Open Lakes, Not Walled Gardens

Unlocking Data for the Age of AI

Raghu Ramakrishnan
Technical Fellow, CTO for Data

Josh Caplan
Head of Product for OneLake

The Vision of Open Lakes for Analytics

Enterprise data estates contain many data sources for a variety of reasons, including differences in usage (e.g., operational databases store data in row-oriented pages to support efficient transactional in-place updates, while analytic systems use columnar table representations), differences in organizational ownership and location, differences in vendors and software, and sometimes just legacy reasons (e.g., acquisitions). This proliferation of data sources has long been the case, and the emergence of public clouds and a rich array of database offerings has further solidified this state of affairs. Analytics, however, requires a unified view of the entire data estate, if we are to draw true insights from data.

Historically, this conundrum has led to many proposals: departmental “data marts”, enterprise warehouses, big data “lakehouses”, and variations such as data domains and meshes. These proposals differ in some respects, including the span of data to aggregate and how to organize data ownership and governance, but without exception, they share a common theme: analytics requires us to create and maintain a current and aggregated view of all relevant data.

We are entering the golden age of analytics with the maturation of big data tools, elastic cloud compute and storage, and especially, the exciting advances in machine learning and generative AI models. Enterprises see the opportunity to view their core assets and processes through the lens of their data, to derive insights, to discover and track trends, to build and deploy machine learned models for predictive analysis, and increasingly, to harness the exciting potential of generative AI models. To fully realize these benefits, we must simplify the foundational task of running these analytic tools on all data, from operational and non-operational sources, structured and unstructured, in the sprawling modern data estate.

The first step is to ensure that there is a single data format that all engines understand and can run on directly. We believe that this format must be open, to avoid locking customers into any one vendor’s toolchain. With the emergence of open Parquet-based table formats such as Delta-Parquet, Iceberg and Hudi, we have viable options, though none of these has emerged as a dominant format yet. Fortunately, all of them store data in Parquet files, and differ mainly in how they represent table metadata, and it seems possible to interoperate using metadata-based translation approaches, as is proposed in the XTable (1) open-source project.

The second step is to enable customers to bring all enterprise data together. To this end, we believe that customers must have full control over their data and metadata, including how to aggregate it from diverse sources and what engines to use for different types of analytics. Historically, customers have had to create and manage data pipelines to incrementally copy and maintain tables from diverse sources in their analytic "lake". Increasingly, hyper-scalar service providers and pipelining services are making this step easier and fully managed. Furthermore, if source and target tables are both in the open lake format, we now have a powerful new option—virtualize the remote data for analytic access by defining a read-only reference to it from the lake, rather than physically copying it into the lake.

These two steps, realizable with existing open technologies and standards, provide a good starting point for open lakes, allowing customers to easily create a single aggregated view of all their data, across all sources in their data estate, and bringing to bear any analytic engines that understand these open table formats (from any vendors). Realizing these steps will require vendors to support open Parquet-based table formats and references to their data from external lakes.

This is by no means all that is desirable. Databases traditionally support multi-table transactions and table schema alterations, and databases built on updatable Parquet table formats constantly do physical reorganizations. Standardizing protocols to insulate queries that access data via references to tables from such ongoing table updates would be desirable, e.g., extending concurrency control concepts such as Snapshot Isolation appropriately. As another example, databases also implement access control, storing granted access rights in their operational catalogs and enforcing these when a user seeks to query or update the data. Today, there are no widely accepted standards for representing access control policies. The common pattern used in aggregating data is for a user who has access to the source data to set up a copy (or define a reference) in the target environment (the analytic lake, in the context of this discussion) and to define anew the desired access controls in the new environment. It is desirable in some scenarios to define some access controls that must "travel with the data", e.g., if data is sensitive and estate-wide policies are in force for sensitive data, we must ensure that these restrictions are reflected when data is copied or referenced and enforced appropriately from all access points.

These are areas for further work, and we hope to see this addressed in future open-source projects.

In the rest of this paper, we describe our implementation of the open lake vision, including data storage, analytics, sharing and governance, in Microsoft Fabric and OneLake, together with Microsoft Purview for data estate wide governance.

Microsoft OneLake in Fabric: Design Principles

We have designed Fabric to make it easy to aggregate the entire data estate in OneLake and to use a rich array of engines in Fabric. In addition, by adhering to open standards we fully support customers in bringing non-Fabric tools to bear on data in OneLake. Openness means external data can easily be brought into OneLake and analyzed with Fabric tools, and equally, that OneLake is not a walled garden—customers have choice in what analytic engines to use, including other services, such as Azure Databricks.

Enterprise data must always be governed and secure, and any approach to aggregating data must address these core issues. With built-in Fabric data organization, governance and security features integrated with data estate governance powered by Microsoft Purview, we aim to support flexible and comprehensive data management that encompasses access control, data federation and sharing with business partners, life-cycle management, and compliance support for Fabric in the context of the entire data estate. Taken together, Fabric and OneLake represent a new generation of converged lakehouses and warehouses that extend current systems in ease of use, broad range of fully composable tools, and comprehensive governance capabilities.

