Time series forecasting has been an essential component of business intelligence and an area of continuous advancement. The renewed interest in time series forecasting research has been activated by the need that industries have to gain advantage from rich, high volume and high velocity data generated daily. Manufacturing, sales and financial business areas generate time series data that require continuous monitoring and that will greatly benefit from advancements in time series forecasting efficiency and accuracy. Intelligent differentiation of recurrent patterns from one-off outliers will allow analysts to trust forecasts even more. The business areas where forecasting is applied require accuracy above all, but response time and consumption of computing resources also play an important role. The approach for offering forecasting capabilities will depend on the resources available.

Time series forecasting has advanced mostly in two typical areas: the accuracy of individual forecasting methods and the efficiency of automatic combination methods. Autoregressive Integrated Moving Average models, and their extensions like Seasonal ARIMA and Exponential Smoothing, have been the dominant time series forecasting methods for decades. They are highly regarded by the forecasting research community. Still, when using fixed parameter smoothing, they only offer at most a heuristic way of capturing longer-term patterns. Recently, driven by the availability of large amounts of data, powerful computing systems and new paradigms in artificial intelligence, there has been a renewed interest in expanded time series forecasting capabilities. These advancements are aided by innovations in artificial neural networks like Long Short-Term Memory Networks and Convolutional Neural Networks. They are supported by strategies like Temporal Fusion Transformers, N-BEATS and unified time series deep neural architectures, as well as hybridization of traditional and modern statistical and machine learning forecasting methods.

7.1. Time Series Forecasting Techniques

When it comes to time series forecasting, the first and most well-known technique is the classical ones, ARIMA. ARIMA models belong to the family of models which capture temporal dependencies in data, that are mainly based on linear or auto-regressive terms and forecasting errors, through an integrated component. ARIMA is often a benchmark for comparison with other machine learning methods applied for forecasting. Exponential Smoothing is based on weighted averages of past observations, where the weights decay exponentially for older past observations. It is a classic naive approach with a formulaic and interpretable structure, which is often competitive with more complex procedures. Croston's method is widely used for forecasting intermittent demand. Prophet is based on the additive model, which decomposes time series into trend, seasonality, and holiday effect components. One interesting tool available in Microsoft Azure is the Automated Machine Learning approach of Forecasting. A sensible AutoML strategy with respect to classical forecasting techniques, is that often a simple classical model will be better than all candidate models generated from a complex machine learning approach.

Time series forecasting is a well-researched problem in computer science and dozens of packages exist to handle it. It is focused on the problem of estimating the future behavior given the historical observations, where the time-dependent component is and should be an important component of the forecasting task objective function. Common examples of forecasting applications in business that we often see, are sales quantity, stock, and balance prediction in Finance. Demand forecasting is ultimately what drives replenishment plans and logistics execution in Supply Chain. In Sales and Operations planning, demand forecast efforts are focused on attempting to find the quantity of products that customers will want over a specific time period, within a specific timeout window. Although this demand determination would be the ultimate goal, is not the only one and sometimes it is necessary to generate nice probabilistic forecasts for other linked variables through the forecast horizon.

7.2. Use Cases in Business Forecasting

Forecasting problems can be classified into two groups: business-oriented and science-oriented systems. Business-oriented systems comprise a majority of forecasting applications because business decisions require a longer time span than many other real problems. To help business decision makers at all levels of management craft their strategies, policies, and tactics to increase the probability of business success, scientists have developed forecasting models to predict what will be happening in the future with respect to factors, both internal and external,

that will significantly affect their businesses. These predictive relationships can be used to optimize business operations over time in the future, including product development and rollout, marketing communications, advertising expenditures and placements, media and sales alternatives, production plans, human resource levels, and cash flow and inventory levels.

The goal of business forecasting is to predict within a specific time horizon the future values of a variable crucial to business success, using information from the limited past of that variable and from one or more time sequences of other variables reflecting the decision environment, especially those that are unusual in some sense. To achieve its goal, a business forecasting model should shed light, if not provide answers, on the "why" of predictable fluctuations in the target variable, as well as on the "how" of long-run, cyclical, and short-run business decisions. Thus, it is the gap between the likely intuitive hypotheses and the quantitative forecasting results that weakens the impact of forecasting models in the business decision-making process. If "laymen" cannot appreciate the implications of a forecasting model, the forecasting model may be ignored or be used only as a supplementary decision-making tool.

8. Anomaly Detection in Microsoft Fabric

Anomaly detection, or outlier detection, is the task of classifying a given subset of patterns in the dataset that do not conform to the expected behavior or distribution of the majority of the data. Such anomalous patterns typically occur infrequently and are sporadic relative to the size of the dataset, negatively impacting the quality of the data analysis. Campaigns of computer breach attempts, thefts, defect patterns in manufactured materials are some examples of anomaly detection categories for various domains. Numerous tasks in AI rely on understanding the typicality of some characteristics, including what constitutes an anomaly. Anomaly detection is an important part of the larger topic of novelty detection, which happens whenever there is a known class of normal instances, but there may be entirely new classes of other instances.

Anomalies are data points that differ significantly from the rest of the data and can be either due to a data error or due to a substantive difference in behavior. Detecting outlier patterns in the data setup plays an important role in whether subsequent analyses and data preparation procedures will produce valid results. Outlier patterns can often hide important features that provide discriminative knowledge, such as major anomalies or exceptions, from the user, but they can

also be attributed to trivial data preparation processes, such as noise or data errors, that need to be solved first. Furthermore, outliers can also be due to interesting, significant differences among data sources. Anomaly detection in fabric is suited for problems in domain areas such as fraud detection, financial detection for stock movements, intrusion detection for network packets, fault detection in mechanical systems, sensor fault detection in a sensor network, and image outlier detection.

8.1. Techniques for Anomaly Detection

Detecting anomalities, or point anomalies, is a problem that is often encountered in various machine learning applications. It consists of locating a small set of data items that deviate from other items in the whole dataset by a significant degree. Detecting point anomalies is usually the first step in more complex data processing pipelines, such as in fraud detection, shape-based object retrieval, recommender systems, and score-based classification problems. Point methods do have some drawbacks. In particular, they do not take into account the relationships between data items, something that would allow capturing more sophisticated data models. For this reason, abnormalities can also consist of sets of items or a progressive error, in which the error for a data item is not that significant, but on a set of consecutive time frames, the error accumulates and becomes salient.

A Time Series forecasting scenario can be used to detect point anomalies in time series data, leveraging specific anomaly detection algorithms. To create a forecasted dataset of the Time Series scenario, you need to specify both the target variable to be predicted for each timestamp and a few additional timestamping-related parameters. The Time Series Forecasting module allows you to specify classify a variable as either a timestamp or a time identifier then generate a predictive dataset of future time series values that could leverage additional business rules. The Predictive model is automated, providing a time series error to evaluate the quality of the predicted dataset. You can also submit a Dataflow notebook in which you can add ad-hoc logic test or additional business rules, test and retry cycling different forecasting algorithms available.

Alternatively, if you do not want or cannot use a time series forecasting approach, a possibly more simple solution is to directly use anomaly detection algorithms that are specialty-designed to identify the presence and/or height of anomalies for each sample in a time series, without needing predictions for unseen timestamps. You can apply these anomaly detection methods if you have enough training data to accurately estimate the hyper-parameters of the method.

8.2. Applications in Fraud Detection

Technically, everyone knows someone that has been a victim of fraud. Either they lost their wallet, or someone charged several purchases with their credit card, which, of course, was left in a restaurant or bar after a long night out. But nothing compares with online fraud performed by criminal organizations that can charge millions of dollars from customers all over the World. Tech giants, financial institutions, and other companies that operate online have to deal with problematic users trying to use their services for illegitimate purposes. Organizations have begun to deploy online fraud detection solutions as an added layer of security to help the Fraud Operations team make the right decisions and minimize losses from fraud quickly. This allows the team to automate huge work centers that otherwise would be tedious and might lead to human error. These detection solutions also allow organizations to cut detection times, minimizing losses and chargebacks.

Fraud detection is especially relevant today because as technology evolves, new rules replace traditional strategies. The recent shift towards an increasingly automated customer experience has opened the door to ever more sophisticated forms of fraud. Fraudsters can exploit them using stolen credentials and insider data, offer counterfeit services, or claim for refunds or benefits they do not deserve after causing trouble. Additionally, fraud must be tackled across a variety of channels. For example, social media has recently emerged as a new vector for fraud. In this case, improving security for companies that use it as a business tool is quite challenging, both because it is a public channel and also because it often specializes in dominating wide audiences for little investment.

9. Decision Support Systems Powered by AI

Decision support systems have been, for three decades, about providing the right data to decision-making units. Named often as DSS, they provide support to semi-structured decisions, or choices that have more or less fixed procedures, but are not totally deterministic. In the past, these DSS employed descriptive and prescriptive models, like predictive simulations or optimization, or also descriptive statistics, where the results are not moved by the interpretation of the output analysis, but from given rules. These were also often complemented with business intelligence like dashboards and reporting. All these were data-driven approaches, usually based on past events. But how DSS 2.0 – Decision Support Systems powered by AI? What is the role of AI in modern business?

With today's AI and ML, different innovative decisions support systems are possible. For sure, the traditional DSS are still in place. However, AI is changing and augmenting the role of transparency, data driving, and automation, in cyber-physical systems. Every day, millions of hybrid decision support systems powered by AI analyze events, data or business scenarios, and predict what will happen, and support people decision-making, but also automatically optimize or choose the most preferred decision out of the decision model definition. Robotic processes, augmented analytics and intelligent process automation often utilize these capabilities to fill necessary gaps or run tasks that machines cannot perform automatically. AI can act as automatic decision-making in many business processes: automatic fraud detection, supply chain planning and optimization, vehicle routing, price optimization, and many others. AI can also support people making more complex decisions like demand forecasting, social media engagement, autonomous driving-aviation, product recommendation, victim seeking in the police force domain with complex stakeholder models, but all these are still in the DSS domain.

9.1. Integrating AI into Decision-Making Processes

The accelerating deployment of AI into activities such as question answering, text and image generation, response prediction, and vision understanding offers both new opportunities and challenges for decision processes at both the group and individual levels. Just as software mechanized how computers conduct tasks in a highly programmed way, and later cloud computing enabled efficient sharing, access, and deployment of that software, the latest generation of AI aims to reduce the tightly programmed nature of those tasks and use inferences to step into more intermediate creative stages. In support of enhancing decision-making processes in groups and alone, we discuss the benefits and risks associated with automation of intermediate tasks in a decision process. The goal here is to enhance decision processes by human decision agents rather than replicate the role of the decision agents.

One main reason why people group as a single decision unit is that they have the capacity to engage in constructive debates and enhance the information pool of the group beyond what each individual may bring to bear. It is this aspect of group decision activities that has been a focus of decision support systems, with their capacity to enhance decision speed in groups, introduce more rational analyses, and reduce bias in the information basis of a decision. Support system designs have focused on the stages of a decision process, with careful programming of how they operate in support of one or more human decision-makers. The programming of operations is based on normative models that

specify desired outcomes, while the use of AI is forecasted to support a more adaptive simulation or pattern matching basis. However, this more human-like operation raises novel questions of what the nature of the collaboration is and when is an AI system able to replace a human-based decision-support system?

9.2. Case Studies of AI in Decision Support

This section examines several case studies that demonstrate the use of AI in organizational decision support. The focus is not on the technical details of AI implementation, but rather on how the technology is utilized to enhance the decision support capability of an organization and the impact on the organization. The cases demonstrate the variety of techniques employed, types of AI applications, areas of organizational decision making that AI is applied to, and the different types of organizations that are adopting AI.

The first case study examines the use of AI in forecasting and demand modeling at a power corporation. The paper describes how the corporation uses neural networks to predict the power demand for its regional grid, using many variables, such as weather patterns, date-time seasons, temperature levels, and weekday/weekend patterns. They use both a 2-month-ahead monthly seasonal pattern forecasting approach as well as a multi-time step forecasting methodology for AI-based load forecasting. The implementation of AI techniques on these demand forecasting processes has reportedly led to an improvement in forecasting accuracy of about 20-25%. Improvements in forecasting accuracy are expected to yield utility cost savings of around $87 million. While this case depicts the implementation of a decision support tool on a strategic organizational business process, it is also pointed out that data scarcity may hinder the implementation of AI techniques in forecasting.

Another case study describes the role of Artificial Neural Networks (ANNs) in supporting the tactical decision of portfolio construction of an investment fund. The process of feature selection for generating an optimized ANN model for predicting excess returns over a variety of time periods is discussed. It is demonstrated that an ANN-based selection mechanism yields excess returns that are higher than those obtained using traditional multi-factor models for asset selection in investment management. The results indicate that the use of ANNs to construct tactical asset portfolios could lead to better performance than that obtained from conventional models. This will thus aid fund managers in the tactical decision of building an optimal portfolio for enhanced returns.

10. Deployment of Machine Learning Models

Deployment of Machine Learning (ML) Models

Unlike traditional software, machine learning (ML) models are not static entities. They continuously learn from user behavior, and an ML model need not be static in the sense that it cannot be updated or changed after deployment. At some point, a deployed model needs to be routinely refreshed or even redeployed. This chapter provides an overview of the operational aspects of deploying ML models.

The first section covers best practices for deploying models. With the right methodology, deploying ML models can be easy and straightforward. The second section discusses the importance of monitoring deployed models in Production. Monitoring is key in understanding the incoming traffic for your model, and key KPIs to monitor include latency and request volume. Lastly, we discuss the need to maintain deployed models so that they remain useful and relevant, presenting and discussing automated and manual strategies to maintain deployed models over time. After reading this chapter, you'll have a good understanding of how to operationalize deployed models, and how to keep your deployed models updated and relevant.

Your business is unique, and your ML workflows will have differing levels of complexity, specialized demands, and coding requirements for your organization. Due to these differences, there is no specific model development or operationalization methodology, but instead a range of processes across the ML lifecycle, from development to validation and deployment.

10.1. Best Practices for Model Deployment

Empowering users to create their own predictions from the models you have created for them allows them to perform actions based on those models that they would not otherwise be able to do using standard reporting tools [1-3]. When deploying a model to a user, in simple terms, you are effectively creating an API element, and that means developing features that should really be there in a commercial system. Your actions will probably be driven by the intended use, the importance of the model within the business, and whether or not the model will be repeatedly used. In terms of deployment, you may allow users to define their own parameter settings, or you may have some default values. Do you include a description of the model? Will a user applied with the same data that the model was originally trained on get the same scores? Is it easy to interpret when you may be using different data to that which was originally used? Is there some contextual help? Will the user understand the other information that is often

included with a model, such as the algorithm used or the feature importances? Have you played with the data you are using to deploy the model so that the deployed model is more personalized, especially if it is being used for recommendations? Have you considered automated testing of the model output? What is the speed of response? If the predicted output is to be used within a day-to-day system, but the user must wait a long time for results, what will the impact be?

10.2. Monitoring and Maintaining Deployed Models

Machine Learning is about solving business challenges. Deploying a machine learning model is just the beginning of the journey toward solving that challenge. To succeed you need to monitor and maintain the deployed model throughout its lifecycle. There are many model management capabilities that are parts of the Data Science solution. In this section, we will dive deep into the capabilities and tools that can help implement machine learning model monitoring and lifecycle management. Remember, what can be measured can be improved!

Once your model is deployed, you need to make sure it is correctly serving predictions all the time while also solving the business problem that it was initially designed for. There are several strategies to consider in order to monitor your deployed machine learning models. Monitoring the predictions that your models produce is arguably the most important and straightforward approach. This is usually called Bias Monitoring or Prediction Drift Monitoring. Whenever your model is producing predictions that are biased or that have significantly shifted from what you expect, you may want to indicate an issue. This can be related to data distribution changes in the feature used by the model. Being able to explain why predictions are different over time is important, this is typically called Prediction Explanation or Model Explainability.

More rigorous oracles can be based on prediction performance. This would require actual ground-truth data, thus is much harder to implement. One approach to leverage in this case is Monitoring Model Performance or Quality Monitoring. Your prediction performance could also be sensitive to a number of user or task-specific factors, you need to factor in different segmentation dimensions.

11. Ethical Considerations in AI and ML

AI and machine learning make decisions and predictions affecting people's lives in many different ways: from deciding who is accepted into a university or hired

for a job, to who is offered a loan or living accommodation, to who is suspected of criminal activity or is recommended for parole, among many others. There is an increasing body of evidence that these and many other uses of AI and machine learning are biased in important ways, and that they are, therefore, unfair to certain races, ethnicities, and genders. As such uses of AI and machine learning are increasingly used to assess how tailored educational programs can promote and improve students' interest in mathematics, it is imperative that there be dispassionate inquiry into whether such tools in fact are fair. Other important and related issues relate to how the training data and model evaluation processes reduce bias, what level of model performance is considered acceptable for different groups, which fairness metrics are considered relevant, and whether use of AI and machine learning in fact is the appropriate or best tool in the particular context.

Vanquishing such discriminative bias in how AI and machine learning systems are constructed, evaluated, and implemented is obviously not sufficient: AI and machine learning applications raise many questions about privacy and data governance raising significant academic and policy interest. Examples of such concerns in the many areas of AI and machine learning deployment include data collection, data security vulnerabilities, inappropriate use of captured data, unauthorized sharing and third-party access, model transparency and explainability, and user awareness and consent. As rapidly expanding and pervasive technologies, the ethical and regulatory challenges surrounding artificial intelligence and machine learning technology are vast and encompass legislation, guidelines, industry-wide best practices, the auditing of algorithms, and the ways forward.

11.1. Bias and Fairness in Machine Learning

In this chapter, we focus on ethical considerations in AI, gather insights, and comment on what is done to mitigate the discussed issues. We begin with the most discussed theme in AI, and that is the aspect of bias and fairness in the suggestions of AI or ML models. While we have mentioned ML model development as a technical topic, it is governed by what is social to an extent. When we give AI/ML the data and instructions to learn — we are doing it with an assumption of being unbiased or not being favourable to certain aspects. The choice of input data and its representativeness put the aspect of a technical guideline's social governance. Certainly, if you want people to accept AI or ML suggestions, it has to be a representative suggestion governed by careful consideration of input features, i.e., data.

Bias may arise from groups in the data, be it from gender, age, religion, or other social aspects, and we have seen too many instances of negative AI effects arising from biased suggestions. The more pervasive AI and ML become, there are greater negative aspects to the society that are possible to arise from these inputs riddled by bias. Hence it is critical for us developers of technology that we see fairness as a guiding principle in creation ML models. This would require addressing the issue of potential bias on different factors in representative ML modeling. Additionally, it is encouraged that organizations take measures ensure these principles are followed, involving proper governance practices. And finally, there are different tooling and frameworks which can be leveraged to take a precautionary approach addressing ML bias.

11.2. Data Privacy and Security Concerns

Microsoft Fabric democratizes analytics, and its AI and ML capabilities offer a radically faster way to build and train ML models. These capabilities unlock new ways for everyone to gain faster insights from and make predictions based on their data. However, as AI and ML technologies advance we must be mindful of new ethical concerns that arise. In this chapter, we outline the top two concerns that should be considered while working with the Microsoft Fabric workspace: the risk of data privacy exposure for shared workspaces and community datasets, and the inadvertent propagation of biases that can be imparted by AI-generated or community-published resources.

There's a risk that a user in a shared workspace can see sensitive data not intended for them, even in workspaces that implement RLS. This is because not all components of a shared workspace implement RLS, and also because RLS only protects the datasets during data preparation and cannot prevent sensitive data from being exposed while model training is happening in the background. Furthermore, some of the resources deployed to the shared workspace can be traced back to the actual path of components. However, the temporary components can be identified only when the pipeline is actively running. Currently, Microsoft Fabric does not support allowing other people to share a workspace without also getting access to sensitive data; this may happen in sporadic cases where RLS support is not consistent. As Microsoft Fabric moves to being generally available, we recommend using Microsoft Fabric in an environment where trusted collaboration among its users is already enabled.

12. Future Trends in AI and Machine Learning

Advances in computing power, cutting-edge algorithms, and the availability of vast troves of data have led to unprecedented momentum in AI. Major enterprises have dedicated enormous resources to AI, and competition for talent is fierce. No longer just a futuristic vision, AI is now an integral part of business life, and its future development will inevitably forge profound changes. Mastering the power of AI—by deploying machine learning and embedded statistical methods—at scale throughout an enterprise will be a key element of successful transformation in coming years. The technology landscape is littered with brave startups that introduced innovative new tools—some successfully gained traction, some flared briefly and then faded away, and some were acquired by larger players. Today, many aspects of machine learning are maturing, and the future trend is toward integration and embedding advanced analytic capabilities in familiar decision-making tools. Predictive has been integrated with dashboarding and geospatial visualization in enterprise-centered solutions. Business process applications are beginning to embed services that interact with predictive models, optimizing key functions like portfolio management and customer care; other enterprise applications expedite application development—with variations that add the ability to deploy at scale or tailor models for special applications. Programmers will also enjoy the productivity advantages of integrated tools—whether full-fledged data science development workstations or hybrids. Business leaders should plan for the future of AI by investing in integrated tools to empower talented developers and business analysts from across the organization, and by developing partnerships with trusted vendors who can help them meet the more specialized analytic needs of different functions. AI will be in the background everywhere—making predictions, augmenting decisions, automating processes—enabling businesses to operate at a new level of efficiency. They can't afford to wait. The time is now.

12.1. Emerging Technologies in AI

Generative Artificial Intelligence (AI) tools continue to break through across industries and functions. By far, the biggest change – and highest potential – is Generative AI, where a machine enables the creation of text, images, and code. The worldwide market for Generative AI is expected to reach $43 billion by 2023, providing further fuel to our economic recovery. Although, among Chief Information Officers, Generative AI has provoked the widest gulf of opinion.

Large Language Models (LLMs) are changing the AI landscape fast. Only a handful of components – not big nor expensive nor special in their architecture

nor in their training – are enabling a diversity of enterprises and startups. But these LLMs are becoming a world where everything and anything is available. Everything is hoarded; the market is in consolidation phase, awaiting a second burst to flow. Not everything is positive. Trained LLMs hallucinate. The biggest stakeholders are working on Fine Tuning Solution. It will come – it'll just take time. Chatbots are in front of our face.

GPT-3 is often referred to as the "Dungeons and Dragons game master for everything." And it's true. The most popular at ChatGPT is based on GPT-3 architecture. How did that happen? Because language can be scaffold for human's imagination; LLM can be that game master for all our creativity. GPT-3 is based on a simple workhorse architecture that's been around for a long time and can be reproduced in less than a month.

12.2. Predictions for the Future of AI in Business

There are countless trends surrounding artificial intelligence in business, and most of them seem to tell the same theme: change may be here to stay. Some of the most commonly noted AI trends for 2023 (and beyond) are hyper-automation, more focus on model regulation and auditing, natural language processing improvement, increased multimedia utilization, business model and operations restructuring via digital twins, AI evolution, embedded intelligence, and more explainability are all the trends operating inside the business. For AI in business, the following trends are most common: personalized consumer experience, augmented intelligence, more voice assistants utilizing AI, customer data utilization, employee efficiency training with AI, conversational virtual agents be improved via AI.

Demand for AI tools is expected to expand rapidly within businesses given the breadth of their applications. AI for IT operations tools and business intelligence and analytics are the fastest-growing segments in enterprise AI, and they represent large parts of the overall enterprise AI market today. Enterprises are witnessing data flooding in from multiple sources across the organization, at a pace and scale never seen before. Business decision-makers who harness the capabilities of AI and deep learning tools inside intelligent business applications will reap fast benefits - whether their goal is to embrace state-of-the-art, automated visual data-oriented decision processing for artificial intelligence and deep learning or other approaches that combine script-based and abstracted deep learning AI training with effects-based simulation, development and testing, model training, retraining, or deployment. For enterprises with business application models that are too tightly coupled, it can be very challenging to

incorporate different artificial intelligence technology approaches for different aspects of the business process.

13. Conclusion

This text explored the extensive capabilities of Microsoft Fabric and how these capabilities are integrated and delivered through a cohesive experience. Microsoft Fabric lowers the bar to be able to take advantage of capabilities such as Data Engineering, Data Warehousing, Data Science, Data and Business Integration, Observability, Real-time Analytics, and Machine Learning Collaboration. These abilities are made easier and faster with integrations powered by native support for Serverless SQL, KQL, and allowing easy integration with services. Furthermore, the low-code environments such as Data Factory's experiences for pipeline orchestration, Power BI's dataflows, and OneLake's file management and discovery center make it easier for users to get started.

Despite the low-code, no-code approach to making it easy for many others to access these capabilities through Microsoft Fabric and Azure, no enterprise strategy for implementation and practice of Data, AI, and Machine Learning can exist without Policies and Governance. Organizations that neglect this important area do so at the risk of eroding trust in the use of Data and AI Tools. We recommend that customers develop a comprehensive Policy and Guardrail Strategy and maintain awareness and checks on the level of Shadow IT that exist within their organization. Without proper policy and governance Management can quickly become paralyzed with the challenges of managing their environments. Azure provides a wide range of functions and capabilities to support this function. It is important to enumerate the responsibilities of Organization's Teams for Cloud Security and Governance, and the key Building Blocks that Organizations can use to implement their Cloud Security Strategies and Governance. By developing automated operational and governance security solutions, Organizations can build and strengthen their Cloud Security Posture.

References:

[1] Shivadekar, S.(2025). Artificial Intelligence for Cognitive Systems: Deep Learning, Neuro-symbolic Integration, and Human-Centric Intelligence. Deep Science Publishing. https://doi.org/10.70593/978-93-7185-611-9

[2] J. Lawless, "Microsoft and the Future of AI," 2018.
[3] J. Bosch, I. Crnkovic, and H. Holmström Olsson, "Engineering AI Systems: A Research Agenda," 2020.

Deep Science Publishing, 2025
https://doi.org/

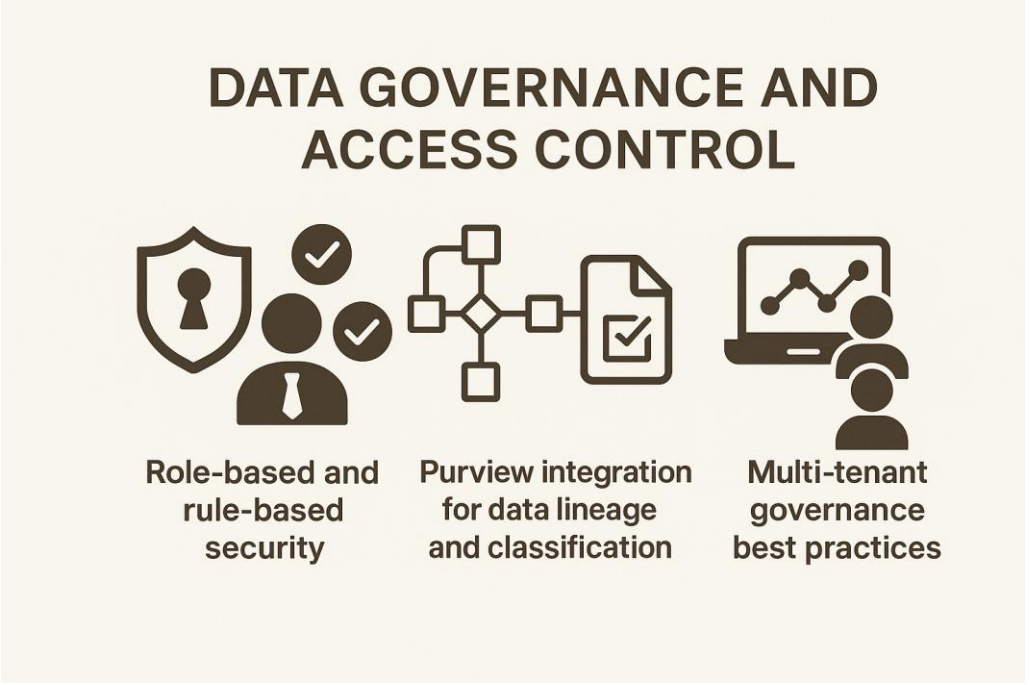
DeepScience
Open Access Books

Chapter 6: Data Governance and Access Control

1. Introduction to Data Governance

Data governance refers to the strategic process of managing and overseeing data within an organization. Defined as the overall management of data availability, usability, integrity, and security in an organization, data governance ensures that data is consistent and trustworthy and doesn't get misused. An effective data governance strategy incorporates architecture, policies, practices, and procedures that appropriately manage the full data lifecycle needs of an enterprise. Data governance enables an organization to ensure that data is accurate, available, and secure – meeting both the business and regulatory needs of the enterprise. Data governance promotes data stewardship and all data elements as enterprise assets that provide business value to the enterprise and its stakeholders. Data governance is made easier, and the results are better if the organization has a strong data architecture.