In this white paper, we present the design principles guiding OneLake and Microsoft Fabric, and then outline the product features that realize the vision of a governed, secure, and unified lake of all data, as illustrated below:

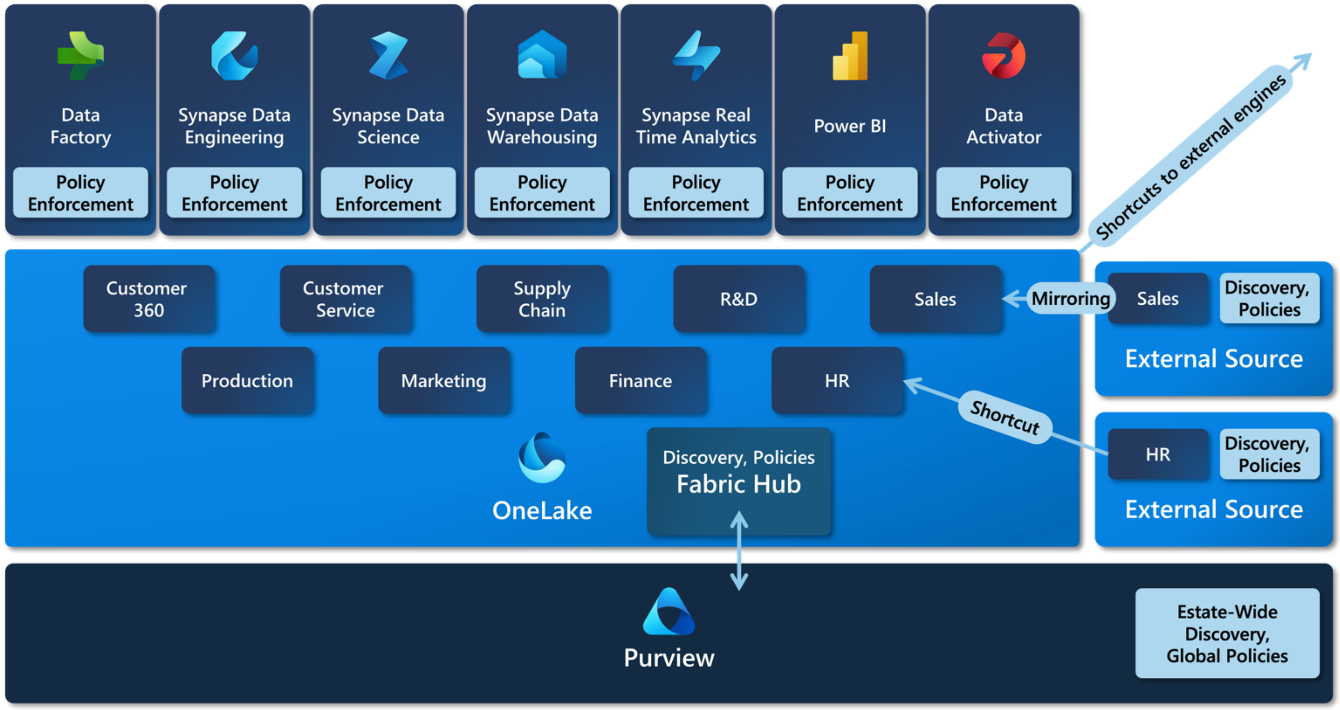

Figure 1: The OneLake Vision: A single data lake for the entire organization, secure and governed

Microsoft Fabric brings together a powerful suite of analytic engines, all of which store data in a single format to enable seamless interoperability. All data is available in OneLake, the unified data lake in Fabric. While OneLake is similar in many ways to the lakehouse in (2), we emphasize composability of engines and a global data estate perspective. OneLake is an open lake of all enterprise data, more precisely a unified view of all enterprise data (including sources external to Fabric). Collaboration, sharing, and global governance across the entirety of the data estate, including but not limited to OneLake are fundamental considerations.

Our goal is to democratize data analytics by making it as simple as possible to carry out the full range of analytic tasks with a single sign-on in a secure, governed environment. To accomplish this, we address several distinct challenges, guided by the following principles:

Analytic tools must compose easily, without the need to copy or load data tool-by-tool

The good news is that there are excellent tools for virtually every phase of analytics, from data ingestion to integration and cleaning, data science, interactive queries and serving, business reports and visualization, machine learning, and to acting on the insights gained thus. Typically, however, these tools are silos with their own data and metadata representations, requiring data to be imported afresh into each tool and making it tedious to carry out multi-step analyses going back and forth across tools. In Fabric, we have rearchitected all structured-data engines to work on the same underlying data representation in OneLake, as we discuss shortly, and have integrated these engines to compose seamlessly. Once data is in OneLake, there is never a need to copy or move data into any Fabric engines.

SaaS: One-click to data

Thanks to cloud computing, we have come a long way from the old days of sanctifying the machine room; ordering, receiving and installing hardware; then installing software, all prior to writing a lick of relevant business logic. Still, we commonly begin by installing and managing software in the cloud (Infrastructure-as-a-Service, IaaS), or at best provisioning resources to run our software (Platform-as-a-Service, PaaS). To truly democratize analytics, we need to make analytic tools such as business intelligence, Spark, and SQL as easy to use as online Office tools such as Excel—just select the data in the tool you want to apply, and let the system handle resource provisioning, billing, etc., in a secure and governed environment. Generative AI is making it increasingly possible for domain analysts who are not experts in SQL, Python, and other data arcana to carry out sophisticated analysis, without the help of data professionals. The least we can do as systems and service architects is to get the plumbing out of the way and allow users to get directly in touch with their data with their analytic tool of choice.

No silos: The entire data estate must be accessible easily in OneLake

As we noted earlier, enterprise data estates contain many data sources for a variety of reasons. Creating and maintaining a current view of the entire data estate is a tedious and error-prone task that adds significantly to the total cost of analytics.

Through features such as shortcuts and data mirroring, which we describe shortly, we have made this much easier. All Fabric engines write tabular data into OneLake in the same open format, and data in external sources can be virtualized easily via shortcuts or incrementally mirrored in OneLake, in the standard open format, with no user managed ETL pipelines.

We observe that silos are not only a challenge arising because of data sources across the estate that are external to the analytic lake—many organizations struggle with analytic silos arising from data ownership and organizational silos; shortcuts eliminate such silos as well, with no data copies, as we discuss later.

Security and Governance

It should be clear that we see data in OneLake including data that is mastered elsewhere in the customer's data estate, where it is undoubtedly secured using features of the local data store (e.g., Grant/Revoke statements for row and column-level security in external SQL engines). Thus, the access control capabilities of OneLake's security framework should be capable of enforcing the controls in the source system when that data is accessed through OneLake, either via a shortcut or mirroring.

Further, as data estates continue to grow ever more distributed in nature and regulatory and compliance requirements continue to increase, we believe it is important to have a global view of the entire data estate, and to be able to govern it holistically. For this reason, Fabric is being integrated with Microsoft Purview, to give data administrators both a global catalog and the ability to set global data policies such as mandatory access control rules and data lifecycle policies. The integration with Microsoft Purview already enables collection of Fabric audit activities and labeling sensitive data with Information Protection sensitivity labels, which are widely used to label sensitive data in Office. Enforcement of Purview global information protection and data loss prevention policies are on the roadmap. Our vision is to extend governance capabilities that are already widely used to govern unstructured data in Microsoft Office to the entire data estate, including structured databases.

Open Ecosystem

We must not add proprietary barriers—proprietary formats, proprietary protocols, proprietary tools—to data in OneLake. Customers should use Fabric engines because they are easy to use and deliver great price-performance in a secure and governed environment, not because the data is hard to get at with alternative products. We must make it easy for other ISVs to integrate their services and engines to run over data in OneLake and integrate with the rest of Fabric.

This commitment to an open ecosystem is in line with recent regulatory trends, e.g., new EUDB laws that speak to the portability of customers' data across vendors and cloud providers.

Microsoft OneLake in Fabric

We now take a closer look at how the above principles are realized in Microsoft Fabric.

OneLake aims to be an open, unified data lake that brings together the entire data estate of an organization for analytics, across all data sources, users, applications, regions, and clouds. To help organizations realize this vision, we simplify the task of making all enterprise data securely available for analytics, while following governance best practices (e.g., data mesh) and respecting constraints (e.g., organizational ownership, location, workload SLOs) that require independently managed data sources external to the analytics lake.

We do this by:

1. Making it easy to reference or mirror data in external sources—across all the organization's regions, clouds, users, applications—through shortcuts and data mirroring. Shortcuts bring together open lakes with no data copying1 and mirroring allows us to handle proprietary and non-columnar sources by incrementally maintaining a near real-time copy in OneLake.
2. Enforcing security in OneLake at the folder level and within each Fabric engine. This includes support for access control and network isolation.
3. Complementing OneLake's security and governance with enterprise-wide global policies through Microsoft Purview integration.

In this section, we focus on how to create a comprehensive view of enterprise data in OneLake for analytics, and in the next, we discuss security and governance. While we aim to further simplify and extend our support of enterprise data security and governance, a core principle we already ensure is that data is always secured when it is accessed, whether data is physically in OneLake or virtualized through a shortcut.

[1] With the exception of remote sources that are cached for performance.

Microsoft Fabric: Comprehensive Analytics with a Single Sign-On

Microsoft Fabric makes it possible to bring all data, enterprise-wide, together in a secure way, and to use a powerful suite of analytic engines to gain insights, collaborate, and act upon the insights:

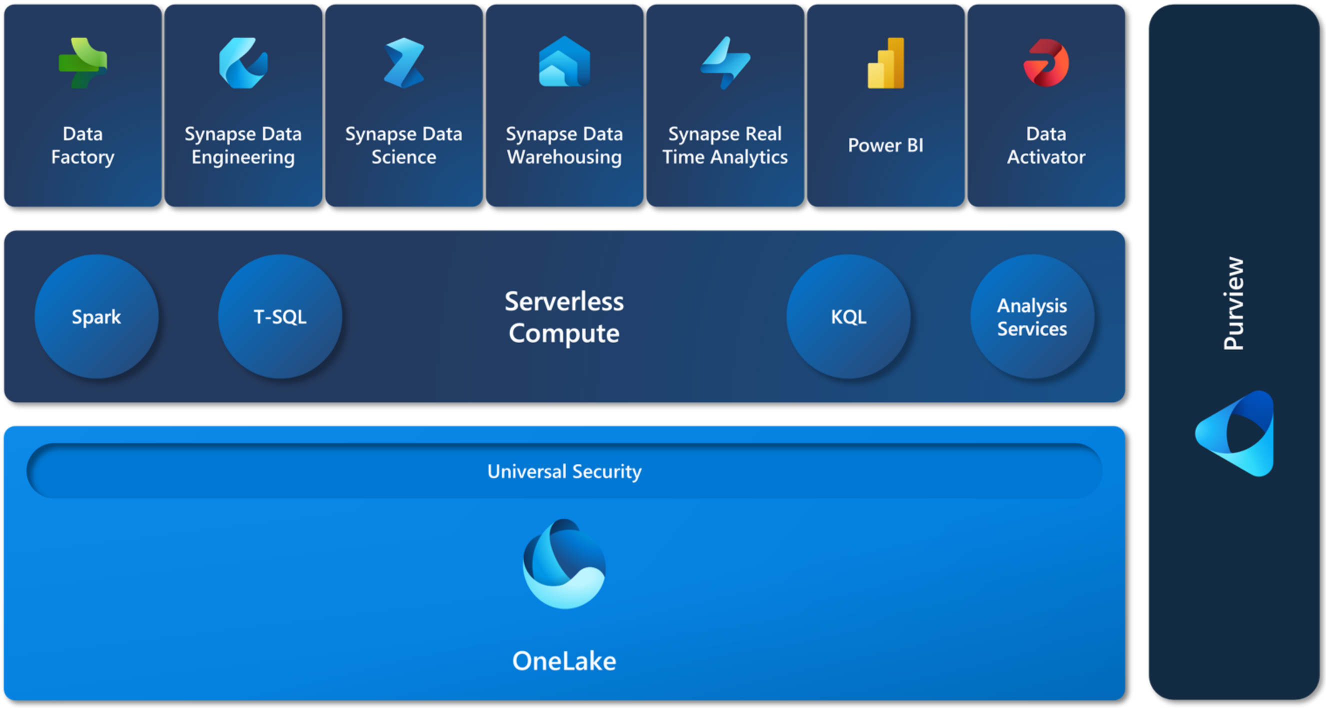

Figure 2: Microsoft Fabric: Complete data platform with tools for every data professional

We strongly believe that file and table formats should not lock customers into different vendors' products, and to this end, we have embraced open formats and interoperability. Fabric has standardized on an open Parquet-based (3) data format to store tables in all its engines. This format is currently Delta Lake (4). We are actively working with the Apache open-source community on an interoperability project called XTable (1) (5) to enable support for other Parquet-based open table formats including Iceberg (6) and Hudi (7).