Data governance involves a lot of people and processes – people who create and maintain the data, people who govern the processes, business processes that are triggered by events who create/update/delete data. Organizations must be able to see and analyze the data lifecycle and have an inventory of all data assets related to business functions. What is not often recognized is the myriad data interfaces that cross organizational boundaries and involve many people in the collection, maintenance, use, and dissemination of the data. Data also flows through many systems often across third-party vendors. These pathways can lead to data errors or omissions, misinterpretation of what the data represents, or inordinate costs to produce the data. Data governance provides accountability through processes and policy enforcement to mitigate these risks.



2. Role-Based Security

In information security and access control, role-based access control (RBAC) regulates data access based on the roles of individual users within an organization, making it easier to manage users based on their jobs or responsibilities. By assigning roles to individuals, administrators can more easily manage file permissions that control users' access to particular data or applications. RBAC is a standard for protecting access to databases inside the health care and medical industry. Role-based access control overcomes the limitations of traditional discretionary access control and mandatory access control. However, role-based access control is not a panacea. Problems like role explosion, role mining, and dynamic separation of duty exist in role-based access control.

2.1. Overview of Role-Based Access Control

The bulk of computer security mechanisms are based on mandatory policies for disclosure and access to information, and rely on cryptography for confidentiality and integrity of data. Disclosure policies either restrict or allow the distribution of information based on the originator or by commonality of interest. Disclosure

policies are generally embedded in data and are enforced by facility security officers. Access policies prescribe the conditions under which authorized users are allowed to read and modify specific objects. Access control allows the secure sharing of data stored on programs or objects with other authorized individuals, while preventing access by those not authorized to view the information. Subjects are granted rights to objects based on associations with the users that have the rights. It is at this level that role-based security begins.

Role-based access control provides the ability to manage information security requirements of large organizations efficiently. In modern organizations with hundreds or even thousands of employees, access to electronic information is shared based on a security clearance, work function, or job responsibilities of users, not on individual preferences. As a result, the access control list checked for eligibility to view or modify security classified data is very large and very seldom checked for accuracy against personnel actions. Users in the same group must be able to share information with each other, yet without relying on a lengthy and rarely monitored access list. Role-based access control allows users to get subroles within a system, as well as define new objects and specify the subroles allowed to use them. Such flexible security requirements cannot be met by traditional methods, which require the access control list to be modified each time there is a personnel action. In addition to ease of administration, role-based access control provides the added benefit of enhanced security.

2.2. Implementing Role-Based Security

Once the roles have been defined, implementing a role-based access control (RBAC) module consists of the following steps:

1. Identify user roles and associated permissions: While setting up an RBAC module it is important to first identify all user roles and all system resources. The security administrator then associates roles with permissions. During this initial setup the permissions define what users are allowed to do with the system resources. For example, a controlled object could be a business document or utility files or utility programs that are normally run only by programmers. The permissions define the allowed operations as well as any field-level conditions, such as access and update rules, or locked fields. The operations are: Read, Write, Delete, Add, and Execute. RBAC supports the use of structured data definitions created by the Data Definition Modeller to examine the definition of user-defined data for field-level access control.

2. Assign users/roles: Once the roles have been set up, users can be assigned to a role. Different RBAC implementations have varying methods of making this

association, and different methods apply to different roles. The assignment of functional roles is typically done as part of the user provisioning process by the security administrator. Users can be assigned to a role based on a variety of criteria: they have the same job function, share a common task, are in the same department, or need access to the same data for auditing purposes. Remember that there is a one-to-many relationship between users and roles.

3. Task execution: At this point, the system is ready to start accepting transactions from users associated with each role. These transactions may read, update, delete, add, or execute various objects according to the pre-defined situation rules associated with the role. For the duration of the transaction, the system knows which user is associated with which role, and it automatically uses those role associations to control the specific access and operations allowed.

2.3. Benefits of Role-Based Security

The motivation for role-based access control (RBAC) was to address the limitations of identity-based access control (IBAC) systems. The most important tasks are to reduce risk and work involved in managing permissions, reduce risk and work involved in managing roles, enable consolidation of disparate identity management systems, enhance compliance, and unify disparate access control mechanisms behind enterprise security. This section will only touch on some of the benefits of role-based security to provide motivation for investing in it. Security must be considered in any data governance strategy.

Reducing Risk of Increased Identity/Role Explosion

Missing role hierarchies, coarse-grained IBAC mechanisms, insufficient resources, and simple facts of life — poor role design, the use of temporary identities, and the excess privilege problem — result in identity/access pair explosion. These problems lead to excessive numbers of roles, identities, or both. Excessive numbers of identities increase administrative burdens. More importantly, conflicting or redundant permissions create security loopholes, which put the data, applications, and platforms containing sensitive data at risk. Protecting sensitive data is a key compliance activity for organizations processing personally identifiable information, health, or financial data.

An access control protection overlaps between the data items being viewed and the actions being performed. Hence, the design of identities, roles, and permissions plays a vital role in risk assessments. A multiple assessment cannot measure risk and compliance without considering the potential for cascading dual-use resource-to-resource convergences. An organization should first conduct these assessments for multiple resources, including resources that are

dual-use by sensitive databases and applications. Risk assessors must quantify these risks, as there must be negative multipliers associated with repeatedly crossing the individual risk thresholds.

2.4. Challenges in Role-Based Security

The large web of interdependencies between roles within role hierarchies is difficult to manage. The trusted model — and, to some extent, the support for subordinate roles and privileges, and constraints on the dynamic modification of roles — is also difficult to manage. The more complex the role structure, the more difficult it is to troubleshoot role-related security problems. Role-based security, while it might simplify certain tasks, does not necessarily remove the need for experts in security technology. The shared roles within organization structure may also lead to an information security violation when an organization requires a very short time span for tasks that demand the predefined role memberships.

Like rules, policies and any kind of external processing, authorization using RBAC becomes a bottleneck, with a performance hit on the speed that operations can be run. This might also incur considerable effort for decisions that due to enterprise size might have to be checked many times. The activity for role assignments, or at least for their administration, might lead to reduce the speed of the business operations. If an organization is too rigid and assigns roles to users for a large period of time, there might be problems with activity overload or of an administrative bottleneck. For this reason, prioritizing the assignment of roles in smaller time frames or defining periodic time periods and review, monitoring and updates either for assignment or just for review of assignments are recommended. Automating role assignment is a solution using organizations' transaction activity. The integrated data warehouse allows this monitoring.

3. Rule-Based Security

Because of the importance of data to an organization, and the necessity to control access to this resource, access control has always been a concern. However, many traditional techniques provide limited flexibility. Rule-based access control moves beyond this by allowing the policy administrator to specify rules governing access to data in different situations.

In rule-based models, a combination of attributes defines a rule that provides or blocks access to an object. Specifically, an attribute of the user, an attribute of

the object, and an attribute of the environment can define a condition that supports access control decisions. When there are no additional specifications regarding the user, object, and environment attributes, they can have different meanings in different specifications, allowing for significant diversity in rule-based access control.

3.1. Overview of Rule-Based Access Control

A Rule-based security consists of a set of access control rules which enforce access control for user operations on objects. The Rule-based system is designed considering a security kernel that provides the following functions: policy management, policy enforcement, and audit management. Policies define what rules are part of the access control system and define the actions for users and groups, domains, objects, and operations. Simple operations support the creation, deletion, and updating of a set of objects that define the execution of the security kernel and require visits to the kernel every time access control is needed. Rule-sets can contain different types of rules, which specify roles or just access operations that are allowed per user per object.

The policy decision function determines whether the operation of a user on an object is denied or allowed. A message with the action payload is sent to the audit component that stores the actions performed on the objects or user operation attempts that were denied. Auditing the events of execution of the security kernel is highly important because it provides the administrator with the activities during which users accessed the objects and, more importantly, with the attempted accesses that were denied by some rule in the set. Auditing is also useful for performing a “policing” of users because it can give the administrator an indication about the correctness of the rules, especially the ones based on users’ roles.

The domain set describes the operations of the users authorized to perform on specific objects only. The Users-to-Groups set lists the users assigned to a group and that execute operations authorized for that group. The Groups-to-Domains set defines the domain user-to-group accesses; however, domains may be defined without being assigned to user groups. Thus, group membership is not required to specify operation access control policies.

3.2. Implementing Rule-Based Security

This section addresses an important issue in access control for information systems, the rigorous identification of security policy, and the implementation with the help of an executable security interpreter. Current implementation techniques - using host-based protection or network-based protection - have

serious shortcomings. They lack a clear specification of the security policy. This is enabled by a security kernel that includes the host-based protection of a few specific important resources used for communication and resource sharing. The architecture we present in this section provides the capabilities for the implementation of complex security policies.

Authoritative protection requires a strong separation between trusted and untrusted operations. It can be implemented by inserting hooks into system calls, using procedure call interposition or software switching between contexts. The analysis of network traffic - monitoring of packets entering and leaving the system and checking their consistency with a communication policy - requires the capability of observing the full range of communication activities. For devices such as Ethernet where the observation can be done by any of the hosts, this is easily achieved. On the other hand, built-in hardware security cannot provide access control to devices on the same level of abstraction as the other security services. The hardware can merely provide execution observance.

In the architecture described in this chapter, a monitor process triggers action in the enforcement mechanisms upon acceptance of data on exit, or channel or the detection of an entrance by special network traps. It must be realized that the monitor just facilitates a convenience service to the different protection modules and operates at a lower level than them. It removes from the kernel the unnecessary overhead due to protecting the whole activity in the kernel. The implementation of rule-based security policies is possible with devices since its semantics directly support the abstraction of policy, service, and activity.

3.3. Benefits of Rule-Based Security

This section reviews some of the reasons for using rule-based security, which are concrete advantages over traditional discretionary and mandatory access control systems. The benefits are presented in four different axes: explicitness, adaptability, performance, and expressiveness. First, RBAC has an explicit connection to the data model of the application, so it can be refined along with the data model. For example, the data and permission for a GIS-based application can be enriched by adding new elements to the GIS model and defining new permission along the line of the available RBAC rules. The model over which the security kernel is defined should evolve at each new version of the kernel. The evolution can lead to unpredicted results. The results can differ from a version to the following one. This fact is troublesome, especially at certification time.

The use of a rule interpreter, implemented as an Application Programming Interface, allows a fine-tuning of ACL security. For example, several users could be allowed to read (and not write) a document in the next three days in order to view its content at a given time but not change it. In this sense, the implementation of application-dependent elements is as easy as customizing an SQL query. The kind of capability is difficult to implement in traditional ACL models. From a user's point of view, the flexibility of the permission is important. Users would expect the same requests to produce the same answers from the system, assuming all data have not changed, as well as expecting temporal "what if?" parameters. From the above discussion, on expressing temporal constraints on the permission or the role, it may appear that RBAC is more restrictive than Discretionary Access Control. In this sense, there are no DAC models like the one proposed. This is just a particular implementation of the idea.

3.4. Challenges in Rule-Based Security

Many organizational policies are expressed naturally in the form of rules. For an e-commerce example, assume people from a company are neither allowed to sell to customers from another company, nor to sell certain items. A general solution to this problem is requiring users to identify their customers' attributes with a certain granularity before the operation. However, this may not be possible in practice for business.

The simple, primitive approach to rule-based security is to augment the list of the relations affected by each operation. An additional costly check may be necessary every time the relation involved is accessed. Tuple addition is required at insertion and update times, and net relations corresponding to deletions must be removed afterwards.

Advantages, however, come at a cost. The implementation adds complexity, perhaps incurs extra overhead, and scalability may be an issue. Complexity arises mainly because rules may depend on user's current properties – and these properties can be very dynamic. A natural solution is caching. However, both security schema for decision caching security are challenging and implementation of a particular caching strategy is complex as different configuration options are possible for efficiency, performance, and security. Caching may also not be valid semantics in certain situations, or it may have certain limitations.

With regard to efficiency and scalability, an example is when the number of files in a file system is very large, and each one has a different owner. In such a case, a user is enforced to have rules on all the files each time a new file is created in

the system. However, rules are not cached at the initial time and are actually gained at the final cached time. Hence, at the initial state the performance may be too low and the scalability may also not be valid, as adding rules on a large decision cache may be prohibitive.

4. Integration of Purview for Data Governance

Data governance defines how data is formally managed in an organization. Data governance is built around the processes that define what data is created, where it is stored, how it is secured, and on which platforms it is available to business users. Access control is an important aspect of data governance. Many organizations have visibility into the summary-level information of the data they store but lack a fine-grained view into who has access to what specific datasets at what level of access throughout the organization.

Microsoft Purview provides an integrated data governance solution that provides organizations with visibility into both data assets, as well as data access. With Microsoft Purview, organizations can create a comprehensive data map across the hybrid landscape in their data estate, including on-premises locations and third-party data clouds. Microsoft Purview enables organizations to classify sensitive data throughout their data landscape, manage how this data is secured, and govern its usage.

4.1. Introduction to Microsoft Purview

Data, an immensely valuable corporate asset, must be competently managed and secured uniformly across proprietary platforms and third-party service providers. A suite of tools provides a unified data governance solution. Organizations can manage, classify, and monitor data for security and compliance using data catalog, data loss prevention, and data classification tools. Implementing data governance via this suite provides organizations with control over their environment. Data governance enables organizations to understand where their data is coming from, what its contents are, how it is being used, and whether it complies with company regulations and local laws.

The suite was previously known for focusing on data governance for data within a specific platform. Services have since then expanded outside of that platform, with the introduction of a compliance portal that implements services for various applications and data. In addition to the compliance portal which centralizes data loss prevention and data classification policies for the services, organizations can

utilize security risk management tools that leverage classification capabilities to secure and protect other third-party solutions integrated into the ecosystem. These platforms allow organizations to deploy and manage data governance, security, and compliance across disparate data sources from a single solution or integrated solutions, to ensure that sensitive data is properly secured regardless of the location of the data.

4.2. Data Lineage and Classification with Purview

When it comes to enterprise governance and compliance policies, the challenge to align business lines with information technology teams seems to have taken eternal proportions. It is common practice for larger companies to have security, compliance, and policy departments that will model what needs to be implemented at the business line level, but then there seems to be a battle for positions long forgotten in history. Information technology departments then implement rules that are gold-plated and thus very difficult to maintain and evolve through time. This is rather dangerous every time that data required by the business line for operations and analytic purposes is categorized with overzealous rules. Accessing and analyzing data during operations will usually require assistance from those same security and compliance departments as the business line's users of exchanged information are not allowed to speak to the information technology staff.

The business line will hire consultants to understand how to break through the barriers planted from the information technology team to the rest of the business. Thankfully, the use of fiber optics, copper wiring, and cloud computing resources have made it easier for all the parties to build a marketplace of data requests and exchanges where information technology teams can ensure data governance, company compliance, and cyber defenses. This allows the creation of a data catalog that will usually contain descriptions of corporate data which are the application and configuration server under management by the business line for both operational and analytic requirements. It also allows capturing hybrid data flow patterns, whether managed on-prem with services or in the cloud using various data integration tools and numerous partner services and applications provided directly through various platforms.

4.3. Best Practices for Purview Integration

Once you have calculated the costs and have an outline of how you will utilize the Azure Purview service, consider taking a look at a couple of the following best practices during the design and implementation phase.

Understanding the native and external data sources supported by Purview is crucial — reviewing all of the capabilities associated with each data source to ensure the right decisions are being made. If the way your organization stores documents changes, and you have trained Microsoft Purview to understand the document types you have uploaded, re-scan your SharePoint Collections so that Microsoft Purview can detect the new information. The incremental scan option allows you to scan your existing document collections without incurring the cost of a full scan.

If you haven't started your Microsoft Purview implementation, consider doing the implementation in stages — especially with scanning your data sources. Start with the most critical data sources utilizing full scan options with the classification you want to evaluate. Once you have an idea of how classifications are being applied to potentially sensitive data, you can customize the scan settings based on that information and create schedules for the other data sources.

Endpoints are an efficient way of optimizing a pipeline for data ingestion; however, Purview has a limit of two endpoints per data source. Consider utilizing one endpoint to ingest into a staging area where other services can facilitate the access control patterns that your organization developed for that data, and then populate the subsequent endpoints in the data source that will land in the fields as your Azure Purview instance is scanning and analyzing.

5. Multi-Tenant Governance

Overview Capitalizing Cloud gains involves more than just building and hosting application services on shared pools of resources. It also requires creating an appropriate governance structure that aligns with the architecture, processes, security, and costs in a multi-tenant model. Right at the start of your cloud journey, it is important to decide on a governance structure that transcends an organization. But a guarded approach should be taken in terms of fine-grained controls. With multi-tenant use of varied workloads over large sets of shared infrastructure, having a flexible but well-structured cloud ecosystem for resource management and policy creation will reap long-term benefits. Like most security mechanisms developed for information systems, governance for cloud systems aims to facilitate the core objectives of confidentiality, integrity, and availability. Multi-tenant resource management is further supported by bringing in components that create configurability, elasticity, economy, manageability, and sustainability.

Multi-tenancy provides the flexibility to deploy external-facing applications or provide application development and hosting services to business partners and customers. Enterprises today are looking for fast-tracking their IT-devolved efforts and extending IT services to their partners and customers. With inherent support, features to customize look and feel and workflows, and minimal or no coding, external applications can be built and managed by the users themselves. With multi-tenancy, multiple self-service environments can be created and operational. While the internal workloads may be governed and secured by the enterprise, multi-tenancy simplifies the governance of the external workloads associated with the partners and customers. Appropriately architected cloud services can facilitate secure transformation of a variety of global business processes across functions.

5.1. Understanding Multi-Tenancy

Multi-tenancy is generally referred as shared services. Multi-tenancy makes service providers more efficient by off loading the management of the infrastructure and platform. Multi-tenancy gives a provider the ability to share the IT infrastructure and software application across multiple clients, or tenants. Multi-tenancy is a single-instance, single-version platform that is shared by all users or tenants of the cloud. Resources are effectively utilized by hosting multiple, and sometimes hundreds of customers in common production environments. In a multi-tenant environment, a single solution can be delivered effectively and cost, efficiently. Multi-tenancy provides the ability to host all client accounts in a common environment but provide autonomous environments for each client.

Multi-tenancy only goes as deep as the application tier. Everything above the application tier is open to tenant specific customization. The web server tier, the application server tier, the database interaction tier, and all but a few pluggable pieces of the underlying database schema are all shared storefront, business logic, database interaction, and database schema. All data must be managed in one underlying data model, thus access control becomes very important to protect data from other tenants. The provider is responsible for building the right access control to prevent a tenant from accessing data owned by other tenants. This is done by creating database structures that allow control of access across client or tenant boundaries. Data for each client must all be segregated using ids or counters to keep track of records that belong to different tenants.

5.2. Best Practices for Multi-Tenant Governance

There are a number of aspects to managing the governance of a multi-tenant environment, especially one that is so varied like that of Cloud computing with so many types of services on offer spanning IaaS, PaaS and SaaS that are provided by many different entities to satisfy the requirements of disparate clients. They each may have dissimilar demands for governance and security yet there is an astounding number of solutions available in the market. Yet despite the vast expanse of the Cloud, they all have to operate under some nature of governance to be operable with each other. The primary goals of the Governance initiative are to enable the principles of transparency, user orientation, reliability, efficiency, accountabilities, and rights agenda and spearhead research and explore how to develop tools and services to assist users and agencies in discovering suitable Cloud Computing Services and Providers.

Enterprises need to be careful to define key areas for standards in Cloud Computing Governance as well as developing a community practice in these areas. The Cloud Service Providers need to collaborate and co-operate with industry partners, the Government and Non-Government organizations, and the communities to promote the values as mentioned above. Compliance with the Government cloud computing governance will enable greater collaboration by allowing Developers, Auditors, System Managers to share tooling. Developers and Auditors will want to use tools that are easily interoperable, capable of working with many different cloud services, large and small, on many different cloud platforms. System Managers will want tools that help them balance the goals of compliance and agility as they move applications to the cloud.

5.3. Challenges in Multi-Tenant Environments

Governance of multi-tenant environments is a complicated task which is made difficult by the idea of shared responsibility between the user and the CSP. The alignment of resources is at the root of many difficulties. The demand for the services of many different tenants is placed upon a finite resource pool and the CSP has to balance the requirements so that QOS is maintained. Inevitably there are some tenants who need more than their allocated resource usage allowance who are putting pressure on the model making it difficult to optimize QOS for all. All tenants desire to have the business logic of their applications executed with the lowest possible execution time delay, posing considerable governance problems regarding the specification of resource management actuating parameters for the different tenants.

The external and internal network topology of the CSP can have an impact on how tenants are affected by the behaviors of other tenants due to a user's service request having to pass through the different layers of the topology. With a shared memory environment where resource sharing is at the processing unit level, a large number of design and governance challenges arise. The CSP must be able to guarantee that the processing workloads of each tenant do not conflict, and timeline scheduling of the workloads becomes a governing problem to ensure that all tenants meet their deadlines. Service level guarantees become difficult to enforce because of the shared behavior and because the interconnections between tenants is at a very fine granularity. Definitions of SLA's may not work where some tenants have larger workloads than others; or the resource usage variation is time-dependent, peaking when other tenants are on holiday for example.

6. Data Lineage and Classification

Metadata can be classified and categorized. Each classification requires managing the structuring or categorization of metadata so that it can be used consistently across systems, applications, and databases, especially in large organizations. Data classification, as the name suggests, defines data categories, relevance, usage, business priority, and compliance requirements for a specific use case. Data lineage is a method of tracking information regarding data origin, data life cycle and movement across data storage systems such as databases, data files, data lakes, and data warehouses.

6.1. Importance of Data Lineage

A clear data lineage helps a business understand the flow of the movement of mission-critical data from the source to sinks and the relevance of each stage in terms of data categorization, usage, and priority from a business point of view. Understanding how mission-critical personal data such as social security numbers, credit information, employment information, and others move across the various systems can help meet compliance requirements and comply with reduced retention time frames, on-demand data removal/retrieval, and zero-sum concepts.

Implementing data lineage helps variant use cases such as validating business intelligence reports, conducting data accuracy audits and checks, enabling more effective impact analysis, using compliance impact assessments to determine software application maturity, and indicate cloud migration and application portfolio rationalization.

Data is becoming the first-class citizen of organizations and industry. However, data privacy and protection has become one of the biggest challenges faced by organizations today. With various regulations directing organizations to define, process and protect sensitive data according to strict guidelines, the burden of ensuring continuous compliance rests squarely upon the organizations. Failure to comply with these regulations can result in huge fines and damage the reputation of the organization. Organizations by force should understand the provenance of their sensitive data, the journey taken by the sensitive data at rest, in motion or in use. Data lineage has become more important than ever and is a first step in achieving data governance.

Data lineage refers to the tracking and visualization capabilities detailing the flow of data elements from their source(s) through the various transformations applied to the data at different stages in its lifecycle till it reaches the final output. Some of these transformations help derive crucial insights from data such as predictions about future events or trends; some transformations change the purpose and meaning of the data, enabling it to be used in a different manner such as the transformation that forks data into the data lake and the data warehouse drive operational and business analytics respectively. Some transformations change the policies applied to the sensitive data, while retaining the sensitive and identifiable data in the data warehouse. Some transformations enable detection and elimination of anomalies from the data, either during the processing of data or as an analytical use case.

6.2. Techniques for Data Classification

The process of categorizing data and metadata is called data classification. It can be done manually or automatically. Relying on rules such as file name extensions or file naming conventions is sufficient for a small database. But in large organizations with thousands of data sources, manual data or metadata classification relies on users tagging data for search and discovery, which is prone to human error and inconsistencies. Therefore, an automated approach employing data patterns and AI-powered model training is used widely in database services and products currently in the market.

Classification of data is the primary step for the implementation of data privacy. Many techniques exist for data classification such as Data Discovery Tools, Data Tagging, and Information Flow Control, Knowledge Based Techniques, and Infrastructure-Level Techniques. A Data Discovery Tool relies on pre-defined rules and uses cryptographic techniques. This tool is mostly valid for semi-structured and no-structured data sources, which are widely distributed, but do not guarantee a high level of accuracy. Tagging data is another widely used

technique to classify information. However, the accuracy of the process is directly related to the number of people involved in the tagging process. Keyword alignment can be used to classify metadata and is valid only for a well-defined ontology. The Information Flow Control technique relies on the information flow in an organization and associates a function to each branch of the flow. The classification of data and documents is automatic and can be related to a company's organizational structure.

A possible solution is based on the idea that data classification is an instance of the K-assignment problem. In this proposal, a company's organization structure is an underlying classification model that describes the working environment, regulating the users' access to documents in the organization. Each user is associated with an element in the organizational hierarchy, and each document is associated with a value. The problem is the assignment of the K possible values to the hierarchy leaves, which minimizes the cost associated with the classification.

A system is proposed to perform internal classification. Its structure is a probability distribution and is used to classify the metadata of documents, which are stored in a Digital Library. A hybrid approach for documents stored in a Digital Library is also proposed, which enables dynamic document classification.

6.3. Tools for Data Lineage and Classification

Data lineage can be managed as both a service and as products. As a managed service, it comprises searching for data processes, data flow visualization, integrated data quality monitoring, impact analysis, and prioritization, integrated collaboration, and compliance and reporting. The data lineage managed service employs classifiers, custom processes, log parsers, data pipeline monitors, and flow monitors.

Both data lineage and classification tools help in comprehending the digital object life cycle. It facilitates meeting risk assessment, data privacy, security compliance, and regulatory requirements. However, it's essential to consider the direction of context, and the purpose for which the model is created.

A good lineage tool must have the ability to ingest and collect data from multiple sources across a hybrid environment in order to create a collaborative enterprise view. It should also seamlessly connect to data pipelines, wherever deployed, without any modification or impact to the source system. The entity life cycle model should support ease of use for non-technical business users as well as

advanced features supporting the needs of skilled technical users. These enterprise tools should be able to take inputs from lower-level tools used by developers that have more detailed, pipeline-specific information, add contextual information and generate a collaborative enterprise view model giving required information to a broader group of business users involved in data governance and compliance.

There is a wide range of tools which can be used for data lineage, at different levels of abstraction and different stages in the pipeline. Enterprise-focused enterprise lineage and classification tools:

• Alation

• Informatica

• IBM

• Collibra

• Manta

• Amazon

• Talend

Linux based domain corporation tools:

• Software Artifact Management System

• Collaborative Development Environment

• Open Build Service

• Open VPN

Tools for data classifiers:

• Apache Atlas

• Open Data Discovery

• Google

• Amazon

7. Compliance and Regulatory Considerations

Increasingly rigorous regulations globally impact how organizations must protect their customer, partner, and employee data. Noncompliance increases business risks, including legal, operational, strategic, compliance, financial, reputational, and security. These regulations also directly influence a business's capabilities and strategic decisions across many sectors, markets, and industries, who gather data from their customers for their products or services. Data compliance varies from industry to industry, based on each one's available guidelines or requirements for compliance. It is crucial to implement and maintain data governance capabilities in every organization despite being inefficient for some when working toward data compliance. Organizations must focus on creating data-driven goals to align with the organization's objectives, scope of operations, and customer-type. They include Foundational, Minimal, and Oversight stages.

For large organizations, these stages may run in parallel, meaning different departments may be at stages like Minimal and Oversight simultaneously. The four cornerstones of data governance will significantly help organizations in their quest for a data foundation for the data-driven journey to improving decision quality. The foundation will help speed up your journey and align business and data objectives. However, the success of data compliance within your operation will require top management's responsibility and commitment to overcome the inertia, resistance, reluctance, and fear of also becoming compliant, using data ethics as a guiding focus in your data governance practices. By focusing on your data objectives, you can pursue meaningful value-generating activities by providing those business units, sections, teams, and similar operations guidelines to follow to facilitate their daily activities for data-led decision-making.

7.1. Overview of Data Compliance

Data compliance, or information compliance, refers to adhering to an organization's internal policies or the external regulations placed on the organization that relate to rules governing how data is used, collected, retained, and shared. Compliance rules may include legal, financial, security, and regulatory rules. These not only determine how to protect data and the various aspects of a business for security requirements, but also how to protect the data of the company's clients by ensuring the data being used, stored, and shared is complying as well. Organizations can use data compliance to help promote the appropriate use of company data and aid in reducing the risk of security breaches. Organizations that are compliant adhere to rules such as how well they pay their taxes, how their financials are reported, the data related to their clients, business

regulations relevant to their industry, as well as the data security requirements of applicable regulations.

Data compliance is a difficult task to undertake without the proper software. Adhering to compliance can require a massive amount of work, especially when an organization is a larger one. With a larger organization, the chances of a compliance issue happening increases frequency and cost of penalties for failing a compliance audit can be high, leaving your organization in worse shape than prior to the violation. If the company has the proper software and infrastructure to accurately maintain, manage and analyze data, then there is a lower risk of having a compliance issue. There are many laws enacted to ensure that an organization acts in a manner that is compliant, including various regulations.

7.2. Impact of Regulations on Data Governance

Despite the differing approaches these jurisdictions have in regulating the governance of data, the impact of the key regulations is similar. Regulators are imposing rules on organizations concerning how data is ingested, used, and disposed of. The onus is now on organizations to ensure that data is of high quality and used in a manner that follows the whims of data subjects. Regulators are also passing laws to restrict how long organizations can keep data. With this evolution, organizations globally are rethinking their data governance policies to ensure compliance. Aside from the direct regulatory penalties related to data governance, organizations have to consider reputational issues around poor data governance.

Regulations have increased auditors' awareness of the importance of data governance. Auditors may have noted deficiencies around data governance but because the resources of internal audit functions were limited, they would have prioritized riskier areas over data governance. With data-related complex supervision, data policies may fail the review of external auditors, especially if the IT controls are dictated by other regulations that do not touch on data. The constant reviews that these regulatory bodies require mean that organizations may have to rethink their strategy around good data governance and overhaul their data policies. Whether it is an internal or external auditor, organizations may not pass the audit walkthrough or data analytics phase without satisfying clear concerns on the design and execution of data-related internal controls.

8. Future Trends in Data Governance

Data governance aims at formalizing rules and processes for data management and at ensuring transparency and accountability for those rules [1]. As organizations are getting ready to leverage the value of their data, data governance moves from a support function, led by a small group of data stewards, to an organizational core function, driven at the business level, with the collaboration of several stakeholders and an increasing demand of support from the IT department. Technologies supporting data governance are also changing thanks to the introduction of AI-based solutions. Data governance technologies, such as metadata management solutions and catalogs, data management solutions, lineage and impact data tracking systems, will play a fundamental role in simplifying the definition and the automation of rules and processes. As organizations shift their focus from concern about the risks associated with data to maximizing its value, some fundamental themes will dominate data governance in the future. The first is the number of actors interested in data and the types of interactions they have with data. In some organizations, business units have already taken the lead in defining rules and processes about sharing and using data, and data access and usage are constantly in the news as the major issue for ethical AI. Business units are aware that data governance is an essential ingredient in the trust equation with other units and with friendly external entities, as they will be responsible for data-related events in their functional area. Data stewardship and organization, necessary to define the so-called "digital constitution" of the organization, will pervade all the activities of the data governance office. Results will be achieved in a collaborative way, leveraging on data governance software solutions, capable of supporting the governance of data-related processes. Legal and compliance departments will play a key role in collaborating with business units in defining rules and processes for data sharing and usage.

8.1. Emerging Technologies in Data Governance

Data governance ensures data is handled and accessed properly. It can be difficult on its own but with the introduction of more technology, there needs to be more data and policies. Technology will support defining the needs for policies. The result is a close relationship between the technology and the policies. The technology is far behind where data access control technology needs to be. Policies describe the general needs for data governance such as keeping data in the appropriate systems with controlled access and for limiting the use of data. Technology should implement these policy needs. Currently, data transfer is independent of policy and is allowed among disparate environments regardless

of the use of the data. This is a risk that organizations have had to accept and is why there are issues of regulatory fines and hacked private data. The growth of the data is blowing by the current capabilities of data governance and new technologies for protection and monitoring will help move the status quo of policy and compliance. Namely, as organizations grow and technologies grow, the policies and the technology need to grow together.

New technological capabilities will provide more areas for policy requirements. These new capabilities could include predictive analytics, alerting, and reporting on data policy compliance. Policy exception reporting will become critical to pare down the exception reports which can easily become tens of thousands of pages long. The second tech trend is compliance as a service and will be the next step in filling the automation holes in governance programs. No organization wants to allocate staff to cumbersome manual processes to review data access policies and exceptions. Other technology direction will be like data discovery in figuring out what data is used for what purpose and whether that purpose is still valid. Data sharing is the future of data governance – as organizations continue to share data, there will be a need for automated guidance on how to share that data across lines of business and variability of use and the associated risk.

8.2. Predictions for Data Governance Practices

In the next few years, the traditional data governance capabilities will undergo significant changes. First, data intelligence capabilities will be further democratized, so that non-technical business users get access to better visualizations of data use through solutions leveraging low-code and no-code paradigms. This means that more business users become empowered to define business rules and create data quality guidelines, without having to engage technical data stewards every time. Therefore, companies are likely to spend less on data stewarding activities and enable data stakeholders to contribute more actively. As a natural consequence, collaboration tools will play a more important role in data governance initiatives, especially regarding authoring data policies, standards and guidelines. Next, data governance solutions will invest more in process automation and further embed data governance into the day-to-day activities of data stakeholders, especially data creators and data consumers. For instance, data governance workflows will be integrated into common tools used by stakeholders. Additionally, organizations will increasingly rely on data lineage capabilities to automate aspects of data stewardship work and improve data quality monitoring. These changes already help to onboard new data contributors or those that come back after a while to adhere to data policies and guidelines as well as help organizations deal with the issue of high turnover rates

experienced nowadays. While transparency around data handling is necessary, organizations cannot expect employees to constantly remember dozens of data regulations and internal compliance processes, which are often cumbersome. Last but not least, organizations will increasingly integrate data access management with data discovery, data quality enforcement and data operations, so that data contributors understand the importance of sharing data and are enabled to do it in a compliant way. The recovery of the fell-out adage of data that is a liability is long gone. From our perspective, collaborating with external partners is becoming a prerequisite for surviving and thriving in the current business environment. Making it easy and safe to externally share large datasets will preserve organizations from damaging trade relations with partners, as they're just asking for something.

9. Conclusion

This section concludes the paper with a general assessment. Data governance defines who is responsible for data resources and how they are controlled. Data access and security controls put guidelines in place defining how the data assets can be accessed, retrieved, processed, or altered. It is important that all organization, system, and source data assets be governed to ensure that information is guarded throughout its full life cycle against unintentional or intentional breaches of confidentiality, availability, and integrity. Use and access controls are critical aspects of computer-based data resource management. Security tools and techniques should be deployed in all of the layers of application development, data processing, and data storage activities, from the coding of programs that affect the data to the physical storage devices that hold the data.

Policies, processes, and procedures should be concocted and documented so that all parties related to the organization's information resources understand their roles and responsibilities. After data and information resources are appropriately classified, stored, and backed up, proper security controls can be put into place for their confidentiality, completeness, accuracy, reliability, and timeliness protection. Additionally, specific access, control, and use guidelines for data and information resources should inform and guide technical and non-technical personnel as they utilize these resources. Information security breaches concerning data governance and access control can adversely affect organizational performance and decision-making, end-users, and the public.

They can also result in violations of laws and regulations, as well as loss of stakeholder confidence and added unauthorized access management costs.

References:

[1] S. P. Panda, Relational, NoSQL, and Artificial Intelligence-Integrated Database Architectures: Foundations, Cloud Platforms, and Regulatory-Compliant Systems. Deep Science Publishing, 2025. doi: 10.70593/978-93-7185-129-9.

Deep Science Publishing, 2025
https://doi.org/

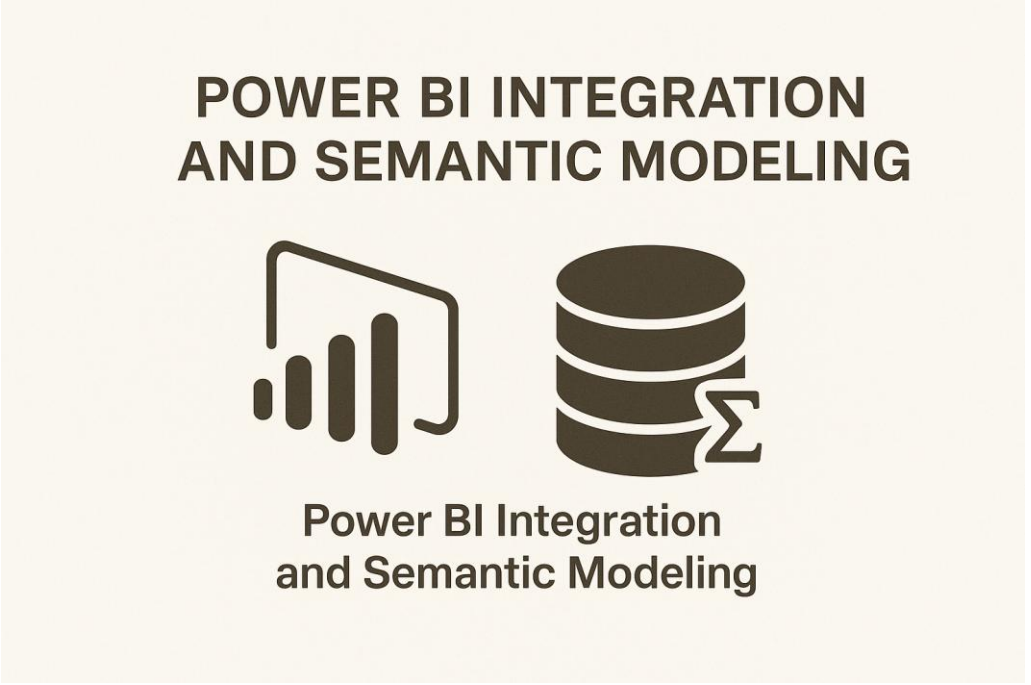
DeepScience
Open Access Books

Chapter 7: Power BI Integration and Semantic Modeling

1. Introduction to Power BI

Power BI is the Microsoft Business Analytics cloud service that provides interactive visualizations and self-service business intelligence capabilities with an interface simple enough for end users to create their own reports and dashboards. It is essential that Power BI Desktop reports be integrated with a Semantic Model designed by a member of the IT staff or a report consultant. This ensures that the data exposed in reports and consumed by business persons follows the business definition of the data. Semantic data modeling also protects the data being accessed via Power BI Desktop and Power BI Service, ensuring that sensitive data is not exposed to unqualified persons.

Power BI consists of several elements that offer unique capabilities, however, the primary focus will be on Power BI Desktop and the Power BI Cloud service. Power BI Desktop is a Windows application that allows users to connect to, transform, and model data. Pre-developed visualizations can be dragged onto a report canvas to produce rich analytics. Dashboards may be created in the Power BI Service by pinning visuals from several different reports onto a single canvas, typically summing up many user exposure choices for specific data. Dashboards help users keep their eyes on the prize, knowing when to dive into specific reports for further analysis. Security, data source management and report sharing are all controlled in the Power BI Cloud service.



2. Understanding Semantic Modeling

What a semantic model does is that it translates the meaning or semantics of data into an understandable format, with a special focus on the data and its nature, structure, usage, definitions, and delivery process, among others. The semantic model also answers questions such as: What is the data about? How was the data obtained? What decision making process does the data deliver insight into? What is the correctness aspect of the data? In what way can the available data be consumed? Who consumes this data and with what frequency? A semantic model empowers the consumer's intellect and allows them to unleash their creativity to ask any questions of the data for their own decision making. It pushes the boundaries of what Business Intelligence has so far been capable of—and with simple tools. Traditionally, Business Intelligence has been a set process defined by experts, with limited and periodic delivery of insight. For many tasks—especially all ad hoc, exploratory or investigatory analytics, the simple delivery of the raw data does not suffice. The consumer may not have the knowledge to come up with the analyses they want. They rely on experts to supply the desired solution. Experts have trouble messaging the complement of Business expertise, Technology and Statistics around the exploratory use of data. It still satisfies a need that the company has, so we work together for this feature.

The solution to both problems is a semantic model. A semantic model allows self-service analytical people to ask as many questions of the data as they want, and go deeper into the analysis at any level of detail—not just the periodic detail and aggregated reports that have been built for them. Providing their Business expertise and intelligence, and with some rules and guidelines, they will quickly compose insightful reports and analyses that satisfy the needs or dim the curiosities of many of their coworkers. The semantic model already makes the raw dataset smart and ready for Business exploration and analytic intuitions. The smart model creates the strength of the need for both Business and Technology collaboration. The learning curve for someone building the model is small.

3. Building Semantic Models in Fabric

3. Building Semantic Models in Fabric 3.1. Overview of Fabric Semantic models have been an integral part of Microsoft commercial products, from SQL Server Analysis Services in 1997 through SQL Server Data Tools, Azure Analysis Services, and Power BI. Additionally, semantic models in the form of OLAP cubes became key value propositions across several major vendor platforms, particularly those used in retail and finance. Semantic models enhance exponentially user experience with their introduction of trustworthy data, consumed as user-friendly entities, attributes, measures, and related tables with multimodel access, best summarized here as an Enterprise Knowledge Model or a business Data Model. Until very recently, however, the only Microsoft offering with banks of enterprise knowledge models were on-premises, with Power BI only becoming the Enterprise tool it is now due to a series of game changes - the introduction of the Data Warehouse within the enterprise cloud, combined with capacity model consolidation, increasing demand for enterprise reporting, and the strategic orientation of Power BI as the ubiquitous business tool, for both external and internal users, generating revenue for all Microsoft clouds.

With the recent announcement and general availability of Semantic Models in Fabric, we now have Semantic Models on the Cloud Data Warehouse, bringing Power BI to parity with the enterprise features of on-premises solutions. Semantic Models leverage the enterprise modeling features of Power BI Desktop for Microsoft Fabric on the Data Warehouse, Windows, and Mac, enabling brands and companies from all verticals to deliver trustworthy data to business users for analysis, sharing, and report authoring with the advantages of a simplified user experience built for ease of use with multi-model access - including natural language queries.

3.1. Overview of Fabric

Microsoft Fabric is an end-to-end analytics platform that unifies Power Query, Dataflows, Data Factories, Data Warehouse, Data Science, Real-Time Analytics, Pipeline, and Data. With Microsoft Fabric, organizations can run their data workflow from ingest, store, clean, and transform to capacity plan, build, train, tune, deploy, score, and monitor their ML models. Building on the foundation of Power BI Desktop and Service, Business Semantic Model (BSM) is a collaborative semantic modeling layer in Microsoft Fabric that unlocks enterprise finance and operational reporting and analytics through ready-to-use business models for all Microsoft Cloud applications, vendor applications, and customer applications for IT and Technical Citizen Users. Additional including are coming. BSM offers easy and fast self-service reporting without needing IT support with out-of-box business models for all Microsoft Cloud applications. With every Microsoft Cloud application having a BSM in the Common Data Model format, enterprise users can search and discover, and use the model out-of-box or customized, collaborating with other teams who reuse or enhance the Power BI model saved in Fabric. With BSM, enterprise users with Power BI can focus on exploring and analyzing data while IT users can spend less time on repetitive model building work. Come join us to learn more about many-to-many relationship, composite model, model metrics, deployment, lifecycle, and CI/CD. When you hear about Power BI embedded in Microsoft Cloud applications, this is BSM. When you hear about the Microsoft Cloud applications publicizing the investment on enterprise analytics, BSM is the answer.

3.2. Key Features of Semantic Models

Semantic models are built on top of a lakehouse storage engine, allowing users to take advantage of the powerful core capabilities — Structured and Semi-Structured data types storage and optimization, transaction and concurrency guarantees, versioning, and governance — while surfacing the data for efficient, semantic-based exploration and reporting. Here is a summary of the main capabilities.

Users of Power BI Desktop will recognize the experience with Semantic Models, as it is almost the same. They will know how to create dataflows already, thanks to their experience with connectors, merging, and editing, with the only difference that in the dataflows, the connectors support only loading data from sources. Data is processed, validated, and written at specific intervals, while the Semantic Models offer a constantly up-to-date view of that data without having to wait for scheduled refreshes.

Power BI is a mature product. It offers a variety of Data connectors to External Data Sources, ranging from file-based sources to database and application connectors. Power BI Data Models use capabilities to extract and transform data from these external data sources and combine them into a data model. This process within Power BI is done through refreshes. A Power BI report will only display new data whenever its refresh occurs. In contrast, Semantic Models are constantly updating their data and data products, being available for instant query or report exploration.

A key use case for Semantic Models is when Line of Business users have their own datasets, either coming from external applications and databases, or processed using SQL and Python. These users would create a Semantic Model pointing at their data products, performing light transformations on them to expose other users to the data in a self-descriptive way, with Business-Focused representations and names.

3.3. Modeling Techniques and Best Practices

When working within a workspace, it is common to create both semantic and non-semantic datasets. First, create a Dataset acting as the Semantic Model, sourced from multiple datasets, even residing in different workspaces. Interested users can create Top Level Reports and Dashboards using the Semantic Model and publish that as shared content accessible to more users. Note that it is not possible to create a semantic model without a source dataset. Whenever changes are applied in the source dataset, the Semantic Model refreshes to capture the changes in the dataset, which are applied in the associated reports and dashboards.

In this guide, we will show you step by step how to create a Semantic Model by utilizing the data modeling capabilities. To do so, we will create a model with Linked Entities using different sources, except for Datasets. Then, we will publish that and create a top-level Report based in Report Builder. The goal is to create a semantic layer that provides business context to Business Users, such as KPIs for their Company, Business Unit, or Market. Then, Business Users will generate Resilient Reports that provide the relevant signals to the Executive Committee or Shareholders. Therefore, it is not the common situation, where Business Users examine the raw data and create their own Reports to explore the Business State. To provide these signals, it is common to link different processes to see the impact of manual or automated actions. And therefore, it is not a star or galaxy schema but rather a complex cube, with measures regarding Revenue, Costs, Employees, Users, Shops and so on. Then, an analysis including all these KPIs provides the overview of how the business is functioning.

4. Governance in Power BI

A logical place to start is to understand what is meant by governance, the answer isn't always as simple as it first appears, there are many facets to governance including who owns the data, who is responsible for its maintenance and also who ensures that there are processes in place to ensure the data is checked for quality, availability and access. Without strong governance processes in place, organizations can quickly end up with multiple versions of the same data in multiple locations. Business questions are being answered off the back of user-created spreadsheets that use stale data or possibly incorrect data assumptions without any challenge. "Data says X" to answer a question is a much harder statement to challenge than "I modeled this with the following steps and assumptions." Building an integration is vital for success with governance, it ensures that users can explore and retrieve additional visual pages from reports built at the enterprise level so that careful thought and planning have gone into the selection of data sources, the measures, and the use of model relationships. Additionally, the enterprise-managed datasets can be refreshed or the data schema modified or additional relationships added that further enrich user discovery over time.

While traditional business intelligence solutions often relied on a top-down approach where the business intelligence team would dictate what datasets were provisioned and held responsibility for the creation of dashboards and visualizations, a more flexible bottom-up approach is enabled. Data is published to the service from various sources across the organization, and groups of users will create reports that are useful to them and that they may want to publish to their peers. The ability to leverage existing datasets increases the speed with which users can create analytical reports, as they no longer have to invest time in acquiring or transforming the data into the required format. Users can also perform their own self-service discovery, though of course these insights must be validated and queries back to the business taxonomy and logic defined in order for business users to gain any confidence in the accuracy of the conclusions.

4.1. Importance of Data Governance

In an organization, data is used for both transaction processing and decision analytics. For these data to be usable for decision analytics, they need to be governed for comprehensibility and quality. This is where data governance comes into play. Most organizations do not govern their operational data because they are treated as "a by-product of everyday business". Data collected during the course of a business transaction are often poorly defined or of dubious quality.

In addition, the business rules associated with transaction processing are buried in the application software that processes those transactions and are often unknown to the users of the data. As a result, however, these operational data are used for analytic purposes both formally and informally. In addition, these operational data used for decision analytics are not published in a data catalog.

But this is old world thinking. In fact, it is now widely recognized that the analytics data, often containing high value for data monetization projects, should be made available to the entire organization in a usable and governed fashion. These data should be discoverable, understandable, and of high quality. Organizations should therefore invest in appropriate tools for enabling data governance. There are many tools available for data governance but these have to be products for the specific data and analytics ecosystem of an organization.

One of the goals of this text is to highlight the key features of the data that should be made available to decisionmakers but without sacrificing usability and performance. Furthermore, the analytics data are most often stored in NoSQL or warehouse optimized file systems and hence are often not governed at the metadata level although both are equally necessary for high data quality and performance.

4.2. Implementing Governance Frameworks

A solid data governance framework is necessary to avoid having abandoned data models which then result in the untrustworthiness and unreliability of the solution as a whole. Start by clearly defining the responsibilities of specific roles and the main objectives of the overall governance workgroup. Data source connection, dataset design and deployment, report development, infrastructure management, and data model change management may be tasks of either a central team or of collaborative functional areas. The critical aspect is that those persons responsible for specific tasks be defined, and that the proper guarantees be implemented to support the correct execution of the governance workgroup objectives.

We recommend implementing all steps in light of the Five Pillars of Business Intelligence. This approach takes into account all areas impacted by the introduction of a BI solution within the organization: Ethics, Technology, Policy, People, and Process. A clear Ethical rationale will support the BI initiative throughout the design and operational phases. Specific Technology components focus on the BI practice expression. BI practice Policy encompasses the principles and rules supporting the way each of the Five Pillars is specialized. People define competencies and engagement methods to ensure the initiative's

success. Business process orientation will help keep the initiative tailored to the organization's needs, while a data-driven approach will ensure that the achieved result is aligned with the intended goals.

4.3. Role of Security in Governance

Data security is often thought of as a safeguard activity, but it has a pivotal role in the overall design of an organization's data architecture. Security is the main enabler of the access policy for data modeling and usage that is an essential part of the governance design for data solutions. A good security model allows organizations added flexibility in how they provide functionality, while at the same time ensuring that business units have a say in how their data is modeled and integrated. It also provides an effective way to ensure protection for data that has stricter access requirements. A well-designed data security model defines mechanisms and processes that data solution developers and data access users can take advantage of when using data tools.

Security provides a set of effectively enforced rules that cannot be overridden as part of the data interaction process in reports and dashboards. Using row-level security within datasets is the first layer of this policy model. It is the primary, built-in mechanism that data tools provide to filter data based on individual users' access rights when they run reports and dashboards. It is configured as an embedded element of datasets and works for all report viewers, regardless of how and from where they access them. Row-level security is triggered at the time of the report execution to ensure that the saved results reflect only those rows of data that the viewer is authorized to see. Any row-level filters or rules must be assigned to each dataset by a user with modeling rights specific to that dataset. In data tools, row-level security must be set up by model designers either through the application or from the datasets view in the service.

5. Performance Optimization

This chapter discusses performance optimization techniques for Power BI reports integrated with Excel workbooks. The goal is to deliver an optimal experience; for Excel data consumers, report developers should utilize performance metrics available on the Service, along with organizations' best practices related to optimal report design and deployment. For those new to Power BI report development, Power BI reports are inherently designed to visualize additional data facts (often time-series data) and the goal of optimization is to ensure minimal lag to initial report refresh; to report page switches based on filters,

slices, buttons; when hovering to see tooltip data details; to cross-highlighting elements in the view, and when drilling down further into hierarchies by enlarging data granularity. Further, for any recommended capabilities, adjustments, or settings, organizations are encouraged to validate within their environments to ensure improvement in report development and data refresh performance.

5.1. Understanding Performance Metrics

Power BI provides Service-based metrics that include refresh metrics and usage metrics for detailed performance tracking and monitoring. The Refresh Metrics deliver details on how long it takes to refresh reports and datasets in the PBI Service using the Power BI Premium Capacity. The usage metrics indicate how often reports are opened and the sections of the report are viewed, captured in the amount of activity. These metrics can be accessed from the Service workspace for a report and its dataset. Additionally, there are tools available to help corporations customize additional internal organizational metrics.

5.1. Understanding Performance Metrics

When building a Power BI solution, performance is one of the most important facets to consider. Users like to see things load quickly, so a slow-loading report can result in the user experience being ruined, even if the report is amazing and has insightful data. While issues with refreshing the model happen far less frequently than slow reports, refreshes can be scheduled for particular times, so they run in the background, and the user won't be bothered if things take longer than expected. But like everything else users see, report load time should be as short as possible. Both report and model performance should be monitored closely during the development of a Power BI solution. To build a high-performing solution, you should understand the common metrics influencing performance. Data latency generally refers to differences between when a user action is triggered and when that action results in a visible change on the device. Report latency refers specifically to when users interact with a report. Chain latency is the time taken between when a user performs an action and the time taken for the back end to process the action. Query time is the amount of time it takes for the data source to process the query being sent through direct query or incremental refresh. Data transfer time is how long it takes to move data from the data source to the user's device during a query request. And visual play time is the time it takes for the visuals on the report to process and render on the user's device once the data has been transferred.

5.2. Techniques for Optimizing Performance

When we are modeling data in Power BI, we need to identify different ways to optimize performance. Several properties in the model directly or indirectly affect efficiency and have to be set accordingly. We will look into different components that businesses often use to optimize performance and give techniques on how to use them correctly to achieve better speed and efficiency.

Model Design

The simplest way to make our model efficient is to have an appropriate dimensional model setup in the data model. Even small things like removing any unwanted tables or columns, combining the date tables to a single one, appropriately using surrogate keys, and partitioning the tables for large datasets can give drastic changes in performance. For any large tables, we must avoid unnecessary data loading. Consider further partitions to load only the necessary data needed for turbines instead of loading all years' data during initial loads. Also, avoid direct input flow into Large Tables. Work on performance tips related to direct query mode if you are using a DirectQuery connection to the data model.

Cardinality

Cardinality is the uniqueness of values in a column. For instance, a column containing IDs has a high cardinality. A column containing a few values repeating, like the Tax Exemption Flag, would have a lower cardinality. Having key or base attributes with high cardinality may create a problem. The dimensional model uses dimensions less than the fact model. Let's optimize something before it arrives at two to three million rows/query to Power BI. It sends four queries to Power BI for each scenario.

5.3. Monitoring and Troubleshooting Performance Issues

The combination of a hybrid cloud and semantic models running in memory, and also directly in source data for some cloud or enterprise scenarios, opens up a wide range of performance scenarios. You need to ensure that all Power BI resources are provisioned and used correctly to prevent extended performance issues for reports and dashboards. Additionally, IT and developer teams will want to check on the performance of resource usage and report execution. Power BI offers a selection of capabilities and insights, as well as integration with the Microsoft Azure Resource Manager to control and monitor the resources used by Power BI and its reports and dashboards.

The Monitoring and Troubleshooting Performance Issues is an extensive feature set that allows you to visualize and explore how Power BI is performing report

data requests. This includes the use of resources in Azure that are assisting with the execution of Power BI reports, and other monitoring capabilities that will help you track down specific reports that might be utilizing too many resources, or simply executing inefficiently and offer a focus for tuning. Integrating Azure Monitor with Power BI gives you visibility into Azure resources utilized by the Power BI service and any issues preventing users from accessing the service.