- We have re-engineered Microsoft engines including Power BI and SQL to operate directly on the open table format, and facilitated mirroring into this format for external sources that use proprietary formats.
- OneLake is built on ADLS Gen 2 and supports existing ADLS tools such as Storage Explorer, AZ Copy, and ADLS Gen 2 APIs (which conform to HDFS open standards (8)).
- OneLake features a Data Hub, making it easy to discover, access, manage and reuse all OneLake items.
- Our governance framework and catalog are based on open Apache Atlas APIs (9).

To use any engine in Fabric, a customer simply needs to make the data available in OneLake, the unified data lake in Fabric. Tables in OneLake are in the open Parquet-based format; files can be in any of the popular formats. Data can be physically copied to Fabric or made accessible virtually through a reference or "shortcut".

The engines available include Data Factory for building data pipelines and transformations, Spark for data engineering and data science, SQL for data warehousing, tools for streaming data ingestion, Azure Data Explorer for real-time analytics and forensic analysis, Data Activator, a new tool, for monitoring data changes and acting on them, and of course, industry leading Power BI for reporting. The list of engines is constantly growing, and we are developing an SDK for ISVs to add their own engines to Fabric.

OneLake: All Your Data in One Place for Analytics

We make it easy to provide shortcuts to data created by other engines in the open formats recognized by Fabric. Shortcuts are pointers to files, folders and tables that are very easy to create. With shortcuts, all Fabric engines can run on the data without moving it or copying it, and with no disruption to existing usage by the host engines. For example, creating a shortcut to Delta-Lake tables created by Databricks allows customers to run Fabric engines while continuing to run Databricks Spark, all without data copying or movement. In the spirit of openness, customers who wish to run Databricks Spark on data created in OneLake by Fabric engines can easily do so as well. Ultimately, our aim is to give customers choice over which engines they use for different tasks; data representation should be a means to this end, not an impediment.

Shortcuts make it easy to run Fabric engines on data in the OneLake open table format, but they are not a panacea for access to all external data sources. First, operations on large remote datasets can be slower when accessed through a shortcut. To address this, we can optionally cache remote shortcut data in OneLake with the customer's permission, thereby also minimizing any associated egress costs from the remote source. (Note that shortcuts can be applied to any data, not just tabular data, and shortcuts to images, videos and small files work well even without caching.) Second, the data source may use a non-columnar or proprietary format.[2] For example, while Databricks uses Delta Lake format, Snowflake uses a proprietary column format (but is extending support to Iceberg). Operational databases, including SQL Server, Azure SQL DB, Cosmos DB, Postgres, MySQL, Oracle, Teradata, AWS Aurora, Google Spanner, etc., all use non-columnar table formats.

We now support shortcuts to popular open protocols such as S3-compatible sources and we plan to add custom shortcuts, which will enable creation of shortcuts from any system into OneLake or from OneLake to other systems, to further enhance openness of OneLake.

For tabular data created by other engines in proprietary or non-columnar formats, Fabric makes it easy to maintain a near-real-time copy with zero user-managed ETL in the open table format of OneLake with mirroring (10), which significantly extends the older Synapse Link feature, using change data capture on the source for incremental maintenance of a copy in the OneLake open table format. We aim to support mirroring of all Microsoft data sources—including SQL Server versions 2017 and up (on-prem and in VMs), Azure SQL DB, Cosmos DB, MySQL and PostgresSQL—to OneLake, as well as mirroring of a growing number of external sources, including MongoDB, Oracle, Teradata, BigQuery, RedShift and Snowflake. We are also working on an extensible approach for any DW / DB vendor to add their data warehouse / database as a mirrored source to Fabric.

We also aim to support mirroring of OneLake data for customers who wish to run other engines over this data—our core principle is that customers should be able to choose their analytic engines with minimum friction, and all data sources should support this objective.

[2] Recall that we aim to support all major open table formats in OneLake through interop mechanisms such as XTable.

Security and Governance in OneLake and Microsoft Fabric

Data is valuable, and all enterprises take great pains to secure and govern their data. Thus, when considering how to simplify access to data from diverse sources through mechanisms such as shortcuts and mirroring, we must take into account how to also address the corresponding security and governance measures.

In this section, we look at how to secure and govern data in OneLake using built-in capabilities in Microsoft Fabric, complemented by global data estate governance in Microsoft Purview, which is already on and integrated into Fabric.

Security begins with Authentication

Microsoft Fabric, like Power BI, is a SaaS service built on Azure and is a highly integrated, end-to-end, and easy-to-use product that's designed to simplify analytics and to protect sensitive assets. This starts with authentication; every interaction with Fabric (from logging in, using the Power BI mobile app, running SQL queries through SSMS, etc.) is authenticated with Entra ID (formerly Azure Active Directory).

Entra ID allows you to set up Zero-Trust based security with Microsoft Fabric. SaaS cloud applications and a mobile workforce have redefined the security perimeter. Employees are bringing their own devices and working remotely. Data is being accessed outside the corporate network and shared with external collaborators such as partners and vendors. Corporate applications and data are moving from on-premises to hybrid and cloud environments and controls need to move to where the data is.