Data Access Monitoring provides an overview and identification of Power BI reports that are executing queries against your data sources and might take a long time to complete. In addition, the Power BI service can track and present information about which of your reports are being used or reference a specific data model, and flags them in terms of speed or errors during the query execution. By including those reports in your monitoring, you are able to take targeted optimization actions on multiple reports using the same data model or data source.

6. End-to-End Enterprise Reporting

Multi-channel reporting is one of the fastest and most popular applications for leveraging a semantic model in an organization. Finance has functioned in a "report or die" mode for decades. Now adjacent departments are adopting a similar "no report, no insight" approach. Improvements in desktop and enterprise report writer technology have made this a reality. The enterprise semantic model serves as a bridge between the publishing of industry or best practice-based financial structures and key metrics, and the ability to evaluate the deltas. Reports are moving out of the dark recesses of the Finance department and being published in a manner where consumers use them without having to go through Finance for every minor change.

6.1. Designing Effective Reports

Analytics is a recommended area for new implementations. It serves as a central data warehouse for transactional systems in an organization and is a good candidate for a Power BI integrated with an Analytics system. Power BI is also a business solution tool that is there to help decision making at all organization levels. Insights from your data should be visible to your decision makers. This section will help you understand how to best expose those insights through effective report design for your audience that includes executives, analysts, operation managers, and operators. Power BI provides different products that cater to different audiences with different use cases for each product. It is

important to understand those products – Power BI Service, Power BI Desktop, Power BI Mobile, and Power BI Report Builder – and their different capabilities to better expose your company data through an effective and interactive reporting process. The Power BI service has a web interface that is easily accessible. Also, user setup in the Power BI Service gives each user different roles and permissions that restrict the functionality they can access. User groups allow different people from the business to collaborate and curate the workspaces that contain the reports for their sector. Power BI Desktop is the desktop application that is used by report creators. This application was designed to create business reports focused on business functionality as opposed to being designed for data engineering.

6.2. Integrating Data Sources

Enterprise reports are generally compiled from more than one data source. As companies become larger, with business divisions using different software solutions to support their business processes, the systems architecture tends to become fragmented. Each division might then end up with its own data source, which explains why enterprise reports usually use several sources to compile results. Companies are also more concerned about their investment in systems architecture. Since they spent a lot of money on transactional systems, they are reluctant to dismantle transactional systems in order to implement solutions dedicated only to business reporting. For these reasons, including external data for reporting is seldom a simple task. The solutions currently offered by software developers for enterprise reporting are not automatic solutions.

Data integration refers to the process of compiling data from several heterogeneous data sources and transforming it into a unified structure [1-2]. The potential data sources include enterprise transactional systems, customer relationship management data, data warehouses, enterprise resource planning systems, financial systems, external market data, and spreadsheets. In data integration, it refers to the process of extracting the required data from each source in order to join it and then transform it into the format needed for reports. Data reporting is the process of retrieving the compiled data previously structured by the integration process and then visualizing it. Reporting developers have developed sophisticated tools, with friendly user interfaces, that automate the data integration and reporting processes for transactional online analytical processing systems used for enterprise reporting.

6.3. Automating Reporting Processes

Every time the explorer creates a new visualization one additional line of code is sent to the Analytics Platform. This is possible because each visualization in the report is not a self-containing image. With that logic a report is very similar to an table and each one of the visualizations is similar to a cell. A cell and a report are constantly exchanging messages so that the explorer gets a unique interactive experience even if he didn't explicitly tell the report to change its content. When the explorer filters data but he is not satisfied with what he gets, he can trigger a new search by writing a new keyword. Such interaction contains two conversations: one at the cell level that just gathers the data that will be used for the visualization; and another at the report level that asks for a new interactive search with the new keyword value.

When a new search is triggered or when he interacts with a new cell, this last cell first retrieves the data to create the new visualization and sends the content to the report. If the explorer clicks in a favorite in such situation, the subsequent actions will be very responsive. If the explorer applies a filter but no data is retrieved, he will see the empty visualization until the new search triggers again. But in an environment that filters available data out you don't expect to see that kind of interaction. In a nutshell, when the user taps the search button he becomes the sole explorer. The Analytics Platform, as part of its functionality, is capable of automating this interaction, no matter how unstructured, by running the report as scheduled batches. When doing this automated reporting, the search bar action will use only the stored actions. The actions are done at the keyword values needed to navigate to reach the destination with less cost.

7. Case Studies and Applications

Power BI is increasingly being used for reports and dashboards for many sectors. These Power BI reports are connected to a wide variety of data sources, many of them supported by a built-in connector. This data is typically in the native cloud, queried using Azure services or Application Programming Interface calls, or extracted and stored in an on-premise data warehouse.

The data sources are typically Azure, SQL Server, Teradata, SAP HANA, Oracle Eloqua. Other sources are also extensively used, including connected Excel files, Azure Analysis Services, SalesForce, Microsoft Dynamics, SharePoint lists, Google Analytics, and Cloud-based sources. The datasets are published to the

Power BI service. Licensing varies depending on the organization but can be based on paid subscriptions or on free while using the self-service analytics route.

Governance, Policy and Security compliance are a few key controls organizations need to keep in mind while using these tools. These factors often inhibit adoption especially in traditional organizations. How can organizations unlock the value of these tools? Should they use governance-centric, corporate-driven approaches or trust/team-centric citizen-led approaches, especially by the Business Analytics? Should they delay adoption till the tools are mature enough and controls built-in? In this chapter, we discuss these points and present lessons learned and suggest a roadmap to building an organization that continuously transforms into a Data-Driven Organization.

7.1. Industry-Specific Implementations

Power BI has built-in connectors for scores of popular services, and Power BI Premium services expose its interfaces to third-party services, allowing these services to push data sets into Power BI. All Microsoft-cloud-hosted services are likely to be Power BI partners at some point. These existing connectors and the ease of developing new ones make it fairly trivial to integrate Power BI into second party SaaS apps. Partner or appendix implementations, however, are just the tip of the iceberg, since BI vendors have been integrating BI into specific domains for over a decade; whatever domain is of interest, there likely exist enterprise applications focusing on the specifics of such a domain, and these enterprise applications have been integrated for that domain with existing analyzers for a while. These specific integrations create a very different reporting and analytic experience.

Here, we discuss a set of case study implementations, each of which is domain-specific; minimally, a case study describes the components developed and the integration. In addition to these case studies, we discuss why domain-specific integrations make even more sense today than when early analyst hope saw little of such implementations. Finally, we briefly consider some of the barriers to these implementations. The question of industry-specific implementations, while key to functional value, is usually neglected in generic BI product discussions indeed, unless explicitly stated otherwise, readers should assume partner implementations for BI products. However, even if generic capabilities are incorporated by Power BI developers, we believe that all such capabilities will be ineffectively generic.

7.2. Lessons Learned from Deployments

This section provides a compilation of lessons learned from several Power BI deployments. They have been selected to emphasize specific areas of Power BI Semantic Model implementations, including technical infrastructure setup, usability concerns, and data governance considerations.

The provided lessons learned apply to implementations that utilize the Power BI Premium service. They have been drawn from the experience with Power BI Premium deployments, some of which also used Power BI Report Server, and other deployments that used only Power BI Premium. Neither an on-premises solution nor Power BI Desktop was affected by these lessons. The implementations these lessons learned draw from are Organisation A, a large public sector organization, Organisation B a large private sector company, and Organisation C, a small private sector company.

Storytelling is essential for report usability. It guides the user in their navigation through the report in the context of the insights that are highlighted. While understandable technical terms are useful, make sure your users understand the industry-specific story in its entirety. Here is just one example: "What is driving costs in this area of the business?" is more guiding than the question "Why is this service centre in the red this month?". Add storytelling elements that enhance report usability, such as custom layouting, consistent color usage, intuitive filtering options, tooltips, bookmarks, and dynamic titles and labels. While it takes time to create a storytelling environment, this time is well spent.

8. Future Trends in Power BI and Semantic Modeling

The semantic data model of Power BI has constantly evolved under the impulse of the community and Microsoft constantly invests in improving the functions and integration capabilities of Power BI and in making it more and more accessible both to the large public of non-technical users and to IT professionals. The historical trajectory showed us how it brings business intelligence closer to the various types of users who need it in the business: from the developer who creates the infrastructure and the advanced semantic model that supports it to the report consumer who can easily find the insights he needs and create ad hoc analyses. In this section, we will examine the trends that we think will most strongly characterize the future of Power BI and semantic modeling. The

business intelligence sector is extremely broad and complex. Within it, there are several interconnected areas such as data engineering, reporting, embedding analytics in applications, self-service and augmented analytics with the application of artificial intelligence, in particular machine learning, on the organization data, data science, enterprise modeling, workflow automation, enterprise performance management, and so on. Each of these areas sees immeasurable investments and innovations thanks to the application of new technologies such as cloud computing, the Internet of Things, blockchain, machine learning, and augmented reality. These innovations, in turn, strongly influence how the entire business intelligence sector evolves, how operations and processes change, and how new job roles and professional figures are defined within organizations. But the most incisive impact is caused by AI and machine learning. There is no sector that does not see the application of AI, and business intelligence is no exception.

8.1. Emerging Technologies

Over the past several years, a rolling wave of new technological capabilities has been dramatically changing the way businesses deliver value and generate new business opportunities. The enabling forces of these technologies result in new and enhanced customer and employee experiences, greater insights from connected products, services, and systems, and improved operational efficiencies. Technology innovations such as machine learning, distributed ledgers, artificial intelligence, augmented and virtual realities, natural language processing, and the Internet of Things are being combined to create an ever-increasing set of new capabilities and opportunities.

A major player in emerging technology capabilities is also an early adopter with various services. The Power Platform combines these capabilities and makes them available to everyone in an easy and consumable way. In addition, many of these new technologies will also become embedded in the tools and technologies enterprise developers and citizen developers are already using. As a result, all organizations – every business and every IT team – will be able to embrace these transformational capabilities. The big commitment to democratizing technology is sure to keep this vision alive into the future; constructive trust and a common platform allowing collaboration across organizations will work, as trust and shared reference designs did in more traditional models of technology. The global, distributed, partner-led development model of the past will be accelerated.

8.2. Impact of AI and Machine Learning

The introduction of tools and technologies to automate tasks has been beneficial to society, but at the same time has made work unnecessary for some people and has emphasized the importance of the ability to combine, understand, and challenge the information created by these tools. In the area of semantic modeling, this combination, challenge, and understanding work is being enhanced by the use of artificial intelligence, ML, and large language models. The innovation behind the advancements on AI during the boom was to combine a general-purpose, always learning language model with a pre-specified approach to feeding new data about a specific client or company. The success of these new AI engines prompted the tech industry to propose the research and implementation of more applications combining general models with domain-specific programs. This two-part approach allows AI to use knowledge, patterns, and vocabulary acquired in many previous projects and apply them efficiently to new specific situations. The specific demand led to the AI boom, allowing companies to practice a simple and powerful strategy: building tools to create and enhance the results of specific domain applications. This is an approach also used in many machine learning models and semantics tasks.

Power BI has an LLM-based semantic model beta in Power BI that automatically generates insights, recommends visualizations, and enables augmented analytics capabilities. Power BI and Azure open LLMs enable enterprise data teams to use the massive capabilities of AI in enterprise applications. As more applications take advantage of LLMs to perform work functions, semantic modeling will become a key technology to create good data environments for companies, increasing the quality of semantic information and business layer functions in BI tools. Data will speak a language accessible to most people, while business intelligence tools will enable the use of structured and high-quality data to create dashboards and interpret information easily.

9. Conclusion

Semantic models enable robust automation in the analysis of business data, and several modern tools empower the democratization of semantic modeling even for data consumers. In this chapter, we demonstrated how to build enterprise-class semantic models and how to leverage them using Power BI. We introduced you to the semantic modeling platform, with the Business Intelligence Semantic Model at its center, and we also covered other tools involved, including Analysis Services.

We guided you through the whole semantic modeling lifecycle, including how to build enterprise-class models and integrate them into business intelligence solutions. You learned how to publish models and consume them in real-life Power BI reports and dashboards, from both the cloud and on-premises. You became familiar with best practices from real-world deployments. We showed you how to control the usability and layout of models. We demonstrated how you can create and publish calculated columns, measures, and hierarchies for your data consumers. We also discussed advanced scenarios when dealing with large models or with additional model layers such as the transportation model and the presentation model.

Semantic models are a critical part of enterprise BI solutions. Because they centralize business logic, they make it easy to have consistent metrics across users and reports. When integrated into Power BI, semantic models considerably amplify the reporting and visual experience provided by Power BI for developing business dashboards and reports based on verified metrics. They delegate enterprise-scale data processing to a dedicated engine. Thus, Power BI becomes a “thin” client that leverages semantic models hosted in the cloud or in on-premises servers like SQL Server Analysis Services or Analysis Services.

References:

[1] Kempson, Ruth M. Semantic theory. Cambridge University Press, 1977.
[2] Fox, Danny. *Economy and semantic interpretation*. Vol. 35. MIT press, 2000.

Deep Science Publishing, 2025
https://doi.org/

DeepScience
Open Access Books

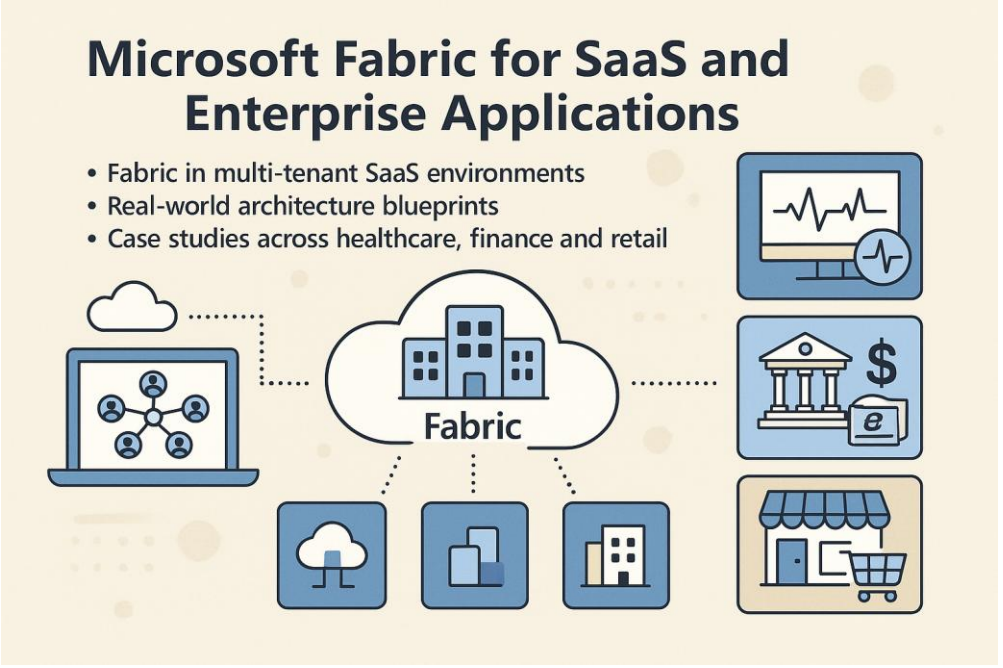
Chapter 8: Microsoft Fabric for SaaS and Enterprise Applications

1. Introduction to Microsoft Fabric

Microsoft Fabric is the next generation cloud-based unified analytics offering platform with one-stop solution for analytics workloads especially targeted for SaaS and enterprise applications. All potential data, be it SaaS or internal enterprise data, generates demand for analytics, be it aggregated batch analytics, business reporting by business analysts using Power BI, data preparation which is light ETL, Modern data warehouses, or transactional business event streaming or analytics. The automation powered by integrated Data Engineering and Data Science aims to reduce time to analytics insights and cost of IT and data science teams while enabling more rapid experimentation. Microsoft Fabric, by offering all of these capabilities in one unified platform, enables users really become self-service, analytics building business experts.

Microsoft Fabric consists of a number of underlying intelligent service capabilities at its core. A combination of these analytical capabilities are packaged into several Fabric consumer experiences optimized and tailored to users of different roles. Data Engineers can make use of the Data Engineering experience or Power BI's integrated Data Flow experience for light ETL operations. Data Science engineers can use the Notebook and Machine Learning capabilities. Business experts can either use Power BI to create a report for data visualization or use Data Warehousing tools or real-time analytics feature to set up more advanced reporting or real-time dashboards. The Business Analytics capabilities are powered by Data Warehousing. The Data Warehousing capabilities at Fabric's core provides a low-code experience for business analysts or experts to query batch analytics stored on the Data Warehouse, build business reports on top of the Data Warehouse and consume the reports on Power BI

Portals. The Batch Data Analytics capabilities are powered by the Data Warehouse capabilities at Fabric's core. Power BI's Real-time Streaming Analytics are powered by Fabric Integration.



2. Understanding Multi-Tenant SaaS Environments

Built using cloud application hosting platform services, a Software as a Service (SaaS) service is typically built once and used many times. SaaS offerings from application companies may provide either horizontal, domain layer capabilities for service integrity, application security and data source management, or vertical layer solutions for multi-domain service orchestration and tenant lifecycle management. For platforms, SaaS services are best suited for their business functions requiring constantly changing capabilities for time-sensitive, event-based workflow automation and specialized user interactions. Similar to earlier application hosting environments, multi-tenant SaaS environments efficiently use physical infrastructure resources to host application services for a large number of customers. Different from physical infrastructure hosting, multi-tenant SaaS environments have two key differences driven by business requirements. Multiple domains of functionality are grouped to form the business core for

similar industry verticals, such as Financial Services or Health Services. Multiple APIs expose the business core as a rapidly changing set of capabilities for use by internal and external user interact functions. It is how well the software-defined exposed APIs are designed, deployed and run that determine how successful the business core will be. These APIs must provide strict runtime service-level agreements to run successively over days. Multi-domain API development, deployment and management require specific considerations. Different domains, like accounts management, transaction processing or customer management, will have their own specialized capabilities and run-time operational requirements, such as functional isolation, capacity elasticity, data integrity, performance security, transaction buildups, congestion handling, data security, audit trail and recovery point. Within each domain, the actual implementation of services targeted for different tenants may follow a shared resource approach where a core capability is used by all tenants or a specialized resource approach where there are separate resources, either physical or software-defined, for each tenant.

3. Architecture Principles of Microsoft Fabric

Microsoft Fabric for SaaS and Enterprise Applications provides a broad perspective about building near infinite-scale distributed data platform, targeting both low-latency and cost efficiency use cases. A set of architectures principles and guidelines supported by design patterns are used to make design decisions for building robust data solutions. Microsoft Fabric is powered by technology with thirty years of R&D investment in building data solution products of high scale and reliability. The security and access control built in across all services help people focusing on building their solutions.

In this section, we share the architecture principles and guidelines, and design pattern for scalability adopted in use cases implemented for real customers. Following sections discuss in details design and implementation for use cases such as global scale user data storage, machine learning, business intelligence, storage engine for enterprise SaaS applications, petabyte scale data warehouse and fabric explorer that validate the design principles and provide the concrete implementation of services in Microsoft Fabric. The principles and the building blocks described are applicable not only in the solution for enterprise SaaS applications and other customer use cases supported by Microsoft Fabric but also to other near infinite scale data solutions. However, other solutions may take different approaches and may not support the same set of market scenarios or

have the same availability, reliability, security and supportability characteristics as Microsoft Fabric.

3.1. Key Architectural Components

Microsoft Fabric is a cloud platform that accelerates the development and operation of Software as a Service (SaaS) and Enterprise applications. The Microsoft Fabric platform provides enterprise-grade modular components and associated services, which create a well-architected full-stack SaaS platform to build, operate, and support application needs to enable real business continuity, resilience, performance, security, compliance, and operational excellence. In this section, we present key components of Microsoft Fabric including natural interactions, modularity, full observability, account and subscription management, security and compliance, reliability, performance, and cost.

Natural Interactions: Digital transformation and the emergence of multigenerational workforces are driving the need for natural digital interactions with people, businesses, and things. Human-like dialogs, powered by AI and Data + AI, make learning new lines of business and handling tasks easier and more efficient. With natural language and thermal mixed reality as productivity tools, Microsoft Fabric enables natural interactions as part of the full-stack SaaS solution, designed to help organizations be ready for the landscape of the future.

Modularity: An organization might have a need to integrate rare data, unique security guarantees, legal requirements, and specialized pipeline workloads that require private and dedicated resources owned by the organization. With modular Services and diverse blades, Fabric allows organizations to customize solutions with unique performance, security, or pipeline design workloads while managing these solutions from a single project. A low-code and code-first approach allows organizations to develop unique scenarios as private, dedicated services exposed as low-code templates that integrate back to the Fabric core ecosystem.

3.2. Design Patterns for Scalability

Scalability architecture patterns help SaaS systems grow to handle the increasing loads driven by success, take complex workloads off expensive shared servers to specialized services running on cheaper compute servers, and isolate user activities when needed. The high-level architecture of the SaaS platform integrates these scalability patterns, enabling enterprise SaaS applications to ride through demand spikes with their workloads offloaded from shared servers to background jobs, user queues processing jobs executing for each user distributed across service provider and resource consumers, and user workflows following the same pattern.

Complex business need not run on expensive platforms all the time to satisfy random spikes of demand. Background jobs can run more cost-effectively in Pipelines. Business processes driven by scheduled events, data changes, or inbound messages can trigger job execution to take important but complex workloads run by expensive shared servers off those servers. The ability to create custom activities in a resource tenant allows an app to create a job task to spin up servers with a cost model optimized for those special activities when they are needed. These activities can take transactional workloads off the shared transactional stores to a staging store, where Data Warehousing will consolidate and summarize those activities in the Data Warehouse. Transactional workloads can also be taken off the shared transactional stores to a SQL Database, using the management capabilities of SQL Database to manage the direct server kind of cost structure. What you want to avoid is for the shared servers to experience long lock wait times on the transactional databases providing the service to the users.

4. Real-World Architecture Blueprints

Other than the major features which comprise Microsoft Fabric that we discussed in the previous section; it is always good to explore specific solutions, so that you can get a real sense of how these different components come together to solve a specific problem. In the following sections, we examine three such customer driven solutions built with Microsoft Fabric, with an example from Healthcare, Financial Services and Retail. Each of the sections will discuss the specific requirements through the lens of the specific industry, elaborate on the various micro services and components within Microsoft Fabric that are utilized in the solution and then present an architectural diagram detailing the various components. The goal is to give you insights into how to compose Collections, Pipelines, Dataflows and Datasets within Microsoft Fabric, work with Lakehouses, Workspaces and Fabric Applications using the Lakehouse paradigm and thereby bring together the various, rich elements of the Microsoft Fabric offering to build a solution that meets your organization and customers' needs. Before we talk about the architecture blueprints, let us first look at the specific problem statements for the three chosen industries. First, to provide a complete and accurate understanding of a patient's journey, it is necessary to integrate data sets from multiple, varied data sources, both internal and external to the organization

4.1. Blueprint for Healthcare Applications

Healthcare is an information-driven industry that works continuously towards creating, managing and improving the quality of life for humans. It is both personal and enterprise, where an individual accesses healthcare information for himself as well as for his family. Healthcare Enterprise Applications (HEA) consist of complex groupings of tasks that require high volumes of information creation, dissemination and psychotherapy in a complicated organization. Due to such nature of HEAs it is a blurring of the line between two separate disciplines of SaaS and HEA development. Microsoft Fabric is a platform with a unique capability of allowing for the assembly and interoperation of specialized parts to build the composite applications that enterprises such as Clinics, Diagnostic Centers, Tele-health Centers, Dispensaries, Pharmaceutical, Medical Devices, Bio-Medical Tech, Health Assurance and Insurance, and Wellness Use. HEAs are technology-agnostic and are primarily based on user experiences and interfaces of medical professionals, patients, and stakeholders. Their success criteria are enhanced patient experience, increased employee collaboration, availability of better dataset, and augmentation of deeper analytic insight into tradition costing while rendering care, increasing quality, reliability and outcome of care activities while cutting cost and using lower resources.

The prototype is aided by ingestion patterns, MLOps for programming sophisticated AI-based services and solutions, and several analogs in the Azure ecosystem. Some enterprise-level Healthcare Artificial Intelligence Technologies use exemplary data, datasets, and overall structure and architecture of Microsoft Azure Cloud Services, though, they fare to address the patient use personalized scenarios. Prototype has several flagship HEAs for Tele-Medicine, Patient Management, Diagnostic Management, Clinical Research, and Hospitalize Care Home Health Services. With areas of patient satisfaction surveys with sentiments and analytics dashboards with actionable insights.

4.2. Blueprint for Financial Services

The financial services industry has increased its strategic use of technology to transform the way it does business; technology is no longer seen as a cost center, but as a key to growth and differentiation. The movement has recognized this, and traditional financial services institutions are spending significant resources not just on securing technology infrastructure, but also on driving business innovation by reshaping the customer experience and rearchitecting intelligent products and information services. Financial services had been major early adopters of centralized systems, innovating with online banking in the early 90s. However, with the rise of digital mobile wallets, banking-as-a-service, and open

banking ecosystems, newly emerging financial services players are short-circuiting the traditional value chain.

The movement has sparked renewed interest from traditional banks in building out microservices-based digital experiences and embedding data-driven differentiated capabilities meaningful for customers and partners. The end-to-end value chain is typically composed of product functions (loans, deposits, investments), services/operations functions (customer/onboarding, credit, fulfillment, servicing), front-office channels (branch, digital, contact center), and support functions (risk, finance, compliance, technology, HR, marketing, logistics). Each of the functions can be underapplied, with heavy-cost elements from overhead nonproductive areas, or overapplied, with organizational blind spots for handling the emerging questions of financial crime risk and cross-product customer servicing and support.

4.3. Blueprint for Retail Solutions

Retailers today increasingly face diverse demands coming from multiple sources at a near real-time pace. The physical retail space is now historically being shaped in new ways driven by decisions made by customers behind their screens. The Digital Twins Technology provides retailers with an opportunity to build a complete digital twin for their business. With this complete view, retailers can develop new analytics to report past events, but also to predict likely future events on which retailers can react. Important decisions in areas such as supply chain, offers & pricing, inventory, cross-selling & upselling, customer-driven purchasing patterns, sales forecast driven resource management, are only a fraction of the areas where a Digital Twin can be used.

By investing in these Digital Twin Analytics, retailers have the opportunity to move from event-based to scenario-based ROI. Instead of mainly reacting to past events through descriptive analytics, they can explore different "what if?" simulations. By leveraging the full set of services, these what-if simulations become a powerful tool to drive better ROI on major decisions. Customers expect services like click-and-collect, where items ordered online are gathered and saved for customers at the local store. They give space to a service model where potentially thousands of items are ordered in a single purchase order and sent from centralized warehouses. Knowing about all the business challenges allows the team to understand your needs better. It also allows all partners and customers to discuss best practices, use cases and design options that will benefit the whole ecosystem.

5. Case Studies in Healthcare

Technology has become an increasingly pivotal element in the growth and transformation of the healthcare industry and different organizations now work towards creating high performing, economically viable and efficient healthcare systems using technological advancement. Leading these advancements are industry giants promoting better patient care experiences with cloud solutions. These technology solutions are at the forefront of managing all operations ranging from seamless adaptation of telehealth solutions to the digitization of operations like patient registration, laboratory reports, bill processing, and patient data collection, with ease and effectiveness. This chapter takes a look at two such case studies that exposed the strengths of a cloud fabric and the role it plays in healthcare.

5.1. Case Study 1: Patient Management System

A patient profile management system is developed to partially automate patient visit management activities. The core of the proposed system is focused on managing patient visit records that cover patient registration issues, organizing patient visit history, assigning appointments, updating specific patient visit data, receiving data related to specific patients, and displaying patient visit data. The proposed system accepts and manages critical patient visit details with modules that enable satisfactory patient experience.

Patient health record management has become increasingly popular to improve care quality, reduce unnecessary medical costs, support quick examinations or diagnoses, and minimize long waiting times for patients. Every healthcare provider's goal is to provide accurate, timely, and better quality healthcare services to their patients. However, this requires a precise understanding of the patient's healthcare needs and clear determination of the most appropriate services and care provided. Simple and quick access to patient information and the health records of different visits makes healthcare services better. Similar access allows special reserved patients to spend less time during visit checks or visits and helps healthcare service providers to better understand care habits. However, manual systems that need to be undertaken are not only time-consuming, but also error-prone, leading to delays and mistakes during patient care and services. Automation of critical patient health record management services has gained interest across the world to mitigate these issues.

A patient management system capable of handling patient visit-related records in terms of a patient visit is presented. The proposed system can help provide better

services by utilizing modern architectural concepts and services. It serves as a partial solution to healthcare-related organizations struggling to automate healthcare management issues by accommodating different types of special developed modules used. Assistance is also provided through developing healthcare IT using a service-oriented architecture concept that directly connects with the external world, such as integrating with different hospitals, clinics, and labs, making it more usable and complete in quality healthcare management services.

5.2. Case Study 2: Telehealth Solutions

Telehealth, introduced to support remote care services, travel limitations, and patients during the COVID pandemic, has sped up since the advent of restrictions. Various major Telehealth players offer different speech, image, virtual portal, and other means for telehealth. However, these separately fast and growing businesses still need to be further integrated with existing healthcare systems, data pipelines, and other enterprise component services. Further integration means that telehealth business data need to be shared for better insights and add more healthcare enterprise functions comprising patient education, intake, triage, scheduling, eligibility, appointment reminders, consent, billing, surveys, assessments, and liaising with still many other departments like reception and case management and transition of care groups afterward.

This process of integration requires that Telehealth service providers and the Healthcare enterprise and service handlers go back and forth to make each other's pipeline standards, API hooks, and method protocols clear enough and this integration takes extreme efforts from clinicians and enterprise personnel. We present a way to integrate Telehealth and healthcare enterprise applications on services and SDK levels with largely reduced effort. Health enterprises have already needed to exchange data and share within each other for years ensuring patients do not return to the hospital readmission due to unaddressed and uneducated post operative care. The established Document Exchange and Fast Healthcare Interoperability Resources formats have proved Confidentiality Security and Integrity processes between partners in this enterprise interconnect have proven to reduce hospital revisit tarif fees that add up to large amounts of penalties if states do not get lowered readmissions.

6. Case Studies in Finance

Data analytics is a great enabler in the finance industry with several financial applications, such as investment management platforms, payment processing, fraud detection, insurance, and payment processing. In this section, we will cover the solution architecture of two applications: a fraud detection system that performs real-time fraud detection based on streaming data and an investment management platform that uses the entire technology stack.

In the fraud detection case study, we demonstrate the solution architecture of a retail fraud detection system that processes streaming point-of-sale data using analytics and machine learning services to find the nearest transactions operating in the same spatial and temporal vicinity of the suspicious transactions and using spatial statistics to compute a proprietary dynamically updated threshold. The threshold is used to determine unusual spatial clusters identified using statistical tests with which we detect the presence of money laundering. The architecture then uses a data lake storage to land processed real-time data into a data lake for reporting or scheduling batch process around the same location. The machine learning model triggers an alert via communication applications when it detects money-laundering fraud.

The investment management platform incorporates two applications: near real-time risk detection for funds using an internal API and scheduling batch reporting upon assigned, triggering alerts via communication applications. In addition to environmental data, the architecture retrieves data in near real-time for the assigned portfolio or fund with an internal API, triggering alerts when parameters are breached.

6.1. Case Study 1: Fraud Detection Systems

Fraud detection systems are essential for modern financial services, especially in the areas of payments, insurance, securities, credit cards, and banking. The financial cost of fraud has always been high, but the advent of deep learning-based advanced models has made it easy to detect even the most complex forms of monetary malfeasance. Transactional data providing details of digital interactions or transactions could be utilized to ascertain misleading instances. As the number of transactions grows, the size of the fraud detection systems consequently increases. The entire fraud detection infrastructure, which groups together several machine learning pipelines, is naturally designed to be microservices-based. The core task of each service is artificially deducing typical behavioral patterns of transactions and predicting the apparent nature of a new

transaction. Any transaction that falls outside these bounds is then classified as potential fraud.

Modeling an app's core behavior is a not-so-easy task, as the decision will be based on a combination of prior history celebrating or otherwise. What works well for one platform might not work at all for another. Fraud prediction methods are constantly being developed and improved upon in microservice fashion. Microservice development within a microservices-based platform seeks to enable the description, construction, and operations of a particular service. It provides the means for building a variance of a full-blown fraud detection platform for assessing fraudulent behavior in a real-time manner at scale. It includes the capabilities for storing giant amounts of data, identifying unacceptable deviations at high speed, and operating such pipelines with multiple machine learning models concurrently.

6.2. Case Study 2: Investment Management Platforms

This case study examines the deployment of a real investment platform based on light-weight assembly of solutions for order management and regulatory compliance, together with deployment of a separate enterprise on-premises applications for portfolio valuation, risk management, and performance attribution. The on-premises solution takes advantage of computation accelerators and high-capacity servers to deploy a multithreaded clustered-code parallel solver for portfolio valuation and risk management. Simulations are run concurrently for thousands of portfolios. Solutions are cached for a period of time at the result store, including “nearby” portfolios that differ by a small number of assets. The presented platform is designed for volatility forecasting using machine learning methods. First, historic excess returns, including public cohort and factor specific excess returns, are estimated a priori. Then, partial excess returns for assets within cohort and/or model factors are estimated with the use of linear regression. After that, after backtesting, the predicted excess returns on a static basis are used in conjunction with assets' share prices to provide alpha forecasts over various horizons – one month, three months, or until the share price reaches some target level. It is important to understand that keyword and label-based filtering of investment policy domains as well as dynamic user-defined criteria are used also to refine portfolios and provide additional company-level signals needed for risk and portfolio management.

Algorithm-based active investing approaches, such as statistical arbitrage, employ simultaneous buying and selling of securities in different markets or in a particular index in an attempt to profit from price discrepancies that exist for a very short time. A distinct sub-group within statistical arbitrageurs relies on pairs

trading techniques that identify pairs of cointegrated assets for direction-neutral trading.

7. Case Studies in Retail

Microsoft Fabric supports two different SaaS-related needs of the retail sector. Firstly, numerous companies within the retail sector provide Software as a Service solutions. Microsoft Fabric helps improve service performance across the board, elevating quality and reliability factors, which ultimately benefits all retail companies and their customers. Secondly, at least 85% of the key functions in a typical retail operation rely on one or more enterprise applications. Traditionally, each of these applications is implemented and operated separately, and these applications do communicate data and share information and insight. However, the lack of coherence among these applications often leads to duplicated efforts in some functions; inapt guidance in several areas; and erratic, sudden, and different responses to similar situations. Microsoft Fabric promotes an integrated management of these enterprise applications, increasing the coherence across the board and enhancing all the underlying functions. While focusing on the retail sector and the integration of enterprise applications, this chapter describes two specific case studies, where Microsoft Fabric boosts the quality and reliability of different retail SaaS solutions and integrates and enhances multiple enterprise applications on the retail realm: e-commerce platforms and inventory management systems. Read on to find out how both efforts benefit the key retail functions related to customer communication and rapport as well as operational activities and supply chain management. The two case studies can be relevant for readers whose main interest lies in the retail field and would like to find some high-level, non-technical illustrations of what Microsoft Fabric can do as either a SaaS provider or a mainstream enterprise application integrator.

7.1. Case Study 1: E-commerce Platforms

Abstract: In this chapter, we introduce two case studies in retail, with a focus on SaaS and Enterprise Applications. The first discusses e-commerce platforms, such as hosted SaaS application solutions that enable retailers to deploy e-commerce services without incurring the burden of painful e-commerce infrastructure development. The second covers ERP solutions that allow organizations to manage enterprise planning, resource allocation, and inventory flow in a cost-effective way.

Introduction: Retail organizations provide goods and services to general consumers. In recent years, retail organizations have turned to cloud technologies for their flexibility, scalability, cost, speed of deployment, and ability to leverage large amounts of data to services across the retail spectrum. We provide two case studies in retail: Hosted E-commerce Platforms and Enterprise Inventory Management.

The shift of retailing from an offline world to a digitally native and omnichannel one has moved most consumer-brand and business-brand interactions to the digital space. E-commerce, mostly in the form of hosted SaaS applications, has made selling online straightforward. This is especially the case for smaller brands located in the Global South. A few globally dominant platforms have enabled brands of all sizes to promote their goods for purchase from the website and social media channel to enable buying online. Leveraging the ubiquity of payment and logistics providers, where buyers can pay through a number of secure online services, and a few logistics providers have enabled fast and relatively cheap shipping across wide distances, e-commerce has opened up new avenues for brand creation and sales.

The establishment of cloud e-commerce more generally, and store delivery, drop shipping, and commingled warehousing of online and in-store inventory pool more specifically, has decentralized retailing economy power from a few players back to the entrepreneurs and smaller businesses themselves. Margins are now spent because there are no bigger businesses to take it away. As is often the case when technology shifts occur in the entrepreneurial space, the few companies control most of the global market e-commerce size. Hence the retailer interest in these platforms.

7.2. Case Study 2: Inventory Management Systems

Electronic retailing simplifies purchasing goods remotely through a few clicks. However, handling and delivering goods on behalf of manufacturers is a complex task. Retailers need to organize the storage of different items at remote inventories and efficiently deliver orders using the least time and cost possible. For example, if someone orders a special model of shoes available only in crystal red color for immediate delivery, the shoes should be on a shelf of a close warehouse to the customer or should be delivered air-freighted from the factory. In cases like this, no one would wait two weeks for a boat to arrive.

Retailers rely on inventory management computer systems to keep track of stocks in warehouses all around the retail chain and in transit. These systems use complex business rules to recommend retailers which pieces should be shipped

to where and for how long, so that stores do not run out of items frequently, and customers do not have to travel to several stores to find what they want. The systems define optimal stock levels at all parts of the retail supply chain, also analyzing factors like demand predicted by sales forecasting systems, the current distribution network, transportation costs, profitability, and lead times. When stores cannot afford to carry enough stock, inventory management systems recommend which products to be manufactured also considering minimum batch sizes and quality levels.

Due to the importance of the data they handle, inventory management systems require not only high processing power and storage, but availability and dynamic scalability at a lower cost. Missing an order pose severe penalties, as business partners penalize with fines all parties related to a failure in supplying the product. Due to the need of processing several transactions large concurrently, these systems are expensive and hard to create and to maintain.

8. Challenges in Implementing Microsoft Fabric

Notwithstanding the advantages listed in the above sections, implementing Fabric into a modeling or analytics process is not free-of-risk. The benefits and challenges highlighted below should be contemplated for a thorough consideration of the overall utility of Fabric. The Fabric ecosystem is complex, with numerous tools integrated between several different workloads and providing services spanning the full analytics lifecycle, from data preparation to modeling to operationalization. This means that coordination of existing processes around this new central tool will take effort.

In particular, Fabric's unified security model, which provides robust at-scale security for each of the workloads, functions, and APIs, while powerful, may be insufficient for some use cases. The Fabric security model is an abstraction around the cloud-level security already provided by Active Directory and Resource Manager. Active Directory and Resource Manager employ a security model that is focused on security roles applied at container, team, and organization scope. Fabric provides security at all levels of the solution, from the Fabric workspace to the individual data items.

Often security requirements mandate a significantly more granular security model than that provided by Active Directory and Resource Manager, and this can be particularly true for data access layer tools. For scenarios that require a finer granularity of data control, there are features that allow for row level

security in datasets or tabular models. This concept can be applied to Fabric as a whole but contains limitations, however, depending on the model in use, and organizations with demanding governance needs may determine that the benefit of Fabric does not outweigh the costs or risks associated with security implementation.

8.1. Security and Compliance Issues

Most cloud solutions currently in use do not consistently support enterprise-level security features. For example, directory services for enterprise applications and solutions do not have full support for certain security frameworks. Shared datasets for enterprise applications enabled through the portal cannot use a service principal that opens a service principal dialog without a security warning. It is possible for the set of APIs used by enterprise applications to have inadequate security controls, including insufficient divergent access management settings between service principals, providing certain capabilities that are not typically allowed for user credentials.

Aside from directory services, cloud solutions may also encounter issues with organization security compliance requirements. Hosted files in common storage locations may result in either a lack of physical control over document storage or a data-at-rest security liability when content is shared publicly or between organizations. Also, many cloud services mitigate security from particular vulnerabilities but don't control all aspects. They deflect related impacts through high availability without controlling the actual upload story since they don't have visibility into how large document files are created.

Some problems may result from the composited nature of data solutions. Beyond the possible inherent existence of exploitable APIs, all the data representing information about an organization is created by and leveraged across different cloud solutions that may not have similar security compliance methods or capabilities. The convergence of operational and analytical workloads can allow for data loss through behavioral mismatch and different recovery methods. Self-service analytics or experience-driven data may violate internal controls as it applies to content consumers leveraging data that do not have appropriate data access rights or have signed off on the shared file for self-service methods.

8.2. Performance Optimization

Obtaining the performance required for a production SaaS application or key enterprise systems within a Power-Query-based architecture is not trivial. Performance involves execution latency and resource utilization. Reducing latency typically involves simplifying data movement and transformation.

Historically this has scaled poorly. Implementing query folding and getting the number of transformations in the analysis to a minimum are key focuses. Performance of interactive queries for SaaS applications is a key business driver and the teams are constantly working on improving query folding, engine improvements, and optimization heuristics.

Training about your usage, such as commonly queried tables, forecast query frequency, and timing, are important for performance. Your administrators can configure increasing workloads driven near continuously by operation systems to help learn optimal timing between refreshes. Additionally, certain dataset properties can be used to optimize performance. For example, for Power Query datasets, refreshing query chunks early in off-hours with the most recent finished data and proximities to heavy utilization times for other datasets or dataflows helps optimize performance of these datasets. Another example is for Direct Lake datasets, enabling storage tables, which are pre-computed optimized tables with the relevant partitions at the right granularity for the relevant queries. They are pre-filled and based on scheduled and frequently run queries over the times and days and valleys of expected open system usage. Please note you have to enable these features while you want to have optimal performance and the costs can bump because of it.

9. Best Practices for Deployment

Microsoft Fabric for SaaS and Enterprise Applications can benefit with some best practices when deploying the platform. A phased approach that starts with a proof of concept is recommended. Once the initial POC is successful and the solution has been further reviewed for some Deployment Characteristics (such as Live / Unplugged, Development / Maintenance effort, Runtime Performance, Data Update Frequency, Data Security / Privacy Characteristics), ingest flows are built and refactored.

As ingest flows need to connect and understand data from external systems, initial queries and development can be performed with Azure Data Factory. After flows ingestion and after External Attribution for tables ingestion / Fabric Factory configurations, grabs with mapping for External Dependencies could be created to identify/explore layers/subsystems.

Deployment of the Application Factory should be done with caution as this will become the main area for Creation and Catalog of Application Assets, such as Data Factory Pipeline Orchestration.

Both Data Factory and Application Factory do have Items Specific to Deployment Environment (costing variables, Dev/Prod dependencies). In order to facilitate development and deployment when using CI/CD tools, consider the following:

Create reusable patterns and put them in collections to facilitate Data Pipeline Creation.

Centralize Shared Configuration. Create common Reference datasets or parameters for common external Access setup, Database configuration Parameters, Dataset Cache.

To facilitate change, have External Dependencies that point to common datasets with References. These can be Blueprinted or SQLed during Deploy.

Monitoring and Maintenance

Microsoft Fabric does provide monitoring capabilities with Azure Monitor capabilities at Pipeline, Pipeline Activity, and run level for dataflows.

Other monitoring items can create Data Quality Issues when ingesting data. Repeatedly checking allows retriggering error issues with the data after some other failed inspection.

9.1. Monitoring and Maintenance

Establishing an observability and monitoring framework for your solution from the start can help minimize the risks and overhead of maintaining your solution in production. You should capture telemetry for various application components and layers, all the way from ingestion of requests to processing and storage. Examples of telemetry that you should consider capturing are:

• Key application metrics

• Application-specific performance metrics

• Latency and failure in the key application processing workflow path

• Latency and failure in various key application backend components during a transaction

• Latency impact of downtime of downstream services during processing

• Scalability during regular, peak, and stress loads.

• Data profiling: the state and characteristics of input and result datasets

• Latency, failure, and profiling the key task flow paths for temporary tasks or buckets

• Resource metrics that may lead to throttling in the key service components and backend services required for processing user requests in low-latency scenarios

• Alerts and corrective actions on significant deviations from normal conditions, and for appropriate metrics listed above

• Routing telemetry to a centralized location and monitoring tools for insight and action: the telemetry data should be routed to a platform or pipeline that can handle the volume and burst jobs in a centralized place.

9.2. User Training and Support

General availability does not mean that the product is ready to be used in production [1-2]. There are many details and nuances that require adjustments and configuration. In addition, each company has its own peculiarities regarding what products are being migrated—data-center-based solutions or cloud solutions, which data-exchange processes are more sensitive and require the least downtime, which users are the most affected during the transition phase, what are the company security policies that the affected accounts should comply with—and so on. Even for those from similar industries or with similar setups, it's unlikely that everything is equal. Therefore, meticulous planning is always needed. More than that, those who are going to experience integrations, replicas, migrations, and other changes are users who may have absolutely no experience with any of those new features that are migrating to the cloud. Their habits don't go through a period of adjustments and are, then, reestablished in a new rhythm. What they do know from the on-prem solutions is that their experience with migrations and other periods of downtime was never a good one. Even after everything is done, features work better or worse, and data storage and retention requirements could be different from what the users expect. The barrier of change is only growing bigger. Therefore, either to help lessen the impact or to serve as a permanent support for everyone, training users is a critical component of the process.

10. Future Trends in Microsoft Fabric

As we look ahead at potential industry trends that may affect Microsoft Fabric and CloudFirst, two come to mind immediately: deeper AI and machine learning integration, and an evolution of SaaS architectures.

AI and machine learning are on a steady march toward being more and more integrated into the products we use every day. Microsoft has embraced this trend strongly with the recent introduction of capabilities such as Copilot in Microsoft 365, and generative AI capabilities in Azure. Microsoft Fabric unlocks a treasure trove of rich data that cloud-based apps often have, and it would make sense for the intelligent features of apps leveraging Microsoft Fabric to have access to these datasets. Predictive analytics, data-driven decision making, anomaly detection, and journey mapping are just a few examples of common use cases that could be enabled with generative AI features tied back to data stored in Microsoft Fabric.

Cloud-based Software as a Service (SaaS) applications began life as largely point solutions for common business tasks. But these SaaS services began expanding their capabilities, then upped their games with deeper integrations with other SaaS services. Major players then went on to build Business Application Clouds, bundles of services that together form a broad-backend digital operations backbone. Microsoft 365, the collection of enterprise productivity applications that includes Word, PowerPoint, Outlook, SharePoint, OneDrive, Teams, and others, connects deeply with Azure Active Directory, Azure DevOps solutions, Microsoft Graph, and Power Platform. Applications built on top of Microsoft Fabric will also be able to use the capabilities of Microsoft 365 and Power Platform to enable business workflows that pull data from the applications built on top of Microsoft Fabric.

10.1. AI and Machine Learning Integration

Enterprise applications can be designed with embedded features that usually require a third-party module developed on top of the existing core architecture. AI is an essential component in most of the products that deal with customer interaction. The data collected during these interactions is the fuel for the AI engine, and the value of AI is directly correlated with the quality of the data. It is commonly accepted that enterprise applications capture only a small amount of the data that is needed to generate a good AI model, mainly company-proprietary data. Combining different proprietary datasets together will create better models with higher accuracy. SaaS product companies need to partner and combine their datasets with others to generate models for customer success. The UI that presents business intelligence or reports is often a marketplace where many statisticians compete to have their models seen first by corporate decisionmaking users. Integrating AI into BizApps such as Power Apps or stub SaaS applications used as part of Excel/PowerBI/Office 365 would allow new, exciting, enterprise-related AI models to be created and used.

Most customers rely on various platforms for management of their customer data layer and for updates on their communications with potential customers during the customer life process flow. These are, thus, the industry leaders providing the first dataset used to fuel large language models specialty tuned for enterprise customer management. Data contained in enterprise business applications is much more accurate than that from social media and has stronger direct correlation to business-related outcomes. The highest demand would be for these enterprise-specific language models if they were easy to create, accessible, and robust. Today, companies turn to professional data-labeling services for these language model training datasets, but it's obvious that conspicuously missing are the low-friction shared utilities integrated inside enterprise products that couple together input service with desired outcome. Imagine an Assistant used by sales during their interaction with the customer that generates the meeting summary useful for the customer success management team.

10.2. Evolution of SaaS Architectures

SaaS is a well-established software model where end-users have access to enterprise applications via a network. While SaaS applications have been around for a while, they are evolving rapidly. The need for easy business-to-consumer and business-to-business integrations has resulted in companies needing to extend basic functionalities of independent Software Vendor (ISV) SaaS solutions. These aspects are mostly unique to Enterprise SaaS solutions. Some of the key expectations from enterprise SaaS solutions include:

- Built for Business Workflows: Enterprise SaaS solutions need to be built for specific domain business workflows when compared to B2C application suites. - Integrated Mainstream Business Workflow Solutions: All use cases cannot be solved by single vendor-centric solutions. It is necessary to implement or enable extending ISV-focused Enterprise SaaS with User Interface plug-in support. - Embedded Domain-centric Process Automation: Combining multiple mundane micro processes together and invoking each other in the right way is an automation challenge. Enable these via domain-specific internal workflow automation across SaaS ISVs. - Business Centralization with a Decoupled Microservices Architecture: Even though applications need to be semi-centers, it should not lead to redundant data living when managing the core business process across ISV enterprise SaaS solutions for the domain. - Switching Cost Barriers for Better Monetization: As business-centric services become highly aligned customized, switching costs become a barrier towards monopoly facilitating ISV monetization at higher margins. Monetization can also shift towards transaction-based models that can enhance SaaS company growth and profit at scale.

These expectations leverage the growth of the SaaS ecosystem. However, the use of no-code and low-code constructs in the above will enable entrepreneurs to address domain-centric application demands appropriately. In turn, ISVs use Fabric SaaS platforms to reduce time-to-market or enhance any one of the above pillars that make their SaaS solutions different and better than other options available to customers.

11. Conclusion

In conclusion, Microsoft Fabric helps organizations to reduce the time to insights, adopting a unified analytics solution powered by a simplified, cost-effective, and fully integrated platform. Organizations looking for a centralized platform for their Business Intelligence solution can easily use Power BI capabilities, embedded in a Microsoft Fabric environment. Other organizations with customized Business Intelligence front-end solutions can also benefit from the Microsoft Fabric Data Engineering and Data Science capabilities to build a data and analytic solution faster than any other similar product in the market. This paper explored the Data Analytics and Business Intelligence features of Microsoft Fabric, explaining how organizations can adopt a complete and unique solution to generate insights from their data with a focus on SaaS and enterprise applications. We described the lifecycle of a data pipeline with Microsoft Fabric, with its previously called OneLake Data Lake and its analytic engines: Data Engineering, Data Science, Data Warehousing, Real-Time Analytics, and Power BI. In this paper, we also guided the choice of the engine best suited for the individual steps of the data pipeline lifecycle. Exploratory Data Analysis has also been covered, showing how incorporating Explorer experiences brings a new dimension to data exploration and helps Data Citizens to start data-driven discovery journeys. At last, organizations looking for a complete solution for building Business Intelligence capabilities can benefit from all the integrated offerings of the Microsoft Cloud, including Microsoft Azure for Storage, Computing, Data Management, Azure Open AI Services, Microsoft Dynamics 365 for Corporate Applications, and Microsoft 365 for Office Applications. In the near future, Business Applications will count with more data-driven features by also incorporating Microsoft Fabric capabilities to their native solutions. Data-driven capabilities are the future of Business Applications and adopting the complete Microsoft strategy for Business Applications, Data and AI, and Work Management will be the best approach for organizations wanting to differentiate and succeed in a competitive market.

References:

[1] Borra, Praveen. "Microsoft Fabric Review: Exploring Microsoft's New Data Analytics Platform." *International Journal of Computer Science and Information Technology Research* 12.2 (2024): 34-39.

[2] Bai, Haishi. *Programming Microsoft Azure Service Fabric*. Microsoft Press, 2018.

Deep Science Publishing, 2025
https://doi.org/

DeepScience
Open Access Books

Chapter 9: Operationalization and Monitoring of CI/CD Pipelines for Fabric Assets

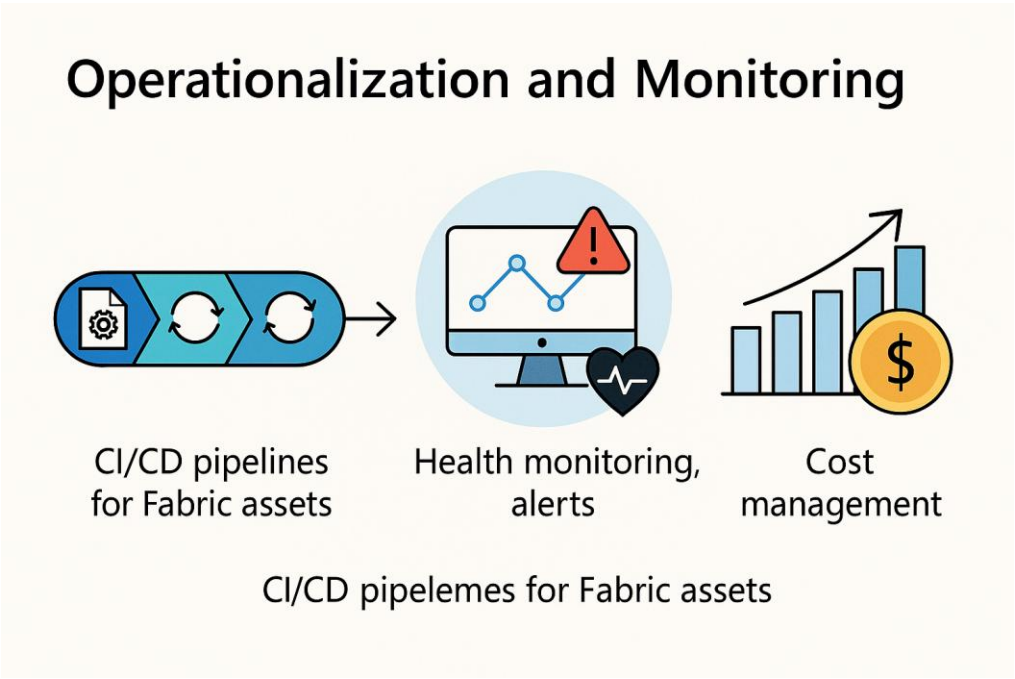
1. Introduction

Monitoring sophisticated CI/CD environments may quickly become unmanageable if done through specialized solutions. In this work we present a solution for hyper-converged infrastructures that merges CI/CD capabilities, through modules for the declaration of pipelines of multiple applications in different stages and support for targets defined in the framework, with an exploration environment for monitoring assets and the pipelines' infrastructure, built upon and extending existing functionalities, namely entity management and events emission at the network level.

CI/CD solutions based on pipelines have evolved, for more complex software systems, to embrace not only code repositories but also the supporting services necessary for their deployment and runtime operation, which may encapsulate business logic, such as data management, as well as infrastructure such as cloud clusters. This supporting services evolution is now embracing configuration files that describe network topologies and system behaviours, moving towards a full description of the system at hand.

We present a set of extensions to the framework that enables it to support Assets, a kind of entity that at higher level embrace both application and supporting services. Assets, based on a richer description, allow for a pipeline structure that orthogonally defines the path distributed along the stages of opening, forking, building, testing, and deploying to target resources of a Software Concept. What Assets enable going beyond the Software Concepts is the ability to also use the asset pipeline either automatically or with little human assistance. Moreover,

Assets also allow a human consultant or consultant team to be defined, enabling human intervention at any point in the asset's lifetime.