Some components of Entra ID that are also available for Microsoft Fabric to enhance security are for example: Conditional Access. This can be used to protect all the data and workloads in Fabric – gives the ability to define/restrict a list of IP ranges for inbound connectivity to Fabric, provides ability to mandate MFA, provide ability to restrict traffic based on country of origin, devices, etc.

Access Control in Fabric

Fabric, OneLake and Purview provide security and governance at different layers to ensure that data is secured all the way from the lake to the business user. We want to ensure that data can be secured at the lake without needing to copy it out to another engine to secure it. All data in OneLake maps to a Fabric data item and workspace. Access to the raw data in OneLake is controlled automatically by the access granted to the workspace or data item and can be further controlled through fine grained data access roles within the item. Security defined in OneLake will travel with the data wherever it is used and across shortcuts.

While OneLake stores the data, it can be served through multiple different analytical compute engines inside and outside of Fabric. Some of these engines provide their own fine-grained security features including row/column-level security and dynamic data masking. The different layers interact with each other through connections that either flow the calling identity through (single sign-on, SSO) or connect as a delegated identity. To further restrict security at the engine layer, an engine can access data in OneLake under a delegated identity and add their unique security features to further restrict a user's access to the data without copying it.

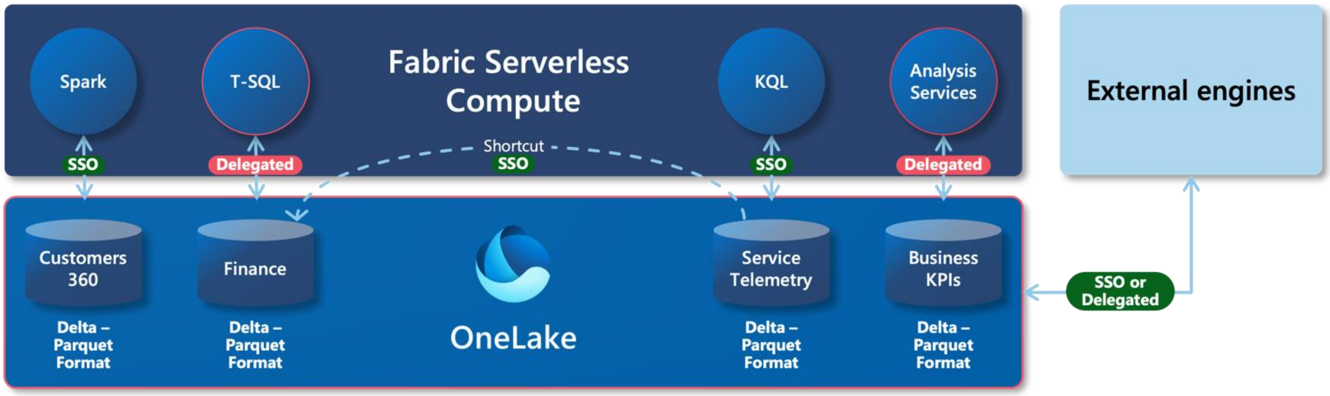

Figure 3: OneLake security definitions enforced across Fabric and external engines

While all data can be directly secured in OneLake, we recognize that it is common for data to be secured elsewhere, i.e., in external data catalogs and database systems. It is not uncommon for an organization to use multiple methods of securing and governing data including many custom solutions built within an organization. This is particularly true for data that lives outside of OneLake and is brought in through shortcuts or mirroring. OneLake makes it possible to build one virtual lake which includes security definitions no matter where they are mastered.

Let's look at an example where data lives in an external store where row-level security is also defined. To virtualize this data into OneLake and to ensure the same granular security, the following would happen:

1. A user who has appropriate access to the original data source would act as a delegator. He or she would either create a shortcut or mirror the data into OneLake.
   - If they create a shortcut, that shortcut would run under a delegated identity.
2. Once the data is available in OneLake, further access control happens within OneLake and the engines that access it. The delegator will define row-level security within one or more of the engines to match what is available in the external store.
3. The delegator would limit direct OneLake file access to only users or identities that are allowed to access the full set of data.

In future, as features like row/column-level security, dynamic data masking and more become available in OneLake, specifying these restrictions will become simpler. Rather than doing it per engine (which is already supported) all security definitions can live in the lake alongside the data and travel together to any compute engine that tries to access it. With SSO connections between all the layers, security can be defined once in OneLake and enforced on access by any engine.

Azure Data Lake Store and OneLake: Better Together

It is worth stepping back and considering how ADLS and OneLake compose. First, all data in OneLake is stored in ADLS; every OneLake file is an ADLS file. Thus, the cost and performance of storing data in OneLake closely tracks that of storing data in ADLS.

The differences are two-fold. First, OneLake adds a layer of abstractions to better support the SaaS model of Fabric. Users do not have to create ADLS accounts, storage is SaaS'ified via OneLake, which also supports abstractions such as shortcuts.

Second, in addition to file and folder ACLs similar to ADLS scoped by Fabric workspaces (key to support for data domains), OneLake supports RLS/CLS over tables. That said, we fully recognize that in many scenarios, customers might already have data in ADLS or use other engines that store data in ADLS. Shortcuts provide a way to bring such data into OneLake without data copying, and in a way that allows customers to build on the Posix-compliant file and folder level ACLs in ADLS. We recognize that access to an ADLS folder implies access to the data therein, and that many customers rely upon ADLS access control to manage security for their ADLS data. While bringing that data to OneLake via a shortcut provides the ability to further restrict by Fabric workspace and table RLS/CLS (and Purview global policies at the file and table-column levels), access via the user-managed ADLS account continues to be a valid and complementary way to manage access. The best combination of access control mechanisms is for users to determine based on their scenarios.

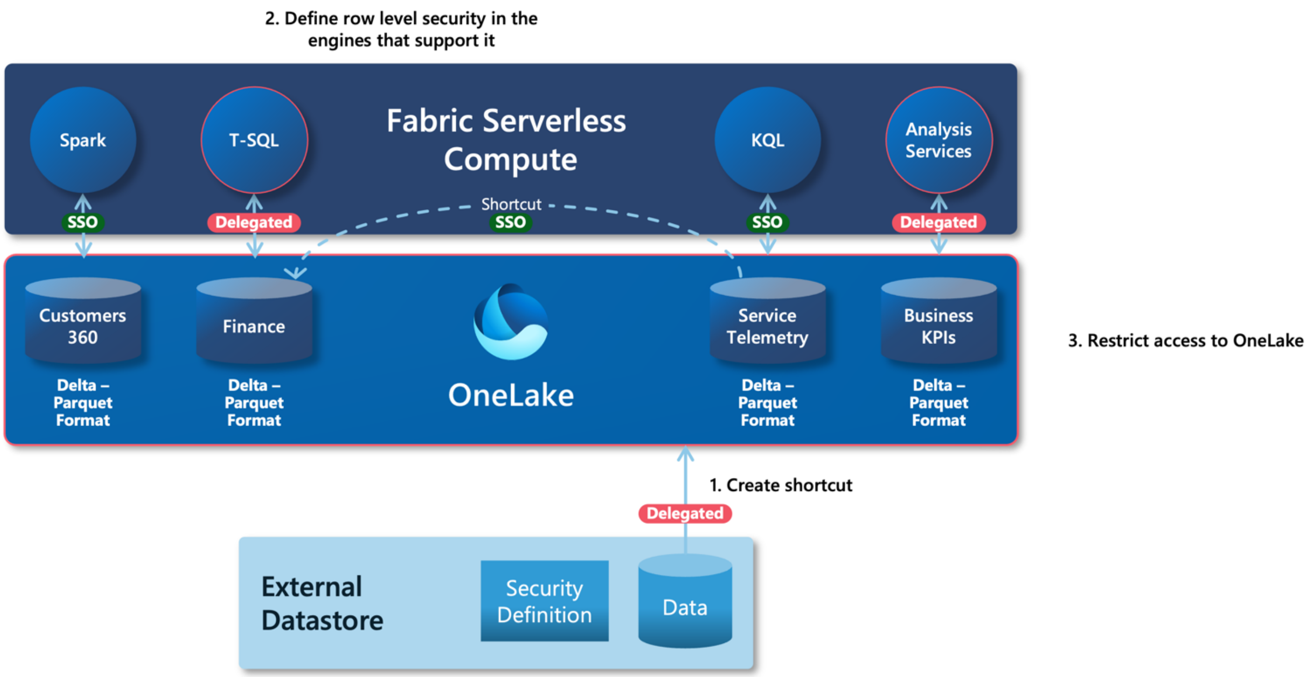

Figure 4: Connecting ADLS data to OneLake with built-in security layers

Collaborative Organizational Governance

The vision of a single unified data lake is a goal for many organizations, but people and process challenges must be addressed in addition to technical challenges. Coordination through a central team creates overhead and federated approaches such as data mesh (11), which aim to solve these challenges by enabling different parts of an enterprise to manage their own data lakes, have gained traction.

OneLake supports the creation of a single logical lake organized into independently managed domains for efficient collaboration, as an instantiation of the data mesh pattern. In Microsoft Office, different teams can have their own Teams channels or SharePoint sites; similarly, workspaces in OneLake allow different teams to work independently while still contributing to the same data lake. Each workspace can have its own administrators, access control, data residency and capacity for billing. All data stored in a workspace is owned and secured by the owners of that workspace.

A workspace typically aligns with a single team or project. A typical business domain will have multiple teams and projects. With OneLake domains, workspaces can be grouped into a business domain which provides an additional management and governance boundary between the workspace and the tenant, thus allowing organizations to optimize to business needs, while allowing granular and effective control. Shortcuts enable sharing of data without data duplication and without changing the original ownership of the data.

We aim to make OneLake the easiest place for anyone in an organization to independently land their data and enable controlled sharing, just like OneDrive and SharePoint are the easiest place to land documents for sharing. The people landing the data in OneLake could be data engineers in IT or business users working on their own projects. Data certifications in OneLake can be used to distinguish between official domain certified or recommended sources of data and other unofficial sources. This way, all data can coexist in the same data lake. When data lands in OneLake, it is governed by default and under the purview of a domain or tenant admin.

Microsoft Purview: Estate-Level Governance

We begin with a brief overview of Microsoft Purview for context. Microsoft Purview (12) recognizes that the data estate for an enterprise has many independently managed data sources, and there is a need for a global catalog of all assets across all sources, global policies to secure sensitive data, and support for managing critical data risks and regulatory compliance. It brings together a global view of both structured (e.g., tables in Azure SQL DB, Microsoft Fabric, Oracle, AWS RDS) and unstructured (e.g., Microsoft Office, cloud file stores such as ADLS) data in a central catalog maintained incrementally in Azure for data in a variety of cloud and on-prem locations. It extends Microsoft Office support for Information Protection and Data Lifecycle Protection through labels to columns of tables, in addition to traditional Office sources such as email in Outlook and SharePoint repositories. For example, a CDO can define a global policy prohibiting vendors from accessing PII data. Using automatic and curated classification to identify instances of PII data in any data source within the enterprise, the policy can be enforced across all sources, including Microsoft Fabric, that support Purview labels, thus complementing any local access control defined in each of these sources.

Microsoft Purview Global Policies in Fabric

Security defined at the OneLake and engine levels is specific to the data it is securing. For example, column-level security restricts a specific column, in a specific table, in a specific data item. Global information protection policies powered by Purview support mandatory access control, adding another layer of protection against over-sharing of data. Policies are not specific to any data item; rather they restrict access to specific types of data. Like files in Office, data can be labeled and classified as different sensitive types. For example, if you want to ensure that PII data doesn't end up in the wrong hands, a policy could restrict access to sensitive PII data to only users who meet certain qualifications. The policy would take effect and restrict all columns in OneLake that are labeled or detected as PII, overriding any access that was granted in OneLake or the engines.