2. Understanding CI/CD Pipelines

The terms Continuous Integration (CI) and Continuous Delivery (CD) or Continuous Deployment, refer to a combination of operations centralized on source code repositories and systems that automate integrating changes and verifying the result through multiple steps that create consolidated and reproducible application packages and services, making it ready for deployment. CI/CD pipelines are software tools that leverage automation to facilitate and execute these operations. In a CI/CD pipeline, software changes are automatically built, tested, and prepared for deployment to production.

CI/CD pipelines share important practices, either related or stemming from DevOps concepts: version control, to systematically track changes; branch and merge strategy, to divide and unify tasks delegated to different teams; package management; infrastructure as code; continuous monitoring; testing at all stages; immutable infra; test in prod; clinical deployments; automated release orchestration. While not all CI/CD tools implement all practices, any serious pipeline must implement compliance, monitoring, and observability, especially

in the case of deployment pipelines. Version control enables collaboration while maintaining consistency. CI/CD pipelines typically monitor a version-controlled code repository for changes and trigger an automatic pipeline execution for software builds, tests, and deployments.

Continuous integration is a best practice in DevOps in which developers integrate code into a shared repository several times a day. With each integration, automated builds and tests are run to detect errors. Delivery requires that every change can be released but is not automatically released to production. It's supported by built pipelines that automate steps required to release the change. Deployment automatically releases the code to production with no manual intervention required.

3. Fabric Assets Overview

TAP's Platform provides Foundations on which Development Teams can deploy their Applications and Data. Infrastructure Footprint and Operation Workloads can be large. Specially when we consider mission critical applications and the associated SLAs. Modern ways of developing and managing these applications have evolved over the years. Complexity among both and collaboration between them is growing. Deploying and managing Middleware, Network Configuration, Security, Monitoring, CI-CD Footprints, Services, Operator, Agent across different Clouds and on-prem Deployments are complex. Both at Scale and Size. Inner Source principles and Vision enables application teams to adapt proven path and solution for Application Lifecycle Management needs. Logically combining Development and Run, effectively automating Triggers, Workflows and responsible parties and Collaboratively Developing Quality Assets and Templates and focusing on Asset and Template Life Cycle Management is key.

The secure fabric onboarding, continuous security compliance automation, scalable connectivity, AIOps and Observability platform uses secure engineered open-source software. Development, DevOps and SecOps teams can collaboratively operationalize the data and application services CI/CD pipelines with predefined policies for deploying, onboarding and monitoring applications and data services and templates/assets. Templates are tailored to address the complexity induced by the multi-cluster environment, the scale and size of deployments, the need for a heterogeneous mesh of Enterprise Applications, the nature of the dependency between enterprises and external applications, and the need of operational efficiency to respond to wide variety of incidents. Templates

and assets are modular, extensible, configurable and templated. Automated on-board monitoring dashboards with built-in alerts are provided out-of-the-box.

4. Operationalization of CI/CD Pipelines

Implementing CI/CD pipelines for continuous automation of Fabric asset lifecycles must aim to minimize the required human intervention for tasks that are repetitive and prone to errors if performed without tooling. Therefore, further planning is required to address the specifics around the CI/CD pipeline operationalization, how pipelines are maintained and supported, and how developers and DevOps engineers run the pipelines for their development workflow and day-to-day operations. This section addresses this planning phase covering pipeline operational requirements, integration with existing enterprise systems, automation of deployment pipeline tasks. The end goal is to deliver developer-friendly and reliable CI/CD pipelines that provide integrated support for a full Fabric asset lifecycle, producing reliable and reusable assets.

4.1. Defining Operational Requirements

Infrastructures supporting Fabric-powered applications may reside on enterprise environments or be created by the enterprise across cloud edge infrastructure service providers, together with Fabric instances and their workload management. Such infrastructures are subject to internal enterprise procedures that monitor compliance with operational policies and security controls and maintain a catalog of supported resource indices. Compliance and accountability over standard enterprise operating procedures must be addressed while defining CI/CD pipelines for deployment, maintenance, reconfiguration, and troubleshooting of Fabric assets. This includes defining check-and-balance gates for potentially dangerous tasks, audits, and rules describing who and when is allowed to invoke such pipelines.

The first step towards operationalizing a CI/CD pipeline follows the traditional software engineering lifecycle of identifying requirements. In this case, the focus is on identifying the operational requirements necessary to automate the build, testing, and deployment of code aimed at modifying the behavior of a Fabric asset. These requirements should cover a variety of concerns. At a high level, they should define the integration and testing coverage necessary for the CI/CD pipeline to provide assurances that builds of the Fabric asset are safe to deploy to Fabric environments. Higher levels of assurance would normally employ larger numbers of test cases, where more critical aspects of the asset's functionality are

thoroughly evaluated in order to establish a much higher coverage ratio, as well as shorter feedback loops, where incomplete test reports or assessment failures are resolved expeditiously to avoid prohibitive delays between code submission and successful asset build or environment readiness for deployment.

However, such longer test suites or shorter feedback cycles would not necessarily be difficult to implement in the definition of the operational requirements, given the most common case of CI/CD pipelines only operating in the horizontal axis of code mutations. Therefore, at a more detailed level, the definition of the operational requirements also imply the cross-cutting concerns that usually dominate the design of these plans. These details include the types of tests necessary for the pipelines, testing configurations and parameters to be used, and how dependencies between tests should be managed, such as controlling the order of test execution and the hierarchy between tests. Existing efforts such as the traditional test classification, testing techniques, and testing varieties may be inherited by pipelines as further guidelines to help such definitions. However, the cross-cutting concerns and the test classification cannot directly define the possible configurations for these pipelines. This must take into account a wider range of operational requirements such as risk management, defined activity lists, and available resources.

4.2. Integration Strategies

Continuous Integration (CI) is one of the earliest and the main components of CI/CD pipelines. CI follows the Unix philosophy of "do one thing" by addressing a very specific challenge: get validated changes into the main branch of version control. Without continuous integration, Continuous Deployment is deprived of the 'testing gate' element. As such changes are rapidly and continuously made to the code base, the feedback from the CI runs that usually are tested through multiple environments, become the most critical parts of the pipeline. It is only natural that an initial thought of setting up a CI/CD pipeline will focus on the integration strategies used during CI. Integration strategies refer to the various techniques, practices, and tools aimed at enhancing the delivery pipeline through multiple validations that will securely produce a validated version of production-ready software, or asset. Integration strategies come into play as a set of pipelines when source code is extracted from the version control repository. In its details, CI serves the overall goal of ensuring continuously integrating latest changes (build) primarily consisting of source code, eventually composing Containers/Packages, and continuously validating such composition by executing the Tests that ensure the Container/Package meets all prescribed criteria (validations). The orchestration of a CI pipeline follows this order: (1)

Pull the code from the source code repository; (2) Build the software – fetch dependencies of the application, build the executables, package the program into a binary - a Container/Package; (3) Test it repeatedly and as much as possible; (4) Publish the container to a registry or repository; the should work with integrity checks. Supply all necessary data from a pull request to facilitate a quick and reliable code review process. Only support validated, up-to-date executables using source control arguments; and (5) Continuously and Automatically Deploy containers/packages from the repository to the target assets/containers/hosts/instances.

4.3. Deployment Automation

One additional component of CI/CD is Deployment Automation in a multi-node, multi-fabric infrastructure. Deployment Automation for CI/CD can include deploying to cloud infrastructure, deploying to on-prem data centers, and deploying to Fabric. Typically, the full machine learning/deep learning pipeline will involve multiple steps from data identification to feature engineering, exploration and analysis, model and testing and finally migration of the model into production. Each of these steps may need to be performed multiple times with different parameters and might involve specific compute environments. The Pipeline should be a generic one that tracks all of these steps and allows for parameters to be configured at each of the steps.

Since all of these steps may be performed by different roles and for different projects and different kinds of models, it is critical to manage these Pipelines. Current CI/CD Pipelines are great for manual deployment of code and model artifacts into a production environment from source code repos after the code has been validated to work correctly. However, the CI/CD Pipelines typically need to call out to infrastructure specific Domain-Specific Language to enable replicated transform of computing environments and for executing the various tasks. The compute platforms are responsible for the data download/qualify/checkpoint and other necessary steps to do the replicated job. The Pipeline definition stores parameters for the user-selected features/metrics/drivers but does not get into the details for each compute target.

5. Health Monitoring

Monitoring the health of deployed Fabric assets is essential to empower exploring (diagnosing) the root causes in the case of failures as well as ensuring a continuous delivery experience and the overall user satisfaction. In other words,

some actors in the deployment/commercialization process would want to assess how well a deployed service is performing as compared to certain preconditions set during the contract negotiation and closure phase or to how well it is performing as compared to the competition. Such decisions are also needed to be made on the asset as a whole but also on each specific pipe/user class that the service is providing. A monitoring phase is launched once the service, consisting of one or more assets, is deployed. This set of services is operated over time in order to determine the need for refresh, renewal and upgrade.

5.1. Key Performance Indicators (KPIs)

Key Performance Indicators (KPIs) Monitoring also entails more than just ensuring that the service is being accessed by users. A hot commodity in the communications world is that of Quality of Service; this is linked to the performance of the service as perceived by the users; it is an average metric over service users. On the other hand, the Quality of Experience is more focused on the specific users and their perceived service satisfaction. Another metric is that of the Key Performance Indicators; these would focus on a specific application or service pipe, setting forth certain conditions that need to be satisfied for the service to be considered successful. This finally implies new missions for all departments in the commodity services. Not only R&D teams but also the Service Reliability Team, Marketing, … These are some of the service deployers that need to ensure QoE based on either KPI or QoS functions that are valid within the deployment contract/user.

Monitoring and alerting are flagships of the DevOps culture and practices. The construction and operation of continuous deployment pipelines brings a factory view to the pipelines and finishes them into a product. As such, they need maintenance and caretaking like any other product that we would create for our clients. It is only natural to have the same approach that we would follow for any of our technology products on the operation of the CI/CD pipeline(s): i.e. monitoring its effectiveness and performance, and alerting for abnormal situations. There are a number of ways to measure the “health” of a deployment pipeline (measured as a CI or a CD function), so as to assess the quality of the pipeline, be it in its construction and evolution, or as a functional service.

There are recognized best-practice performance metrics that focus on the speed and time of various popular stages of the CI/CD pipeline function. Such pipelines are expected to be fast and agile, serving the needs of the deployment process and enhancing the efficiency of the validation. Some popular metrics are: frequency of successful deployments to production or code changes per time period; lead time between a code change being committed to it being deployed

to production; percentage of deployment fails; mean time to restore for a failed pipeline; change ratio; change-related validation duration; the ratio of validated changes; the ratio of automatically and manually validated changes.

5.2. Monitoring Tools and Technologies

Monitoring Tools and Technologies Let's now have a look at monitoring technologies and tools. About the Satellite world, several commercial services already exist for data; one major player, among the ground segment services mentioned before, would be the Space Data Association. Regarding satellites, a good amount of flight data, including data, are made public.

As CI/CD pipelines proliferate, there must be a systematic, tool-driven approach to monitoring the pipelines. Towards that, there are specialized monitoring tools and technologies that can be leveraged. However, traditional monitoring technologies such as monitoring offered by the vendor ecosystem for Fabric-like assets or those for enterprise applications available in enterprise ecosystems can also be leveraged.

Monitoring of general-purpose CI/CD solutions is available in the enterprise ecosystem. For example, Quality Assurance tests are part of any CI/CD pipeline that support system-level verification, and CI/CD software products provide monitoring and tracking of the quality of the business assets. Monitoring capabilities are available out-of-the-box around build/test duration by use of methodology-driven KPIs; it can be augmented further by integration with enterprise ecosystem monitoring solutions.

Others do not provide direct monitoring capabilities, but they can be integrated with common enterprise monitoring frameworks for monitoring development activity tracking. Various DevOps dashboard products do have capabilities for CI/CD solution, providing usages, trend variations, and SLOs to track with respect to using the CI/CD pipelines by dependent users, and their monitoring applies across any CI/CD solutions.

5.3. Real-time Health Checks

Health monitoring of fabric assets is mainly comprised of perimeter-based health checking approaches, which is the most flexible and manageable approach. Health checks are usually performed periodically according to specific schedules configured by system administrators. Despite the drawbacks that have to do with scheduled health checks such as latency in anomaly detection, performance impact of routinely firing security operations, they are the most commonly used methods for perimeter security. These scheduled checks can monitor external

properties accessible from outside the fabric asset, such as the response times of web and database server connections, active or passive remote control access, or the status of VPN connections. Some common technologies in use to periodically export health metrics are.

In contrast to scheduled health checks as mentioned previously, a new class of health-checking methods is emerging. These novel techniques provide administrators with a number of real-time metrics that give insight on how the health of a fabric asset changes over time for risk foreseeing. Recent advances in host resource monitoring and management and software appliance design and development have made it possible to continuously monitor a number of resource use attributes and incentives including CPU use, memory use, disk uses, system load, and swap use. Such APIs typically allow for near real-time observation of a managed resource and notify when changes occur. In addition, a negligible resource load overhead is required as a result of agent-based systems. The periodic nature of these APIs makes it infeasible to obtain multi-snapshot insight on multiple attributes.

6. Alerts and Notifications

Alerts and notifications serve different purposes yet they are both indispensable to operationalizing CI/CD pipelines. In case anything during testing within CI/CD processing goes wrong, developers need to be made aware immediately so that troubleshooting takes place, which might even require engineering teams to collaborate. CI/CD processes are often automated in that once developers code and check in to the version control system code repository, they seamlessly kick off the whole CI/CD process. If any testing is placed at those checkpoints, it notifies developers about the need for code changes either to continue using or building on existing code.

In continuous monitoring, there are different reasons for notifying users, as developers are no longer the only stakeholders impacted by the artifacts and services being built and deployed. The reliability of services created by a built CDFE is just as important and it could also be a user query about issues that flag some other performance metric indicating something might be wrong. In such a scenario there might need to be a Reliability Team as users often find out about performance misalignments before the responsible teams do. In enabling users to alert a team responsible, issue visibility may be enhanced, while issues may also

be resolved faster even after there is a divergence from expected infrastructure behavior.

Even with the examples just provided, the intent behind alerts and notifications is different. While alerts are more prescriptive in allowing actions to happen, notifications are also informative in informing users on actions taken or not, which require no action at the recipients end. The latter are also less likely to lead to action taken, although they could certainly trigger one. In summary, CI/CD alerts are actionable; monitoring alerts are descriptive; notifications are informative.

6.1. Setting Up Alerting Mechanisms

In CI/CD pipelines, a best practice is to create notifications for completed scripts. This can be achieved by having the last step of the script use a notification tool to notify that the pipeline has completed. Notifications can also be set. Whenever the repo under governance has a change, notifications can be automatically sent to a predefined channel. These notifications have the potential to contain all logs or the important logs from their execution.

Another best practice when using CI/CD to deploy contracts is to set up alerting mechanisms to monitor the pipeline, ideally through pre-existing services. If the CI/CD has been set up by wrapping the infrastructure, a monitoring agent can ease the process of alerting when an asset is deployed in a way that could become harmful for the environment such as with pre-established bug doors and privileged access.

It's important to enable alerts for the monitoring agent. When alerting via a channel, it's also important to set up a command to be able to trigger the health-checks in real-time. This would be the first step for a CI/CD wrapper that chooses to connect to a monitoring suit to track the runs. Following these steps would allow the wrapper to execute CI/CD pipelines for various modules, as well as alert on essential steps.

6.2. Prioritizing Alerts

The very reason for an alerting mechanism is to notify a user (or a program) that something "outside the norm" has happened. This means that we are, in some way, marking specific events as being of interest in what would otherwise be a ridiculous flood of messages over a wide array of systems. Due to the critical nature of many alerts that systems might generate, deciding to prioritize which alerts will be shown, and even which alerting mechanism will be used to issue them, may itself be part of the monitoring protocol (potentially the most

significant part.) Answering the question of which alerts to show the user, when, and in what format of presentation (if any) is a task that is typically performed as part of the internal configuration of each alert. As previously stated, different alerts will have different priorities. An alert that notifies the user that CPU usage is fluctuating at approximately 80% for a span of two minutes will usually have a different level of importance to that of a message indicating that a service which is considered critical for company operations are not responding after a given behavior, often for a slightly longer time span. A different kind of alerting mechanism configuration will set different times and parameters for alerts. Each specific monitoring operation, with its own concept of priority, will typically map back to a predefined alerting system condition. In addition, the alert also has to specify who the contact points are, and the right investigation point to report back about the state of the asset in case of clear problems. The overhead created by the myriad of possible parameter combinations for alerts allows users to customize sensibly the way that systems will communicate with them about potential critical issues.

6.3. Response Protocols

Building response protocols enables an effective monitoring strategy when incidents happen. Basically, in response to a problem which generates a technological, and consequently business impact, it should be built a documentation in order to determine how to act and how to avoid that that problem happens again. Creating technical runbooks, it is possible to redeploy the knowledge of how to react without having to ask for information or wait instruction from the team that understands the pipeline. This way of delegating allows to put in practice frequent mental simulations of such situations, so when the time comes, the team or developer tech can respond as fast and efficient as possible, and the self-healing mechanisms can work optimally as well.

There are four ideas in a response to incident that deserves to be explored here. The first one is about prioritization of what has to be put into self-healing scripts, the second one is documentation of self-healing steps, the third one is about dedicated people to monitor the solution, and finally we have the rotation of such dedicated people. For prioritizing what must first be added to our runbook for self-healing, a good first step is to analyze the past incidents and responses. Whenever an incident occurs is interesting to check what should be automated. If a number exceeds two or three times, then it should enter the self-healing scripts for sure. Collaboration is key here because the DevOps concept aims to break the wall between the two previously separated departments, Developer and IT Operations.

7. Cost Management Strategies

Managing the costs associated with CI/CD pipelines requires careful consideration and strategy. Organizations should initially develop an understanding of the types of assets that will be used in CI/CD services and approximately how many units of each will be used on average, as well as forecasting high-water marks in usage that might be expected. This estimate can then be modeled with particular budget numbers for each type of unit cost and a particular time frame. Operations teams provide implementation information, such as how rarely CI/CD services are expected to fail, how long planning phases will be necessary, and how many deployable artifacts organizations expect each week. This information can then be fed into financial models that are designed to generate estimates of budgets and forecasts. Financial models take budgeting in a number of different directions; budgeting should be extensive, covering multiple budget categories, and should be scrutinized closely to keep its numbers as realistic and achievable as possible.

Monitoring the actual financial performance of CI/CD operations against budgets should be done frequently, in accordance with the nature of the operations. Weekly monitoring is advised, as CI/CD operations are typically frenetic, and any unexpected costs should be raised immediately with the operations team. Cost overruns can result from multi-month or multi-quarter cycles during which there is a disparity between the organization’s demand and the services’ supply. For example, total rewards analysis might require several months before full deployment, demand might be highly variable, or some stages have a goal service date but are still active when demand exceeds current CI/CD utilization.

7.1. Budgeting for CI/CD Operations

Budgeting—the process of estimating future needs and making funding requests—is one of the most important functions during the operationalization of CI/CD. Since CI/CD is intrinsically a highly specialized organizational function interacting closely with high-technology asset utilization, budgeting must be done with precision. Due to CI/CD's close interactions with managed systems, its budget should be part of the general annual budget review process, typically prepared by the operations function of an organization. Budget allocation levels requested for any CI/CD function should contain detailed information that demonstrates justifiable growth levels in the following six major areas during yearly planning cycles within increasing business costs due to added operational efficiency and public accountant pressure:

Labor resources and expenses: growing demands for more rapid development cycles circle back, sometimes painfully, to actual productivity and resource utilization. Such issues become further complicated for complex developed business system market sector CI/CD programs that require extensive levels of testing in conjunction with continuous operational utilization.

Rental resources and expenses: as CI/CD becomes integrated within the devOps organizational model, enormous work resource expenditures in terms of facility rental costs assured during operations. Failure to manage or surrender these expenditures for idle-production capacity may force the CI/CD budget to source private-sector consulting at extremely inflated fee structures due to the perceived financial-need situation as the program work survey budget areas become populated.

7.2. Cost Monitoring Tools

Many organizations have begun to use various cloud services and external resources to deploy their products. In general, as you add external tools to speed up building, testing, or deploying your software, your investment in external services may increment significantly. Managing CI/CD costs includes identifying all costs associated with your CI/CD pipelines and optimizing them, including identifying and managing costs associated with the following services: An external CI/CD provider integrated into your workflows can also help you gain visibility into costs. You can view these pipelines and their triggers in the CI/CD service and the potential costs associated with them.

External cloud storage services provide more than just storage. Using external cloud providers to host your container images, static assets, or built binaries can accelerate your service response times, and speed up your CI/CD pipelines, by hosting them in closer proximity to your customers. However, these companies also identify the potential costs involved in accessing the services, such as the egress or download fees. Each cloud provider gives potential downtime monitoring for the services offloaded to each service and periodically accesses them to determine latency and potential long-term service availability. While external services offer more reliable services and fallbacks and higher uptime guarantees, they also charge you for their use and provide you with feedback on excessive costs and on times when your services may degrade the user experience.

7.3. Optimizing Resource Utilization

Cloud services offer on-demand bursts of computing power, but large numbers of those utilized in parallel can lead to high costs. Careful planning for some

items can increase the efficiency of the pipeline. Image builds can be very time-consuming and take a lot of computing power for relatively small changes in commits. As such, it may pay to either build images infrequently and forgo some of the benefits of CI/CD, build only a few images, or find a way to minimize the cost of such large resources. Storing and instantiating large numbers of container images can quickly become expensive and lead to overhead costs. CI/CD infrastructure providers usually offer cost-effective computing types, such as preemptible VMs, and a mix of on-demand and reserved instances that may be a good fit. Softwares help to group operations to run on-demand instances as part of local clusters of preemptible VMs.

The actual implementations of services are also an important aspect. Services configured without using tiering can incur unnecessary costs. A big part of cost management of cloud services is proper tagging of resources used during the CI/CD lifecycle. Resources are then exposed to specialized tooling that can give insight into exactly where costs are incurred, and also provide projections for evolution over time. Proper tagging can also aid in tracking where and when testing is actually done.

8. Best Practices for CI/CD Pipelines

One of the key objectives of CI/CD is to convert potential code delivery bottlenecks into automated checkpoints that force action and feedback asking the essential questions: "What happens if I deploy this change?" and "What happens if I deploy this change today?" Following established best-practices helps DevSecOps teams accelerate code delivery without sacrificing quality and security. Among the best-practices for CI/CD pipelines, we highlight:

8.1. Version Control Integration

Developers use Version Control Systems (VCS) to check in new code. CI/CD pipelines monitor repositories for check-ins, which trigger pipeline actions. A code commit can initiate a sequence of actions (testing, artifact generation, security and performance scanning) that validate the change, irrespective of its file size, ownership, or other attributes, before it is integrated into a master codebase. Version control integration with CI/CD eliminates the wait for code to be pushed to staging and helps development teams get immediate feedback on code quality.

Among the various integrations that allow for the implementation of a CI/CD pipeline, version control system integration assumes the related functions of source code depository, accessibility, tracking, management, and functionalities of integrated development environment for developers to prepare code. It is common that upstream developers store their codes on Interactive Git and primarily use it for collaborative software development and code asset management. As such, the most fundamental pathway for the CI/CD pipeline to process files to be packaged into the containers is collect from the developer's commit history in the upstream branch and read the source codes from the latest commit. Typically, source codes are read from the depository by utilizing webhooks or periodic scans of commit history.

CI/CD tools are available to be integrated with virtually all popular version control systems. These platforms mostly accommodate commands to execute how CI/CD tools should respond to events triggered from the version control repository. As deployment scripts are bundled into CI/CD tools, it is expected that the efficacious management of the associated code in the repository be instrumental in achievement of the purpose in an effective manner. Recommended for the CI/CD repository is the use of tool-specific scripts that need to be kept updated to accommodate the latest product development and deployment changes or any updates or feature upgrades offered by the CI/CD tool vendor. It is also good practice to version control all product-specific settings and check-in, tagged versions upon the product releases. Production systems should at all times be recoverable to any one of the previous releases with tag names to keep the service outage to a minimum. Tags are easy identifiers that are retrievable on all dedicated pipelines which have been especially designated for the purpose.

8.2. Testing Automation

CI/CD pipelines enforce that code goes through a set of automated tests as soon as it is checked in. Development teams use Unit tests to check the low-level correctness of code, Integration tests to ensure that code interacts well with other components, End-to-End tests to validate if the code fulfills user scenarios, Performance tests to probe the system's throughput, latency and availability under stress, and security scans to look for vulnerabilities in code, containers, and infrastructure. Automated testing eliminates releases inadvertently deployed to production without any level of verification.

Automating tests for your web applications is a required step when implementing Continuous Testing; without an automated test suite, Continuous Testing becomes not only impractical, but almost impossible, with rare exceptions (i.e.,

small projects that allow for highly frequent manual testing). Why is this the case? Because when you are continuously committing code, you are also continuously wanting to confirm that such code does not break other areas of your application. If you were to do this manually every time, you would need a lot of shirts to change and would be practically wasting time and resources.

When we talk about "testing quantities" in stateful applications, we are generally talking about the following types of tests: component tests (covering the smallest possible units of a web application), functional tests (executing the code just as a user would, potentially covering end-to-end application routes), regression tests (testing application functionalities for which you already have positive automation tests in place) and negative tests (testing application functionalities with negative scenarios, ensuring that the application does not break).

Let's take a step back for a moment. What is Continuous Testing? If you look for the definition, you will most likely find something along the lines of "Continuous Testing is a software development practice where automated tests are run early and often throughout the CI/CD pipeline". Not bad. I'd say a bit too generic, and not mentioning anything web-development related after all. You'd hope to find a definition that explains how this testing process is uninterruptedly at play when you are working on a web application that is in a state of continuous deployment, as is the case for many web applications available in contemporary times.

8.3. Continuous Feedback Loops

Continuous feedback loops are a fundamental principle of CI/CD and DevOps culture. They enable organizations to receive and act on feedback at every stage of the development lifecycle—from idea inception to usage in production. Creating continuous feedback loops requires an understanding of developer, operator, and end-user needs along the pipeline. Organizations need to rethink the way they collect, communicate, and respond to information, automate where possible, and create an environment where teams have ownership over the solutions and tools in their feedback loops.

With CI/CD, feedback loops are created by instrumentation, monitoring, and alerting, and together they form a self-service infrastructure for toolsets that teams can customize for their use cases. For developer teams, feedback loops focus on understanding whether users are embracing newly released features or enhancements, and how these features are performing across user profiles. For SRE and Operations teams, feedback loops help indicate whether the production system is healthy or unhealthy, and why it is in that state.

However, if a team isn't given the tools or skills to react, feedback loops become too heavy. In extreme cases, they will become zombie loops that add no value. The costs of a poorly planned feedback loop are multiplied when each feedback loop adds its own set of tools that require operations management, maintenance, and resources to keep alive. It is necessary to be aware of the hazards these zombie loops pose, so that you can make conscious decisions about which critical feedback loops to enable and automate. These loops must be customized to meet developer and user needs so that teams invest in them, propelling their growth and evolution. Feedback loops must allow developers to fine-tune their code, operators to respond to high-impact events, and builders to create experiences that delight and amaze.

9. Case Studies

Earlier, we focused on designing and building components needed to allow for CI/CD on Fabric assets. In this section, we will present case studies of previous effort, common pitfalls, successful strategies and systems used to monitor pipeline jobs.

9.1. Successful Implementations

This chapter describes achievements from two years of CI/CD Pipeline operation support for clients hosted in IBM Cloud –aimed mainly at Fortune 500 companies. During this time, we migrated pipelines and their runtimes from legacy platforms to DevOps for IBM Z, delivering associated savings from reduced runtime times. We added new capabilities offering increased speed for integration and delivery from agile pipelines. We provided increased value by developing full-stack DevOps pipelines consuming artifacts from our clients' industrial development with IBM Z, integrating distributed assets from other development teams in synchronized early DevOps pipeline stages, and delivering those assets for client's users via hosting services.