As a first step, Fabric has already integrated support for Purview Information Protection sensitivity labels to classify sensitive Fabric data — a familiar concept to Office 365 users who employ these labels every day on their files and emails. When a data owner in Fabric assigns a sensitivity label to a lakehouse or any other Fabric item, the label will remain with the data through all operations creating derived items in Fabric, including Power BI reports. In addition, when exporting data from Fabric to Office files, the label and Office file protection will automatically be added to the Office files. Security admins can also set global policies to require data owners to use sensitivity labels on new data items created in Fabric.

Fabric gives its admins and data owners useful information about Fabric's data assets with Fabric's Microsoft Purview Hub. Microsoft Purview Hub shows insights about sensitive data, certified and promoted data, and acts as a bridge to advanced features in Microsoft Purview portals. Fabric also works with Microsoft Purview audit, which gives Fabric and compliance admins a complete view of Fabric activities for auditing purposes. All user and system actions are recorded in the audit logs and accessible in Microsoft Purview compliance portal.

Looking Ahead

As we've seen, to unlock data for a data-driven future in which analytics and AI hold transformational potential, we must address many challenges. Like lakehouses, we must support the full spectrum of data, from documents to telemetry to multi-media, in addition to structured table. We must support a wide array of engines, from ingestion and real-time transformations to monitoring, data exploration, data science, training and deployment of ML models, all the way to data serving and reporting. And like traditional warehouses, we require security, governance, and predictable performance at scale.

Our vision for Microsoft OneLake (13) and Microsoft Fabric (14) is to deliver this convergence of lakehouses and warehouses, as shown in Figure 1. In previous sections, we focused on features that are already supported in Microsoft Fabric. In this section, we look ahead to upcoming features that bring us closer to a full realization of the vision, addressing many challenges for the first time:

- Customers can run all analytics over the entirety of their data estate in a simple and secure manner. Bringing robust governance—including discovery, audit and access control—to the rich mixture of diverse data in OneLake and diverse engines in Fabric is uniquely challenging, but fundamental for enterprises. This requires that we architect all engines to uniformly enforce policies expressed once at the OneLake/Fabric level. Given that Fabric engines share caches across users, the enforcement of policies must be cache-aware.
- Furthermore, as we have observed, OneLake is just one part of an enterprise's larger data estate, which also includes operational databases and document sources spread across teams and locations, and increasingly, enterprise-wide policies must be enforced and audited for regulatory compliance. Through always-on integration with Microsoft Purview, Fabric provides easy access to the global catalog and supports policies such as Purview Information Protection labels (essentially, attribute-based access control) and data lifecycle protection, extending widely deployed mechanisms from Microsoft Office to structured data for the first time.
- Finally, cross-company data sharing and collaboration are increasingly common. While shortcuts provide a foundation for sharing that can be extended to be cross-tenant, we must also support governance policies when sharing data across tenants.

Microsoft Purview: Global Policy Enforcement and Governed Sharing

In future, more security features from Purview will be available in Fabric. This includes global protection policies that will control which users can access sensitive data (Purview protection policies will be pushed down and enforced in Fabric) and data loss prevention policies to automatically check data uploaded to OneLake for sensitivity labels and apply automatic actions to reduce data leakage risks if there is a policy violation, such as notifying the security admins and automatically limiting access to sensitive data (Purview Data loss Prevention policies for OneLake).

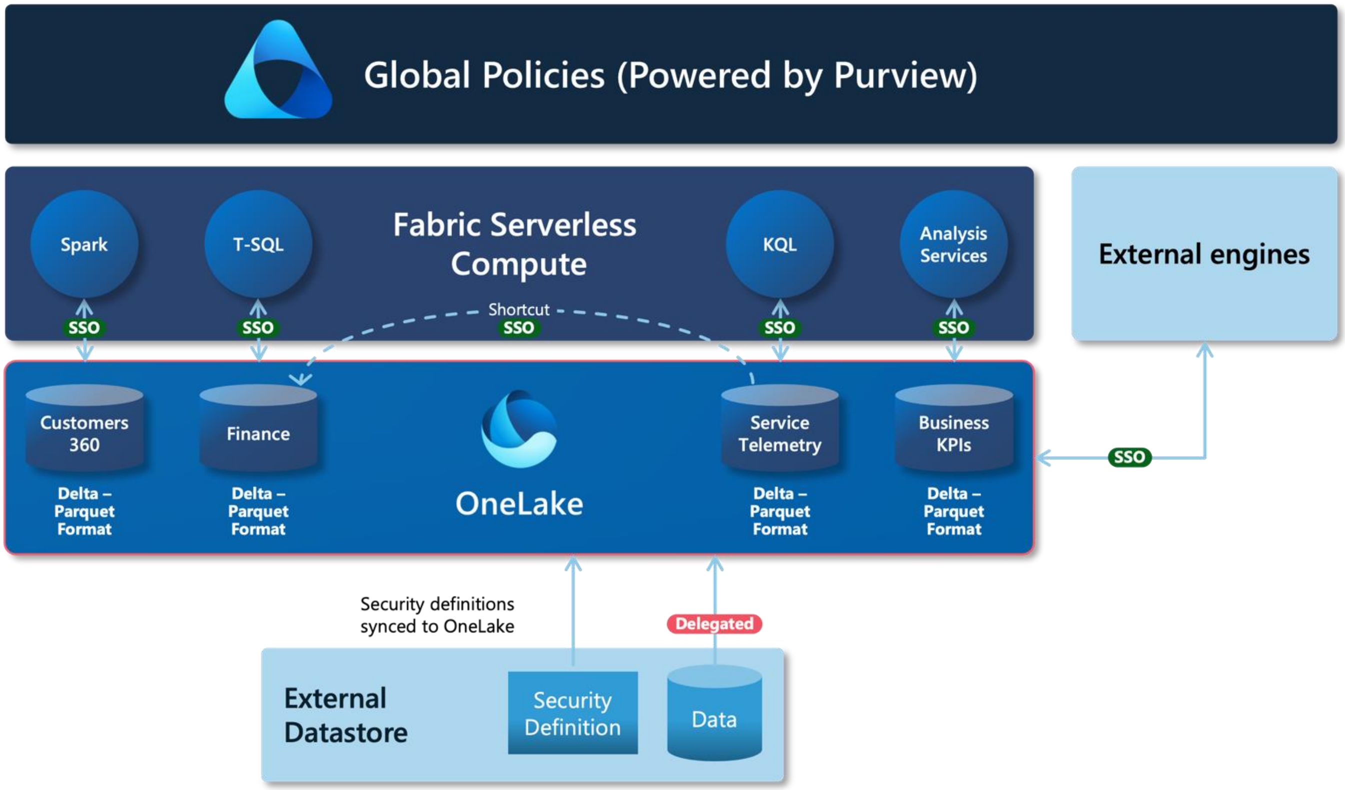

Figure 5: Purview Information Protection and Data Loss Prevention policies enforced in Fabric

Cross-Tenant Sharing

Shortcuts will be extended beyond sharing across domains within an organization to also support data sharing across organizations.

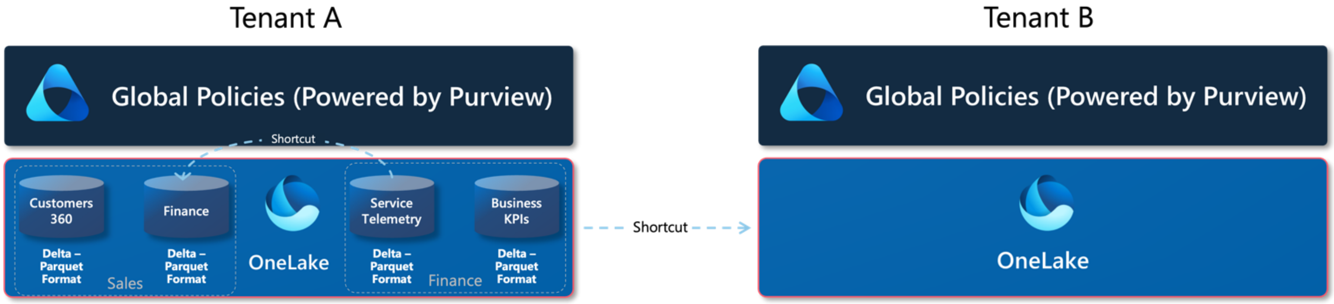

Figure 6: External sharing between Fabric tenants without data duplication

OneLake enables a truly governed sharing model. The same security and governance approach of OneLake that applies when sharing data across the domains of a tenant will extend to sharing across tenants as well. Security defined in OneLake will flow across shortcuts and global Purview policies will protect against unwanted oversharing.

Conclusion

In this paper, we have presented the vision of OneLake as an open data lake ecosystem that gives customers control over their data and greatly simplifies the task of making all enterprise data securely available for analytics, while following governance best practices and respecting constraints (e.g., organizational ownership, location, workload SLOs) that require independently managed data sources external to the analytics lake. Microsoft Fabric brings SaaS simplicity to mixing and matching a powerful suite of analytic engines to bear on data in OneLake. Integration with Microsoft Purview takes Microsoft Fabric's built-in security and governance capabilities to the next level with comprehensive data estate level catalogs, insights, global policies, and compliance support.

Acknowledgements

Bibliography

1. Incubator, Apache. XTable. [Online] https://github.com/apache/incubator-xtable.

2. Michael Armbrust, Ali Ghodsi, Reynold Xin, Matei Zaharia. Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics. [Online] https://www.databricks.com/research/lakehouse-a-new-generation-of-open-platforms-that-unify-data-warehousing-and-advanced-analytics.

3. Apache. Apache Parquet. [Online] https://parquet.apache.org.

4. Foundation, Linux. Delta Lake. Delta Lake. [Online] https://delta.io/.

5. Ashvin Agrawal, Tim Brown, Anoop Johnson, Jesús Camacho-Rodríguez, Kyle Weller, Carlo Curino, Raghu Ramakrishnan. XTable in Action: Seamless Interoperability in Data Lakes. s.l. : arXiv, 2024.

6. Apache. Iceberg. Apache Iceberg. [Online] https://iceberg.apache.org/.

7. —. Hudi. Apache Hudi. [Online] https://hudi.apache.org/.

8. Azure Data Lake Store: A Hyperscale Distributed File Service for Big Data Analytics. Ramakrishnan, R., Sridharan, B. and al., et. s.l. : SIGMOD Conference, 2017. Proceedings of ACM SIGMOD.

9. Apache. Apache Atlas. [Online] https://atlas.apache.org/#/.

10. Microsoft. Introducing Mirroring in Microsoft Fabric. [Online] https://blog.fabric.microsoft.com/en-us/blog/introducing-mirroring-in-microsoft-fabric/.

11. Dehgani, Zhamak. Data Mesh . [Online] O'Reilly. https://www.oreilly.com/library/view/data-mesh/9781492092384/.

12. Microsoft. Microsoft Purview. [Online] https://learn.microsoft.com/en-us/purview/purview.

13. —. Microsoft Fabric OneLake. [Online] https://learn.microsoft.com/en-us/fabric/onelake/.

14. —. Microsoft Fabric. [Online] https://www.microsoft.com/en-us/microsoft-fabric.

15. —. Microsoft Sharepoint. [Online] https://www.microsoft.com/en-us/microsoft-365/sharepoint/collaboration.