We addressed security and IO constraints, DevSecOps applied to non-development cycles, and monitoring improvements to allow more proactive operations. These topics are detailed further in the lesson learned chapter. The services offered to clients afterwards went through their own CI/CD pipelines to validate all automation, generated documentation, and allegedly upgraded code and configurations, and were released to production initially onto only selected input data. Real alerts were opened only after usage upticks. We leveraged integrate-first and migrate-later strategies to do the internal pipeline migrations,

by adding DevOps for IBM Z capabilities to existing self-hosted IBM Z based CI/CD pipelines in our clients and then migrating them into hosted services.

9.2. Lessons Learned

A few of the many lessons learned during implementations of these monitoring systems are detailed here. Addressing security concerns of inaccessible CI/CD pipelines requires a definition of “digital trust” and a process for achieving and demonstrating it, which will often not be based upon currently existing and accepted practices. Locating CI/CD pipelines entirely within a company’s Firewalls is one solution, but it is often impractical for various reasons, such as a lack of skilled personnel, a lack of facilities, and the financial costs of obtaining and maintaining them. Additionally, performing delegated operations on third-party assets requires an implicit or explicit trust agreement between the asset’s owner and the CI/CD pipeline service provider.

Cryptographic signing with public/private key pairs is the basis for trust. Public key being verified and trusted is the primary difficulty. The classic chain of certificates provided by a trusted Certificate Authority is not practical, generally not applicable, and demonstrably not sufficiently secure for binary software and fabric asset signing. CI/CD asset signing for fabric assets must be based upon private certifying keys for individual organizations and/or companies that want to do business together and possibly under specified conditions.

The matter of building secure versions of each of the components of a CI/CD pipeline for fabric assets is a challenge, especially when transparent source code build communities supporting the various pieces are either non-existent or still maturing. Secure commercial versions also typically do not yet exist. Digital forge-tool components significantly reduce the effort to create secure builds for Open Source assets. However, each tool must still be exercised to ensure adequate security by being configured and operated in a secure manner.

A secure, external developer zone for untrusted developers to test and develop assets is essential for creating a guard on the CI/CD pipeline belonging to the owner of existing fabric assets. Secure, fully functional, private key managers are required by CI/CD pipeline operations. Finally, CI/CD operations must support a process for recovering from failures, because failures are inevitable.

10. Challenges in CI/CD Operationalization

Many challenges in CI/CD can severely degrade the effectiveness and benefits in optimization of business processes, whether functional or non-functional because the potential negative side effects of wrong decisions in CI/CD are superceded only by the potential negative effects of wrong decisions in production in real-time, where full visibility and control on every single production asset and on the whole ecosystem of actors, agents and actions is a must-have [1]. Hence, neglect and mis-grooming of CI/CD, as the entire synthetic monitoring, can cause serious consequences on all the actors and processes involved. We are at risk of a downhill spiral of ever-increasing damage. This section outlines common pitfalls in operationalization of CI/CD and some mitigation strategies. However, we need to remind the reader that the diametral inversion between the goal of a "proactive surrogate model of the operating conditions of in-field assets" and the nature of the CI/CD embeds a design flaw that results in a process that cannot do more than a small optimization effort on the dysfunctional real-time processes. As a consequence, whatever effort for enhancement of CI/CD in this sense, is an added cost. Given these considerations, and the limitations of CI/CD outlined in the following, only a few steps are suggested that can be useful in some cases, and seek to have CI/CD converge onto the real-time observable surrogate models in a reasonably short time, at least for sanity checks of the decisions taken through real-time, spontaneous data. Only some of the modeling methodologies proposed in recent years, or simpler solutions, can be the basis. Exploratory data analysis and simple machine learning techniques may also be possible, but require a skilled workforce to design and inspect interactive exploratory tools for previsible models.

10.1. Common Pitfalls

Regardless of the actual maturity of the CI/CD strategy, it might sometimes happen that many projects inside the Fabric could be standing still: changes are not integrated anymore, builds are not executed anymore, people are not working on the project anymore, etc. There might be multiple reasons for this. Some of these reasons are related to CI/CD in general. Others are specific pitfalls of the practical application of CI/CD within the context of a Fabric. The right question to ask is: How do we know whether CI/CD is really helping us or whether we are only doing CI/CD because it sounds so cool? The rest of this section aims to answer this question by identifying several possible reasons for not getting the expected advantages from CI/CD in a Fabric setting.

First, it is not sufficient to merely build the CI/CD infrastructure; someone needs to keep an eye on it. Unless you want to wake up one day and find yourself stuck on a version of the software of one of your business units because nobody bothered to keep implementing patches and upgrades to problems exposed on the release notes, there must be someone (potentially even more than one person) who is responsible for keeping the CI/CD infrastructure up to date. This involves both fixing broken parts and monitoring the job status: Are jobs finishing successfully? If no, why? Are jobs executed every time someone merges to main? If not, why?

10.2. Mitigation Strategies

How can we mitigate the problems outlined in Section 10.1? Risks in the CI/CD suite and its components can often be perceived as systems or development-table-centric. Providing tools to assess and mediate problems with the assets being delivered can have a beneficial impact on their adoption and understanding. However, this must not lead to services attempting to make CI/CD a programmatic solution and not still a human-driven decision engine. CI/CD provides guidance to help steer decisions by enabling faster execution and verification of lower-chops needed to highlight the explicit problems in the total body of work. Executing on all impulses built up at the various stages of the delivery process is not the answer.

CI/CD intended to reduce the costs of fast feedback loops, however if the feedback is not relevant or reusable, it can become additive and counter-productive as well. Additionally, a simple-legislation and tracking of errors based on types is not the right approach either. As companies grow, and the frequency of each component update reaches a critical point and the types of updates along with their cost offsets. With a certain size of changes across various components and the building of trains, a decision structure involving ask for feedback along with different accept/reject loops along with various punishments in different stages of the pipeline having a micro-managed cost may work in larger organizations and against the mistakes happening along the way but at a high effort cost. This should not become a bureaucratic system that chokes innovation or exploration.

Such metrics should only be used as platforms to help capture error types on blame and to draw overall perspective paths on analysis to help prioritize these types of general issues occurring and have better oversight for the resources for the fixing work. Private analytics and internal review boards are often the most healthy way to capture this type of information and provide feedback without being necessarily reactive.

11. Future Trends in CI/CD and Monitoring

While CI/CD practices have been in place for some years now, thus contributing to more adaptive Industry 4.0 stakeholders, they will not remain stagnant, and no technology adopted in this new Area will become global and static. Several technologies are now emerging that may contribute to the future evolution or supported pipeline of CI/CD. These technologies include MLOps, AIOps, DataOps, Release Orchestration, Test Automation, No Code Development and Citizen Developers, Microservices, Service Mesh, Low Code Development, New Use Cases, and Edge Devices.

MLOps. Creating machine learning models is not as simple as development operations. These models need to be properly prepared, monitored, and retrained throughout their entire lifecycle. Furthermore, these models may impact the outcome of business processes, and care must be taken to ensure they comply with the company's values and law regulations.

AIOps. The use of Artificial Intelligence is currently offering benefits such as the automatic response to anomalies, conversion of incoming events to actionable incidents, the provisioning of root cause analysis based on AI, the prediction of problems leveraging AI, and the offering of a context-aware approach. offer advantages like AI-based anomaly detection, data transformation to knowledge, breaches to breaches, change justification, and prefers automatic diagnosis and prevention. Furthermore, it is expected that more organizations will use multitasking models that make automatic decisions using tiny data.

DataOps. The analytics can be a competitive differentiator for any Organization. Care is needed to ensure its availability, usability, and security. It is expected that more Organizations will adopt DataOps solutions to govern and automate their data management processes.

Release Orchestration. While Release Management has been in place for sometime now, Release Orchestration is a subset of Release Management. Rather than focus on policy, auditing, and governance, Orchestration focuses on automation. It coordinates the automated process across the entire Software Development Lifecycle, which ensures that the process is followed at all times and that nothing is missed.

11.1. Emerging Technologies

Continuous integration/continuous delivery (CI/CD) emerged in the software devops domain from Agile principles. They have been applied especially as

companies have sought to release product iterations faster and with better quality verification processes. As the DevOps movement matured, and the same principles were extended into adjacent areas, the CI/CD and related observability and monitoring technologies also advanced in capabilities to cover code, infrastructure, and platform services of web and mobile applications, whether on prem or cloud based. Continuous integration and continuous delivery (CI/CD) enables developers to enable frequent code and infrastructure changes in a rapid-feedback cycle and monitoring technologies then help enable fast issue resolution. A number of higher-order principles govern both the execution of CI/CD processes, and the observability requirements that then enable monitoring and alerting processes to support the pipeline.

The principles of CI/CD are diverse and cover areas such as: Developer, run trigger, speed, agentless orchestration and security, outcomes, templates to avoid lock-in, extensibility with third-party integrations, advanced workflows, pricing, compatibility, and in the case of observability tools, data integration, value, insight context, customizability, user experience and design functionalities, data types, advanced capabilities and partnership ecosystem, and proximity to partners and open-source projects. The principles of observability at the other end of the CI/CD pipeline are: Visibility to signal, alerting, ease of use, multi-environment support, responsiveness, collaboration, security and compliance, configuration based observability, incident management, advanced capabilities, partnership ecosystem, and proximity to partners and open-source projects. As DevOps matures and enters the mainstream, increasing enterprises are looking to CI/CD and Monitoring tools with AI, Security, and Cloud among the key trends shaping CI/CD and Monitoring in the near future.

11.2. Predictions for the Industry

The future of CI/CD is bright, and it sways toward automation and simplifications using emerging technology. In a year, most tech companies will have deployed security along their CI/CD pipeline addressing a huge issue that was usually forgotten, leaving a hole for hackers and malicious attackers to work their way into sensitive infrastructures. These CI/CD solutions will be provided as services removing the complexity from companies having to deploy and maintain a community version of CI/CD and to keep it updated. Edge-computing as a service will be offered by the major cloud players increasing the availability of some low-latency needed for special use cases. We are still years away from a high adoption of quantum computing, and we do not feel that this new technology will change the IT infrastructure at a large scale. Third-party applications will change the way of integrating products with the native CI/CD tools offered by the cloud

environment, and the pipelines will become smaller, dedicated, and easy to consume from integrations.

The number of tools will shrink down to a few that will be the “standards” for CI, building, and deploying software components in the cloud, on-prem, or hybrid. Native tools will take over bubble programs that can execute only in a limited number of cloud environments. They will also be adopted outside the initial space they were developed for. The improvement made by some companies toward simplification of the way coding, and infrastructure-as-code, and configuration of applications will lead to a major improvement in the way code is maintained and onboards DevOps engineers. This will also narrow the gap of skills mentioned above. DevSecOps will become a critical practice widely known and evaluated during the hiring process.

12. Conclusion

The acceleration of digital transformation in financial markets has driven the need for technological solutions capable of enabling enterprises to boost their growth in increasingly contested and drained commercial sectors. As a result of this technological moment, established institutions have been pressed to innovate their business models, shifting from the service economy to transactional relationships enabled by the enterprise services provided by eligible digital factories. Emerging decentralized finance platforms based on open source technology invite institutions and service firms to use the underlying permissioned ledgers to build business services oriented to small, medium, and micro enterprises.

Permissioned ledgers simplify the definition of access rights to the asset tokens managed by the smart contracts deployed on the technology layer hosted in the ledger technical environment. The ledgers shift the complexity of digitally addressing the asset tokens to the logic layer implemented in the enterprise services, exposing the inbound and outbound messaging interfaces focused on the onboarding, offboarding, as well as the related transactionization processes. Defining the inbound and outbound interfaces is paramount. Their correct operationalization ensures that enterprise service risk management has visibility over key risk indicators associated with every enterprise service, especially those related to the risk of failing to deliver in the case of external capacity shortage or service disruptions. The verification of compliance of the transactions to the enterprise policy is discussed openly both in forum meetings and in the meeting,

where decisions are traced to make visible accountability for any eventual noncompliance event.

References:

[1] S. P. Panda, Relational, NoSQL, and Artificial Intelligence-Integrated Database Architectures: Foundations, Cloud Platforms, and Regulatory-Compliant Systems. Deep Science Publishing, 2025. doi: 10.70593/978-93-7185-129-9.

Deep Science Publishing, 2025
https://doi.org/

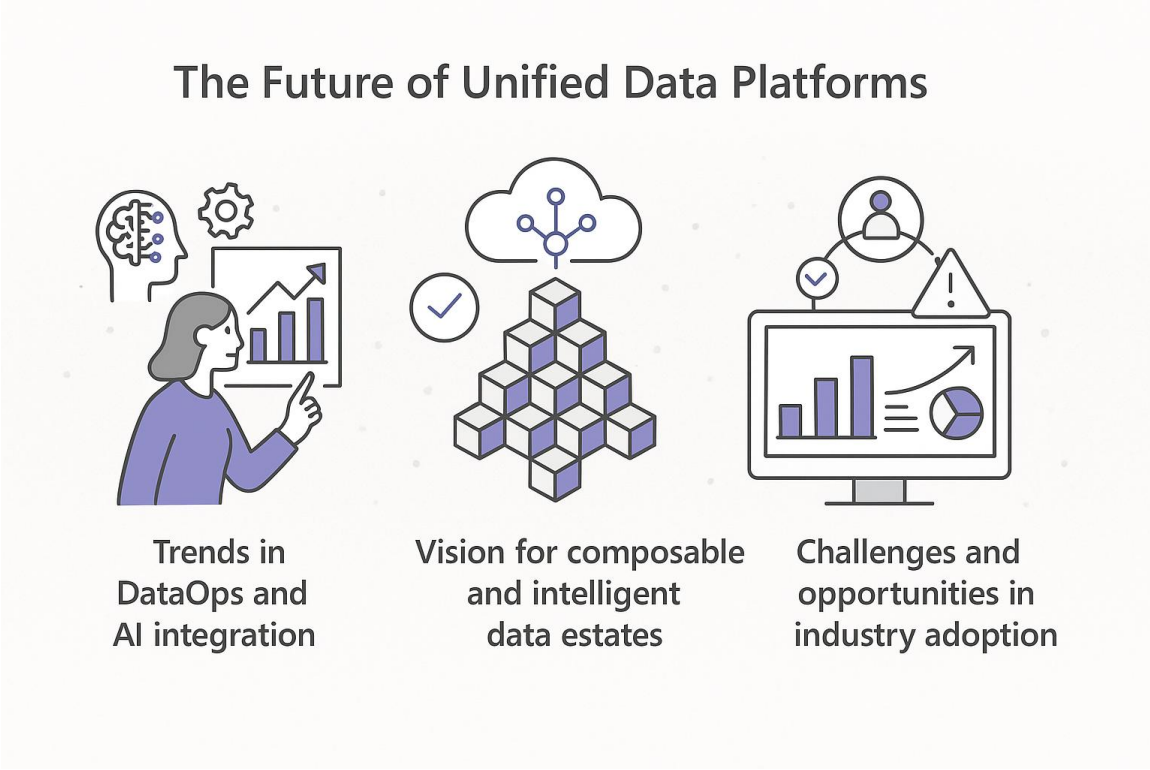
DeepScience
Open Access Books

Chapter 10: The Future of Unified Data Platforms

1. Introduction to Unified Data Platforms

The term "Unified Data Platform" is used to refer to a set of technologies that deliver the capabilities needed to support and optimize end-to-end analytics and AI/ML workflows, including data integration and the building and management of downstream pipelines. A Unified Data Platform means, at the very least, the seamless integration of the following functions: Data Ingestion, Data Storage, Data Preparation, Data Pipelines, Data Collaboration.

The need for a "Unified Data Platform" has arisen because the creation, delivery, and consumption of analytics, whether through dashboards or embedded analytics in applications, is the most common way to "operationalize" Data Workflows. First popularized by "Business Intelligence," advanced analytics via SaaS, coupled with the increasing operationalization of AI/ML, provide users of all types with increased accessibility to data. Enhanced accessibility, in turn, has created the need for greater efficiency and collaboration, accelerated time to insight, eliminated silos, lessened dependence on central IT, increased quality and sharing of data, and improvement in the democratization of Data Workflows and of the platforms that support them. By unifying Data Workflows, Unified Data Platforms allow the consumers of insights to work collaboratively with the creators, plus the citizen-powered creators to work together with central IT, with teams able to share insight pipelines and data sources, making it easier to automate and operationalize repetitive tasks.



2. Trends in DataOps

As DataOps matures, its focus shifts from tools to practices. Forward-thinking DataOps practitioners ensure best practices for managing their data are widely adopted throughout the organization. Others stand-ready to share their knowledge to help DataOps achieve the desired benefits.

As organizations realize the challenges inherent in standing up a DataOps practice, the marketplace expects integrated solutions that help address the complexities associated with implementation and enable those teams to achieve their goals faster. In parallel, organizations are applying agile methods to data to ensure that the highest-impact data sets and pipelines are the ones that are continuously implemented and updated with new data and algorithms. As a result, the category is maturing from siloed capabilities to unified data platforms that help DataOps deliver on its promise of providing trusted data in a continuous and automated manner.

2.1. Evolution of DataOps Practices

DataOps uses practices that have been successful within DevOps, extending continuous integration, continuous delivery, and continuous deployment concepts from code as data to today's data development environment. DataOps recognizes that data development is not limited to ingesting, cleansing, enriching, securing, and maintaining the data, as is traditional within data engineering. The practices of data wranglers, data scientists, and present-day machine learning operations professionals are also part of development, continually preparing data for unique and irregular experimentation by end users, often business analysts or developers, to create any kind of data product.

The QA and test industries are also part of the data operation environment, developing tests for data production that help to ensure the usability and quality of the data from a variety of perspectives. Furthermore, the data operation environment must also consider executing tests specific to the data product and data pipeline that has been developed, ensuring the production data operation correctly delivers the data required by the business at that time. DataOps extends these present activities to embrace and operationalize on-demand data product and pipeline development so that the operation of the data pipelines and service levels associated with their use can be managed properly by data governance and data quality teams. A key focus is on implementing repeatable processes for managing ad-hoc data products and automating the development through to deployment for use operationally in production systems.

2.2. Impact of Automation on DataOps

Traditional data operations processes continue to be plagued by an environment where tasks are executed manually, tuning, rebuilding, and handovers are commonplace. This is particularly evident in pipelines with multiple handovers across data producers and data consumers. Data pipelines are brittle because they are susceptible to all kinds of disruptions and most failures are not detected until data is used by business units which may be days after the data engineering team has delivered the data.

Perhaps, data availability, data quality, and DataOps cost can all improve with effective automation. Automating DataOps involves ensuring different DataOps tasks across the data pipeline are managed easily and seamless handovers are in place. DataOps tools and DataOps as a service bring automation capabilities to the data supply chain. DataOps tools automate reproducing, refactoring, and repairing tasks. Monitoring, testing, and validation tools operate for controlling, auditing, and certifying tasks. Collaboration tools work for communicating,

documenting, and reviewing tasks. Overview and orchestration tools will monitor and manage different DataOps tools managing different tools. The overlap and integration between the three environments will continue to grow.

The rise of different DataOps tools and DataOps as a service offerings have empowered data teams with capabilities to develop more sophisticated data products. DataOps tools reside across the three DataOps environments, bringing different automation capabilities to these environments. Collaboration and communication enable different teams to work on the data through visualizing changes made by different team members. DataOps tools automate and simplify important tasks for delivering quality data products faster. Fewer manual functions will offload the repetitive work. As a result, you'll have less "boring" work and more time to focus on analysing data and delivering actionable insight for your customers.

2.3. Integration with Agile Methodologies

The explosion of big data across organizations has forced them to redefine their business priorities, and therefore, adapt their end-user technology and related processes. As a strong component of the new big data landscape, DataOps has profusely emerged in the last decade as a specialized and agile design, operation and oversight discipline for big data platforms. Some roadblocks to DataOps are its battles with traditional IT and skill shortages. Responsibility for building and maintaining big data services has fallen upon small, highly-specialized teams within IT. However, the APIs and services exposed on the big data platforms are designed for use in the traditional enterprise systems, and mismatches can lead to difficulties and delays. A new generation of specialized providers is emerging on the DataOps landscape who can both innovate quickly and ease the burden on customers. Regulatory compliance is the biggest and most immediate worry for these enterprises.

If DevOps is symbiotic with Agile Software Development, DataOps is symbiotic with Agile business agility. Business-IT relationships might have broken down in traditional IT organizations, but in the new world of digital techniques, both arguments in favor of and windows for alignment have never been stronger than today. Business leaders may argue that the costs associated with digital initiatives are too high or that they are not aligned to business strategy, but such arguments can be countered with examples of lost opportunities associated with being 'left behind'. These initiatives are normally time-stamped – 'do it now, or do not do it'. Senior business executives set the digital clock running and agree that they alone should be accountable for delivering the associated benefits. For operating in this Agile environment businesses, IT must provide responsive, flexible,

complimentary services which support the willingness of business units to budget for developing their own front-end applications. The data that underpins these services is an enterprise asset, and one that requires planning.

3. AI Integration in Data Platforms

Enterprises previously depended on business intelligence for analytics leadership and its role in data integration and organization, nonetheless, during the last decade this role shifted to data platforms such as Data Lakes and Data Warehouses leveraging massively parallel processing data management engines. These tools enable the automatic generation of data schema and syntax analysis for data distribution. Business Intelligence products centered data insights mostly on the multi-dimensional analysis of historical data, however, the advent of AI Systems has changed the role of Business Intelligence and it is now focusing on real-time prediction rather than only descriptive reporting on historical events. Predictions leverage the expertise invested on building AI models that use features extracted from historical data analysis and allows a more precise strategic decision since ongoing data related to current events flows into the predictions in real-time.

The consolidation of a predictive Business Intelligence also changes the amount and period of trust of the available data. In this case, it's not wise to consider available data as final reference, rather to be treated as a work in progress process. When data enters into the platform it gutters into two gutters: one is feeding the Business Intelligent model used for predictions and the other is discarded. Whenever the prediction accuracy is considered to be out of rationale for the impact of the supported decision, the Business Intelligence alert model examines the discarded data against the event for which predictive analysis was done. When sufficient confidence is reached on the new data, a new data pipeline is generated.

3.1. Role of AI in Data Management

Data is the fuel in AI initiatives and that is the reason that the synergy between AI and data has long been present. In the case of integrated platforms too, data that is managed through integrated platforms is a crucial component for building AI predictions with high accuracy, thus being used as input feature data. However, it is difficult to repeatedly and routinely build high precision machine learning models which are the core components of AI applications due to issues of maintaining the quality of this management of input training data. This is

because the data is constantly updated and the dimension of input data has a risk of increasing dramatically, thereby causing increased processing costs. Weaknesses in the machine learning models for high accuracy predictions include the model being unable to respond to changes dynamically and the model being unable to properly predict outliers. For these reasons, integrated platforms are required to manage the four key areas for AI, which include the quality of the training data, the data for testing models, the dimensionality of feature data, and the input data for predictions with user defined constraints.

To this end, we make it possible to integrate data with various raw formats as input for more sophisticated data preparation. This means enabling the new synthesis of time series data, economic factors associated with company data, and sentiment factors associated with news data. The consolidated data is then classified through a service, which provides a machine learning model calibration function and hints at precision improvement through the optimization method. In addition, exams are conducted for local AI models, followed by deployment with respect to user defined constraints such as prediction tolerances and model latency. The AI engines for solving issues such as high uncertainty and high dimension are reinforced through the processes of continual improvement based on integrated data management models. Thus, we provide a true unified platform that enables true AI predictions through integrated data management.

3.2. Machine Learning for Data Quality

Data quality problems impact organizations daily, every hour, every minute, and every second. Data quality has become a major initiative in organizations around the world, and maintaining data quality has become a major task in the day-to-day data management of organizations. Data platform technology has prioritized building and delivering a well-governed, authenticated, reliable, and quality-driven occurrence of data, centralized for consumption, and intuitive to enhance AI potential. These attributes allow organizations to build trust in data and trust in the data systems. Every organization must invest in trust, they cannot afford to invest in anything else. Unified Data Platforms help organizations invest in trust and do it efficiently.

Many pioneering organizations are solving such problems using Machine Learning. These implementations mainly cover table discovery, schema inference, schema validation, record linkage, search, duplicate detection, type inference, labeled data generation, system tuning for user preference learning, and database query completion. Machine Learning solvers have been deployed in various companies. Data quality management is a key aspect of modern information systems, intending at detecting and correcting data errors to ensure

high data quality standards. Random-forest is widely regarded as one of the most accurate classifiers for discrete data streams due to its ecosystem of ensemble classifiers and relative user-friendliness. Decision Trees are preferred over other classifiers because they do not require intensive effort for parameter tuning and feature scaling.

3.3. Predictive Analytics in Unified Platforms

As data operations continue to grow and new classes of insights are unlocked more frequently, predictive analytics will only continue to grow in adoption. Unified data platforms will clearly play a key role in this predictive data future with their operational ease of use and direct data integration for rapid results. Companies should think about how they can extend these use cases for their customers versus others who may be solely focused on making machine learning easier to develop workflows.

Self-service business intelligence has been around for many years. The transition from static reports to dashboards has made it easier to provide teams across an organization with customizable data analytics experiences. But many users are still not comfortable with tools that sometimes require data prep efforts to set up. Data management companies began to fill this gap in recent years with products aimed directly at citizen data analysts.

Natural language processing continues to make exploration via searchable dashboards more common in modern business intelligence platforms. Further improvement with NLP, plus leveraging capabilities in finding patterns and calculating outcomes across a variety of datasets, will only make delivery of insights in this way more accessible. The next logical step from business intelligence is predictive analytics. Data integration and management have always been crucial in providing users with analytics insights. Predictive options are finally starting to become more common for business uses. But they traditionally happen through specialized vendors with a narrow focus.

Expanding predictive capabilities into mainstream uses will be a strong growth strategy for companies focused on democratizing data in workflows for the widest possible audience. Time series forecasting has offered some of the most direct use cases in business, especially retail where demand forecasting has been a traditional workflow. Focusing on SMBs to allow them low-code, good-enough options to help them economically and operationally can be a strong growth area for investors.

4. Vision for Composable Data Estates

More than just the hub for connecting silos and enforcing security and governance, unified data platforms also provide a suite of services for transforming and enriching data to solve many problems in the area of analytics and AI. A platform does much of the heavy lifting for users and enables self-service and collaborative development of pipelines and models. Enterprise needs continue to diverge. Different businesses want different kinds of specialization, and different departments within a business want different services, tuned to their own needs. This is something that a single vendor platform cannot do well. Moreover, solutions to particular problems require deep knowledge in that area, a deep surge of innovation, and deep hardening. But the cost of specialized solutions needs to be more than compensated by the benefit. Skills and delays related to implementation and customization should be low, and maintenance effort should preferably be shared with the vendor. So, we see a future of composable data estates – a palette of specialized tools, which can be connected and orchestrated to provide more general-purpose capabilities of enterprise data platform solutions, sometimes with complex requirements across departments. We will also see deep integration of autoscaling, code validation and security, and automated optimization into these specialized components, relieving the enterprise from low-value and repetitive tasks, and allowing the vendor to differentiate based on product performance and usability – the degree to which more complicated solutions come together without hassle. The solutions come together as either a layer of pre-built apps, or of code that can be accessed through APIs.

4.1. Definition and Importance of Composability

Composability refers to a design characteristic of systems to be assembled and disassembled at their core building blocks — modules, tiles, patterns — like digital Legos, supporting an easy, rapid, and cost-effective assembly of more complex and specific services and solutions. Digital composability is a way to alleviate enterprise complexity, allowing to better visualize and coordinate different areas of the business, regionally or globally exploring and building new capabilities across them. The use of composability principles continually pushes us toward environments that are wide, shallow, and distributed where Lego-like blocks can be assembled quickly to create an infinite variety of combinations, and which require a different security model; a passage to a creative effort by individual lines of business and groups that is approved, coordinated, and checked by IT based on established architectural principles, standards, and systems; and a higher frequency of collaboration and discussion among business

units, often involving operational practices that are irregular and inconsistent by the short-term nature of business, business application, and service needs but ultimately create competitive advantage.

4.2. Architectural Framework for Composable Data

Management of data infrastructure has often become a bottleneck for businesses and data teams because of its complexity and highly heterogeneous tools available for solving specific problems with data. In order for data infrastructure to be composable, there are several layers of architecture that allow different businesses and teams of different sizes to use what is appropriate for them in a seamless manner. These architectural layers enable an almost fully automatable building block solution to companies in building and scaling their data and analytics process. The Infrastructure and Governance layer mainly consists of the plumbing or the building materials, that will allow businesses to build their data estates on demand, scaling resources as needed. The security and governance required are already built-in, and only need to be configured for that business.

The Business Metadata and Tools layer allows builders in business units a way to express a need in the common tools they are comfortable working in. The Builders can ask for what they need through common business tools that plug into the Business Semantic layer, which are automatically validated and translated into Data platform objects required to fulfill needs. The Translatable Layer translates inputs from Business metadata and Integratable tools layer into Abstract Data Objects that are pluggable into Data Platforms. These objects are built on common data models that add a common language layer to Dataverse which is the final layer that is the comprehensive data semantic system that lives both within your data services but also aligned with business language.

4.3. Benefits of Composable Data Solutions

Today's data solutions are traditionally architectured as monoliths: centralized, high-capacity systems that serve the entire organization. Functions can be bolted onto monolithic foundations and scale to accommodate large amounts of data and concurrency. But in our fast-paced world, change is difficult, and the real needs of end users can go unmet. Business units bypass their traditional vendors-acquiring their own data tools and creating an ecosystem of point products. Meanwhile, major enterprise vendors have steadily widened their applications scope so that they command the most important and lucrative data top layer while using third-party vendors beneath them for fulfillment.

Composable data is the answer to our binary world of monolithic data warehouses and siloed data marts. Companies who process wide varieties and

volumes of data with specialized tools need a more flexible and powerful data approach. A company's unique business goals determine its unique data touch points, technology mix, and sourcing and talent strategies. For composable data solutions, cloud services feverishly fight to deliver the best individual capabilities. Data management is distributed ecosystem design: select (or build) the best point products for ingestor gathering and hydrating data; storage; processing; integrating and enriching; modeling; protecting; securing; delivering; and monitoring.

Unlike the gargantuan engines of traditional providers, composable data services are light and swift. Thirsty for innovation, software companies pile on with new flavors of every data function. Interoperability is critical: composability avoids building a platform of interlocked modules, which can create vendor lock in – increasing costs and making changes more difficult. As your data scale and needs change, composability provides the luxury of deleting, adding, and moving data services around at speed.

5. Challenges in Industry Adoption

While the benefits of Unified Data Platforms (UDP) are apparent, their adoption is complicated by several—often contradictory—business and technical challenges. These mainly involve choosing between a cloud and an on-prem deployment model, integrating with existing tools and technology stacks, avoiding performance penalties incurred with scheduling and orchestration, effectively handling complex user- and organization-centric governance models, ensuring security for the enterprise data estate, resolving conflict with business intelligence tools, and modelling system-level data flows. UDPs also require a richer understanding of enterprise requirements, new operating models for enterprise data teams, very tightly defined hybrid or multi-cloud architectural choices, work management models and integration to work management systems, and increasingly, clear models for usage-based pricing related to various user profiles.

5.1. Data Silos and Integration Issues

Data Silos and Integration Issues Most large organizations have built up massive data volumes over several years. Historically, this has happened in largely functional silos within organizations. Data on sales processes live within a different system and software ecosystem than the data on customer support. Understanding how to connect data across these systems in a manner that retains

business sense and insight and that integrates business context, including that provided by the other systems of record, is an important first step for successful decisioning. Often, data connectivity suffers from either bandwidth issues, which entail the transmission of large volumes of data, or modelling issues, where the point-in-time snapshots available within these data silos do not provide the ongoing contextual signal needed for optimal decision-making.

To an increasing and overwhelming extent, organizations worldwide are relying upon the power of machine learning and AI to enable the actions and decisions that drive both operational efficiency and business value. Companies have already begun to realize the incredible potential of leveraging the bounty of data they have collected to integrate into their processes to help them make better predictions and ultimately achieve better results. However, for most of these companies, they are currently utilizing the smallest fraction of their available data resources. In fact, research shows that even the largest companies currently only deploy machine learning and AI for a small number of business functions, indicating that organizations are still not adopting these technologies as quickly as expected.

There are many obstacles to the widespread deployment of machine learning and AI, and this challenge isn't simply limited to making the modelling and servicing of these technologies more automated or simplified. In fact, several hurdles still remain that are currently impeding the continued acceleration of commercial industry adoption. Topping the list is the multi-faceted challenge of effective data preparation, which encompasses the integration and transformation of disparate sources of information into cleansed, consistent, and structured datasets. The need for organizations to ingest, process, and unify vast amounts of heterogeneous formats, both structured and unstructured throughout the enterprise from a myriad of different sources on an ongoing basis represents an uphill battle within itself. The addition of incorporating more external data and looking at data as a shared resource adds even more footholds to climb. Today, the challenges of data preparation fall disproportionately upon the organizations' IT and analytics groups, requiring what are termed data pipelines to be developed for the different types of analyses requested throughout their organization.

5.2. Cultural Resistance to Change

Cultural Resistance to Change The adoption of Unified Data Platforms is challenged by questions of culture and business impact within organizations. More specifically, organizations are generally loath to make radical changes to business processes that materially impact employee productivity and customer experience.

Despite growing interest in advancements in data technologies, user organizations often appear much slower to adopt new technologies. This is due not only to detail-oriented differences in the technologies themselves or costs involved but, more fundamentally, to entrenched social, behavioural, and even political considerations in the user organization. One of the most often mentioned of these considerations is that of "cultural resistance to change." User organizations have been heavily invested in proprietary data architectures and management solutions. They have made substantial sunk capital investments in the hard and soft data infrastructure associated with these solutions. Usually, such commitment is associated with substantial training of employees in the use of these proprietary systems and procedures. The result is that key employees of the organization become adept at using the technological solutions involved. Over time, job functions become tied into these systems and the sunk investment in the proprietary solution is increased still further.

If change could be fully made to occur instantaneously, resistance would be low. However, business users of data management and business intelligence solutions are actively involved in processes and functions that have their own natural timing and job-related rhythms. Over periods of weeks and months, it would be obnoxious and hugely expensive to have these business functions shut down while data management and BI solutions were being changed. Hence, organizations become reluctant to undergo the kind of turmoil such changes would require. Not only would productivity declines be both annoying and costly, but these declines would seem to be caused by external forces and would not therefore be closely controllable by the organization.

5.3. Compliance and Regulatory Challenges

Compliance and Regulatory Challenges The regulation of financial services has historically required last mile solutions or reporting solutions to be regulated with a hard barrier semantically and physically. This can make operating Unified Data Platforms for FSI challenging.

Data privacy regulations are a fact of life in today’s digitized world. Laws impose strict rules on how organizations collect, store, and use sensitive data. For many firms, this means having to monitor hundreds of regulations from a multitude of oversight bodies across many countries and states. While these regulations impose unique challenges on businesses across a number of industries, the impact on data integration processes within an organization is particularly significant.

Regulations create specific bottlenecks for the movement of certain data sets, which presents a challenge for data integration and transformation pipelines.

Furthermore, rules that dictate how the integrated data can be used, and by whom, create challenges in deriving value from the data, in terms of analytics and data monetization, to name a few. Organizations that want to share data with third-party companies, or even other divisions within the same organization, may face hurdles getting data across borders if they cannot ensure that it will be properly managed and protected. This creates barriers to localized data management by platforms, as some use cases for large-scale data integration pipelines is to facilitate centralized management of decentralized data stores supporting different compliance needs. These use cases subsequently drive demand for a two-tier data platform strategy outside the cloud.

6. Opportunities in Unified Data Platforms

Few aspects of business today are as vital as sensible analysis of organization data – if organizations are to maintain or improve their market leadership positions, they must get every scrap of useful information from their data and turn that data into useful information. Indeed, the way in which organizations process and interpret the data at their disposal often marks the line between failure and success. Sometimes, new technologies emerge to lure organizations with the promise of amazing payoffs in speed, decision-making capability, and new analytic insights. In this and following sections, we present three such opportunities: new technologies and innovations that are enabling more diverse workloads on Unified Data Platforms, new business demands for enhanced decision-making capabilities, and the scalability and flexibility of Unified Data Platforms to accommodate new data management technologies and advances, and thus to support a wider range of customers in terms of size and stage of technical expertise.

The demand for enhanced decision-making capabilities is coming not only from the C-suite but also from managers throughout organizations. These users want data that they can understand, and user-friendly, interactive analysis tools that they can apply without assistance from data scientists. In other words, they want business intelligence tools that allow them to make sense of the data that they are swimming in. In addition, they want such tools to cover more than just a specific analysis area and more than just corporate reporting of historical data. They want such tools to be the cornerstones for new collaborative analytic environments that they can use to generate custom reports with whatever data they choose to apply to their search, and to deliver the reports in near-real-time mode, rather than

waiting for library managers to replicate and publish reports in three weeks' time using only established corporate data sources.

6.1. Emerging Technologies and Innovations

No organization will hold the monopoly of innovation forever. However, innovation is, by definition, unpredictable and uneven. Numerous technologies are emerging, on the fringes of acceptance within custom development project budgets, which when considered together could become much more than the sum of their parts. These underlying technologies increasingly tie together to support integrated infrastructure/middleware/platform software that will hasten the pace of more unified data structures for analytics in the cloud.

Cloud computing is but one of those driving forces. Dedicated, integrated public cloud services, scalable for big data workloads, are only now coming into their prime. As those services begin to realize the promise of a pay-for-use model that cuts costs and complexity around data warehouses, enterprise applications, integrated development environments, and other costly resources, organizations will look to new innovations by these external solutions for internalizing. Cloud is much cheaper in the short term, but much less controllable and audit-friendly. Organizations will start to view public cloud services as yet a new layer of distributed computing for online, semi-structured analytics, but one with a lot of transactional features built on years of e-commerce experience. The very features that have led organizations to overload enterprise data warehouses were supposed to be avoided in distributed, online service architectures that were simpler, cheaper, and less liable to abuse. With careful control, perhaps these very features can succeed in both.

They will then put more of their enterprise applications and business processes in the cloud. More enterprise data will become subject to externalized, public if not public-or-private, shared-service big data analytic budgets for what are now internally huge and inefficient misuses of data warehouse services. That will lead to greater opportunities for outsourcing the ETL of relatively simple and repeated ad hoc analysis of data that demand more structuring, interactivity, and derivation from the core services deemed the focus of internal specialization.

6.2. Enhanced Decision-Making Capabilities

Data is a key part of today's organizations. It is the basis of their operations and influences their decision-making processes. It helps to include strategies and projects that align with the organization's objectives. As well, it is a source of new avenues of exploration to innovate and improve products, services, and operations. In this sense, decision-making has been driven mainly by analysing

and interpreting historical data. The introduction of Business Intelligence and Analytics systems for data visualization and trend forecasting has contributed to facilitating the task of decision makers. However, as we enter the era of big data, organizations have a mountain of data, structured and unstructured, internal and external, that can help them better understand their operations in their various dimensions. It can help them detect, predict and prevent problems proactively and quickly respond to changes in customer needs. Currently, many organizations are aware of the potential benefits of using data and have been investing considerable financial resources. Integrating all their data into a single unified data platform and leveraging data science and machine learning capabilities and no-code or low-code tools for self-service go beyond analysing and interpreting historical data. It allows providing predictive, prescriptive, and contextualized data, enabling augmented decision-making capabilities.

Therefore, the future of data management should allow organizations to provide all their users with the appropriate recommendations to support their decisions at the right time; whether they are automated decisions established by the organization or only recommendations based on a set of decision rules. Organizations will have to leverage the decisions made by the unified data platform with the decisions of the decision makers to operate in constant co-evolution and keep adapting to their dynamic reality.

6.3. Scalability and Flexibility in Data Management

As companies grow and define new market opportunities, their needs change. The data needs that may have worked for them 5 years ago may not fit their needs today. For example, you may have moved from creating a few dashboards and visualizations that look back to seeing detailed KPI analysis and predictions of future growth. As these needs evolve, a need for increased interactivity and improved data literacy across thousands of company users creates a need for more sales data to be stored. At this point, companies may want an open, cloud native data warehouse that allows users across various departments to own their data and visualize it in the way they want.

People, however, don’t just want to see past data in sales dashboards. They care about what will happen tomorrow. They want better predictive capabilities that can predict lost opportunities and lead to better retention. This increased focus on predicting demand and understanding market shifts requires a more fluid connection between predictive and data analytics. Often this requires companies to have both visual/data tools that generate statistically and mathematically sound predictions, along with clear and actionable KPI dashboards to see the past and the present. The closing of demand, though, is just one small piece of the pie.

After servicing a customer, companies need apps that can take the input of the current customer quotes and make smart and intelligent proposals that the customer wants at the lowest costs. An intelligent proposal generates various proposal examples and then can use insightful pricing algorithms to continuously learn and suggest the price that would drive the proposal order and maximize profits. So it becomes requisitely important to connect these proposals back to the sales data loud and clear and visualize that data so that the right pricing algorithms and request posts are used to profitably fulfil customer requests.

7. Future Directions and Trends

The Future of Unified Data Platforms is cloud-centric and fast moving – driven by requirement for truly unified platforms, a spurt of innovation by cloud service providers and the delight of services and security baked into the platform by default. These trends, supported by a few other factors will also be instrumental in shaping the future of the Unified Data Platforms market.

The biggest driver for change in Unified Data Platforms is unlikely to be from the customers using the Unified Data Platforms. The driving force will come from Cloud Vendors which will choose to innovate quickly in delivering data storage and analytics services combined. Their speed of innovation will benefit all Vendor Organizations in the Unified Data Platforms space since it will help them deliver on the promise of Unified Data Platforms. In short they will work towards making UDA a reality of Today rather than a Dream of Tomorrow. For those who operate primarily in the cloud area, we can hope to see an explosion of specialized services that can address specific needs in managing Data and Information Products at scale.

The democratization of data will drive the demand for Unified Data Platforms, making it a compelling proposition for Enterprises. Improved Services and Security from the Cloud will be the biggest reason for the popularity of UDA and the subsequent growth of the market. We envision a data economy that creates synergy between Enterprises pursuing information as a product and Vendors focusing on building the backbone for enabling these visionaries. The Platform itself will continuously evolve with the innovation from the Cloud Services reaching its customers through the Unified Data Platforms. This market will create possibility for new exciting roles in a enterprise data ecosystem.

7.1. The Role of Cloud Computing

Cloud computing has played a crucial intermediate role in the development of unified data platforms. Its core characteristics include on-demand access to a full range of computing capabilities, which can be scaled up or down, with pricing only for what is used. This change in resources supply has essentially eliminated the difficulty of handling large-scale data loads. Moreover, through provided services and capabilities, it has been possible to combine different technological stacks and products, often from different vendors, in a cohesive manner in a big data environment exposing high levels of complexity. While a number of these services offered a focus on ease of usability with reduced need for technical capability, the overall ecosystem created in combination with existing technology solutions and products has allowed for the democratization of data processing, enabling a wider range of users to exploit previously hard-to-handle and derive insights from large volumes of data. At the same time, the set of capabilities exposed via cloud platforms has moved recently through focus on operational support for traditional on-premises workloads, to the speed of development, cut down time to market and ease of operationalization of solutions using emerging AI/ML models. This change is linked to the restructuring of technological stacks enabling integrated unified data platform solutions that are openly embraced and provided cloud native by a number of vendors. This chapter takes a look at potential current and future areas of development of these unified data platforms noting their links to the available broader cloud environment. Unified data platforms have been invented to support simplified, quicker, and cost-effective development solutions, freeing up data scientists from manual activities and paperwork, allowing data to be focused on higher-level decisions and creativity aspects rather than low-level operational tasks.

7.2. Data Democratization and Accessibility

The future of Unified Data Platforms is a future of data democratization. For centuries, data has taken many forms, from physical storage on paper or a record, to stacks and stacks of bound documents, each with many pages of information, to massive storage arrays that span floors of buildings. As long as data storage needed to be physical, access was limited to those with the means and capability to develop access. With the advent of the internet and the worldwide web, the speed at which it is moving from physical to digital is accelerating. Regardless of speed, is data's journey to democratization fast enough? One user uploaded 350 million photos in three months – an amazing feat. Yet, these 350 million photos represent only a fraction of the ten trillion photos and video clips forecasted for 2022. Data still remains locked away in individual devices: phones, tablets, personal computers.

Despite the variety of collection sources, data still remains unobtainable by the general public. Unlocked data is still the province of data scientists and IT specialists. While novel tools democratize access using artificial intelligence to identify and auto-tag information and facilitate access, the vast majority of tools for seeking out and identifying data still require programming expertise. Innovative tools monitor enterprise systems to detect security anomalies, but provide visibility without intuitive access to the data itself. The future of Unified Data Platforms is a future of data democratization – of simpler, intuitive tools that enable greater participation in data discovery and understanding. The focus will be on user-centred design that enables conditional data access beyond the walls of individual organizations and system integration with standard formats and protocols.

7.3. Integration of IoT and Edge Computing

As the proliferation of connected devices continues, the sheer volume of data being generated is growing at an exponential rate. For Unified Data Platforms, the integration of IoT and Edge is a critical step to be able to readily ingest and process the massive data streams from these devices. In particular, there is a large amount of unstructured data at REST and at Motion that requires sophisticated and purpose-built platform components to efficiently parse and process this data.

Unstructured data that is at REST includes such things as images taken from traffic cameras, or video data taken from smart stores or smart offices. This data can be quite large and may have specific processing workflows, but may be used to build and horizontally scale ML models that may be distributed and run at Edge to be able to process image or video data streams in real-time. In other cases, the Edge may utilize compute resources provided by the Unified Data Platform. A collaborative approach whereby both the Edge and the Unified Data Platform work together enables the rapid building and deployment of ML models at Scale.

Unstructured data that is at Motion includes such things as telemetry data coming from sensors that are fitted on vehicles, mechanical parts or devices that report on their usage. It may also include activity such as in-store foot traffic, web events by users such as clicks on ads, webpages, or demos. Such data when available is typically shallow because it is sparse, but also is able to provide a lot of insights about what latent behavioural patterns may exist to be able to peek into the future such as a potential upcoming churn of a user against a competitor product or a service. Also, it is shallow because it is sparse, and also specialized processing workflows to convert such data into features that can be utilized for building ML models is required.

In both cases above, the Unified Data Platform needs to provide more than how to ingest, cleanse your data and build a data pipeline, but also how to provision the infrastructure to run such specialized tasks at scale.

8. Case Studies and Industry Examples

Successful Implementations of Unified Data Platforms Various organizations are successfully leveraging unified data platforms to address their data needs, allowing them to operate with enhanced efficiency and productivity [1]. This book section describes some of these organizations that have implemented the various phases of a unified data platform and discusses some of the approaches they have taken. Learning lessons from these industry leaders can help others design their data systems with the advanced capabilities possible today to drive their digital business transformation. These examples should not be viewed as the only solution. The creativity of technologists can produce a wide variety of solutions to analogous problems, as evidenced by the diversity of hyperscale companies around the world.

8.1. Successful Implementations of Unified Data Platforms

Unified Data Platforms (UDP) are an attractive solution to the problem of vertical integration of enterprise data architectures. Business systems and applications are often born disconnected by business users operating in silos, resulting in domains of functional expertise that themselves comprise isolated, monolithic application and database models that are managed not only separately from one another, but also by separate domain owners, resulting in “siloed data”. Multi-domain integration of enterprise data becomes complex, brittle, and difficult to maintain, especially when the underlying systems supporting the application models in the different data silos change and evolve over time. Organizations are therefore always looking for approaches, toolsets, and more fractal architectures that offer a solution to achieve the enterprise-wide multi-domain integrated state. UDP is one such state.

Many organizations are moving toward the UDP model as the predominant architectural evolution of their enterprise data footprint. The UDP abstraction supports organizations that are focused on building new critical applications utilizing emerging or growing technology footprints such as cloud native development, real-time streaming, or machine learning, but the UDP is equally relevant to lift-and-shift integration of legacy application and data models,

supporting the migration and evolution of data and analytics strategy in mission critical programs.

Organizations that focus on critical application investment decision-making, empowered investment governance, and the implementation of best practice program management strategy and execution patterns for risk management and control, specification and design, plan and build, run and test, and adoption and data-bias training are able to exploit the capabilities and dynamics of the UDP model for successful business outcomes.

8.2. Lessons Learned by Industry Leaders

Lessons Learned from Industry Leaders Data infrastructure design changes constantly, with new, smart approaches that can increase business impact. Several leaders in the public cloud space are investing heavily in expanding their data infrastructure. While there are many possible companies and sectors to highlight, this chapter focuses on organizations embedded in the data arena that demand novel design choices to maximize data business value. The private-sector examples here still exhibit elements that are relevant for government enterprises. Many of the data business challenges are very similar, and, likewise, the tech choices are often shared across sectors. Doing more with data is generally a goal of any organization. Looking at a set of company portfolios that permeate enterprise space walks as thought leaders can provide inspiration for forward-looking organizations.

Over the years we have learned lessons based on these experiences, the products we built, the customers' businesses and their data needs. As the innovators in the unified data platform space, we hope to share some of these lessons to help guide others who have the same vision and goals we had but may have gotten stuck in the crowded, fragmented, data pipeline, data lake, or data warehouse product markets. Our conclusion, echoed by our customers, is that there is a substantial difference between point solutions for data ingestion pipelines, data lakes, and data warehouses, and their integration into unified data platform solutions; meaning one product for Cloud Data Infrastructure, Data Engineering, and Data Analytics, all built on the same Cloud Data Platform software foundation. While any of the point solutions can be done quite well, it is difficult for organizations to do the integrations themselves; and even more problematic to have to switch tools later to scale for growing business data demands. By providing full end-to-end capabilities needed for Cloud Data Infrastructure, Data Engineering, and Data Analytics, data teams use the same solution for all aspects of their business;

from data preparation and ETL, data storage and management, to business user self-service analytics dashboards. This speeds and simplifies development while eliminating internal customer provider silos.

9. Best Practices for Implementation

This chapter provides you with a set of best practices for successful implementation of UDPs based on the expertise of our team and key learnings from the field.

9.1. Strategic Planning and Roadmapping

Enterprises have many competing priorities and have been investing in many areas of Digital Transformation, which can lead to a lack of focus. Having an informed and strategic roadmap helps enterprise use its resources optimally. Senior leadership should assess the current systems infrastructure, silos, and product acquisition initiatives and help articulate a strategic vision along with a roadmap that identifies key phases within the roadmap. This prioritized series of phases should focus on initial disruption blockers or low hanging fruits that can show progress followed by areas that can benefit enterprise users. Inversion of data use and delivery and prioritizing UDB end-user consumer engagement and experience are fundamental to roadmap design.

Strategic planning and road mapping activities are critical in successful implementation of unified data platforms. There are three aspects of roadmapping that need to be considered: Strategic alignment, planning and prioritization, Advanced use-case roadmaps.

Although there are compelling arguments for an enterprise-wide approach to unified data platforms from a data and technology perspective, the business case needs to carefully consider the cost-value balance from both an operational and capital perspective. There may also be an evolution or maturation involved in the use of unified data platforms where the business value on data products grows slowly and requires investment over time in a set of prioritized use cases. In many organizations today, the highest priority use case is often for real-time situational awareness – where data products are used for helping inform and advise operational decisions.

Use-case roadmaps that go into advanced territory would include data products that provide sophisticated decision influence or decision automation capabilities such as those found in the application of advanced analytics, machine learning,

or AI automation techniques. With the embedding of sophisticated decision influence or terminal value, the significance to the overall enterprise could also be high. Such activities could also imply a longer investment duration with longer time to operationalize. In this case, it is useful to develop proposed timelines for data readiness, capability building, productization, and airport maturity and readiness.

The sophistication of roadmaps can vary. Some roadmaps just show the commercialization timeframes while others include discrete and elaborate data productization and data use analysis timelines, operational capability building timelines, associated investment, as well as projected results with business case justification.

9.2. Stakeholder Engagement and Collaboration

Developments along the roadmap will touch many groups across IT and line of business units. Facilitate collaboration across units to ensure that multiple use cases are planned within a release. Design easy and relevant onboarding for multiple people, and focus on user experience and adoption strategy, as collaboration will be critical for success of your UDB initiative. Define a set of core UDB services together and begin their adoption across broader initiatives like advance analytics, MLOps, operational workloads, or BI reporting. Just-in-time training for line of business unit staff should happen before a use case implementation starts, and workshops for support staff in line of business units should run parallel to workload implementations. Data discovery and democratization is fundamental design and operational characteristics of UDBs. Organizations should design easy data access through familiarity around BI tools wherever possible.

Success in the data domain requires organisations to go beyond the technical implementation of a Unified Data Platform. It should include an understanding of local data needs, resource and skill development, management of the new operational model, and active promotion of the platform's usage. As such, the need to engage and collaborate with a diverse group of stakeholders is vital to ensuring long-term success. Stakeholder engagement is an important aspect of realising the strategic vision of unified data platforms. Their success depends largely on identification of common processes and data requirements across the organisation, and agreement on a common way to meet those requirements. Often, there is no single authoritative data source that will satisfy the needs of each functional area and department, and routinely common data is often housed both centrally and within the individual departments. Through engagement and collaboration, directed by a centre of excellence, organisations can tap into the

benefits of predictive and prescriptive analytics rather than just hindsight, or what has occurred in the past and when. Dismantling silos through a centre of excellence and collaboration with key business stakeholders is critical in developing a unified data strategy. Stakeholders understand the business problems being faced and can help determine how information can improve decision making and how to create data hygiene procedures, such as audit, access, and training on usage to build trust. Key stakeholder involvement is usually created through discussions, workshops, or collaborative sharing sessions with the information units to generate input and feedback for specific scenarios. Creating a stakeholder engagement strategy provides a clear mandate for the centre of excellence during development of the platform, responsible for ensuring input into the solution's design, by means of workshops with the visualisation and modelling teams.

10. Conclusion

Unified Data Platforms are a new category of data platforms that allow organizations to analyze all data types from an easily accessible location in the same way, enabling significant operational efficiency. Although the need for Unified Data Platforms has been known for a long time, we are only now seeing the market evolving for various reasons. First, possibilities offered by Cloud-based architectures and parallel compute frameworks such as distributed Massively Parallel Processing systems and vector-based pre-trained Neural Network architectures are enabling features. Second, external forces such as regulatory standards requiring organizations in many industries to engage with unstructured data types has provided support for this category of platforms.

In conclusion, the category of Unified Data Platforms are only in their infancy but can only grow from here. Data limitations brought about by traditional vendor monopolies are being rapidly lifted as organizations begin to move unstructured data into the mainstream of operational business use cases. The benefits of such a transformation are substantial business ROI in terms of operational efficiency and improved decision-making. As organizations face increasing pressure to derive value from every single data source, the capability of Unified Data Platforms to expedite and simplify the development of AI solutions to problems spanning varied data types will offer tremendous appeal. The inherent nature of Unified Data Platforms to simplify the integration and deployment of AI solutions so critical to addressing use cases prevalent in organizations across every industry may be one of the most attractive features of the platforms. The

ability to do all of this while including easy-to-use enterprise data management processes and streamlined developer use patterns is what will clearly set apart Unified Data Platforms. The demand for such a Unified Data Platform that caters to the varying use cases and complexities across all aspects of AI development in your organization is urgent and could not be timelier.

References:

[1] S. P. Panda, Artificial Intelligence Across Borders: Transforming Industries Through Intelligent Innovation. Deep Science Publishing, 2025. doi: 10.70593/978-93-49910-25-6.