Microsoft



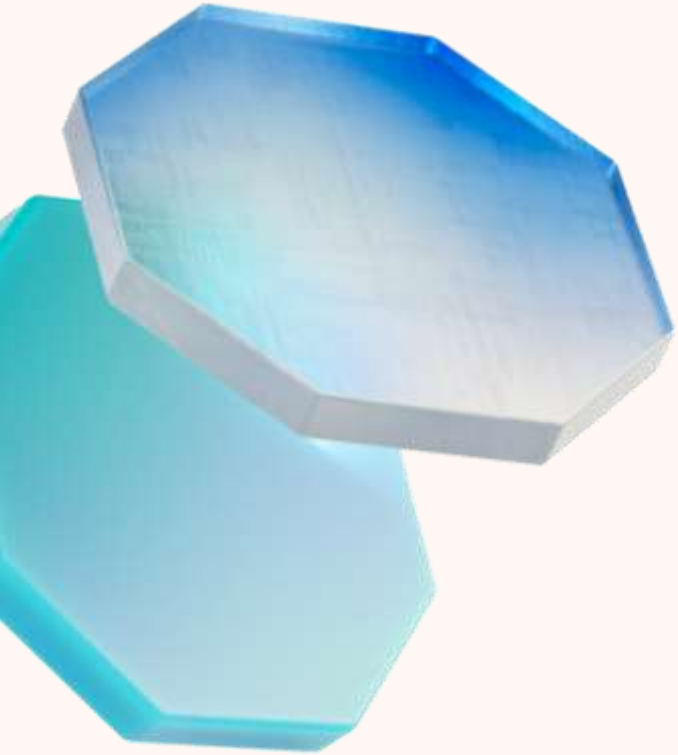
Microsoft Fabric Warehouse

A Technical Overview

With hands-on

artur.vieira@microsoft.com

LinkedIn: amvieira



Artur Vieira

Principal Program Manager – Fabric DW

Évora - Portugal

- Former Military, Consultant, Trainer, Field Engineer, Architect
- Focus on Data warehousing (MPP) and Microsoft BI
- I love motorcycles!

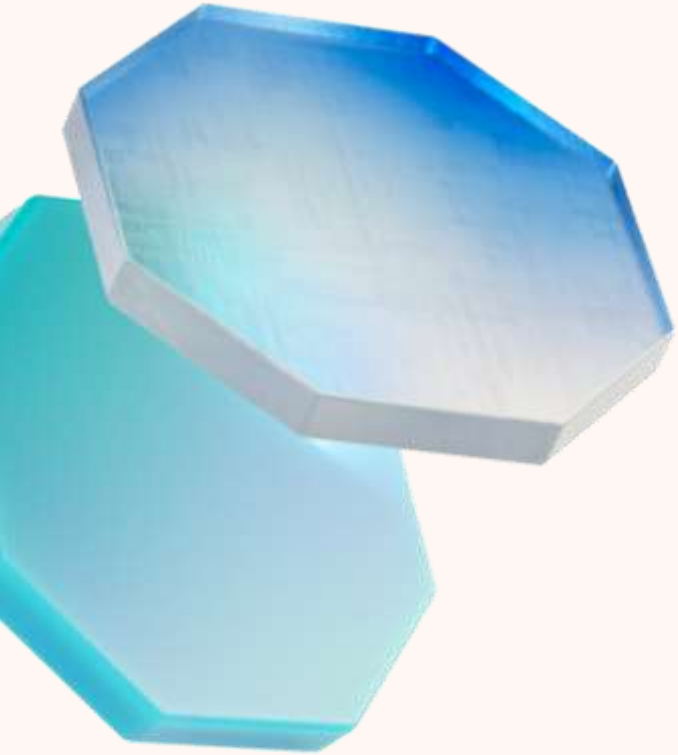
Lars.Andersen@Microsoft.com
X: @LarsADK
LinkedIn: mslars

Lars Andersen

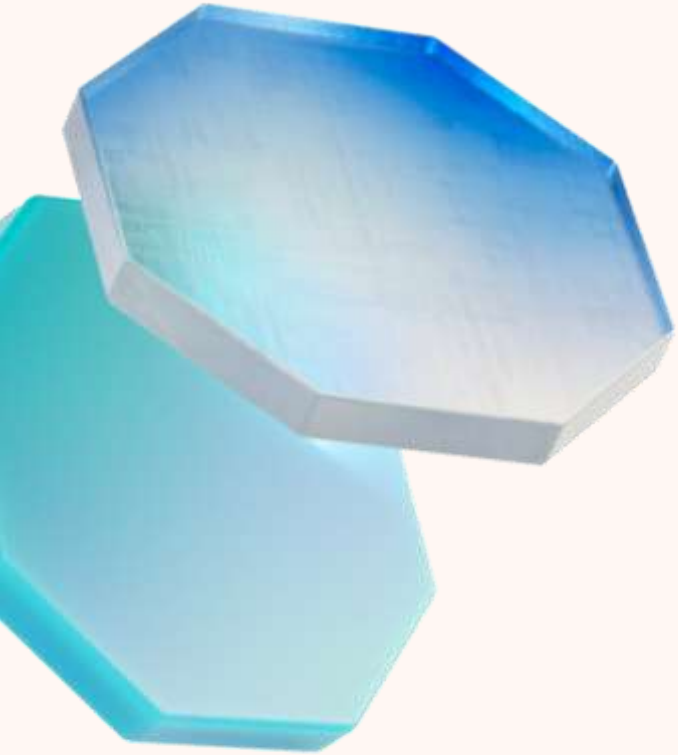
Senior Program Manager

Roskilde, Denmark

- Working with data for 20+ years
- Joined Microsoft in January 2014, and CAT in September 2021
- Previously developed DW/BI solution for toy retailer
- Love craft beer



Filip.Popovic@Microsoft.com
X: @FilipPop_MSFT
LinkedIn: popovicfilip



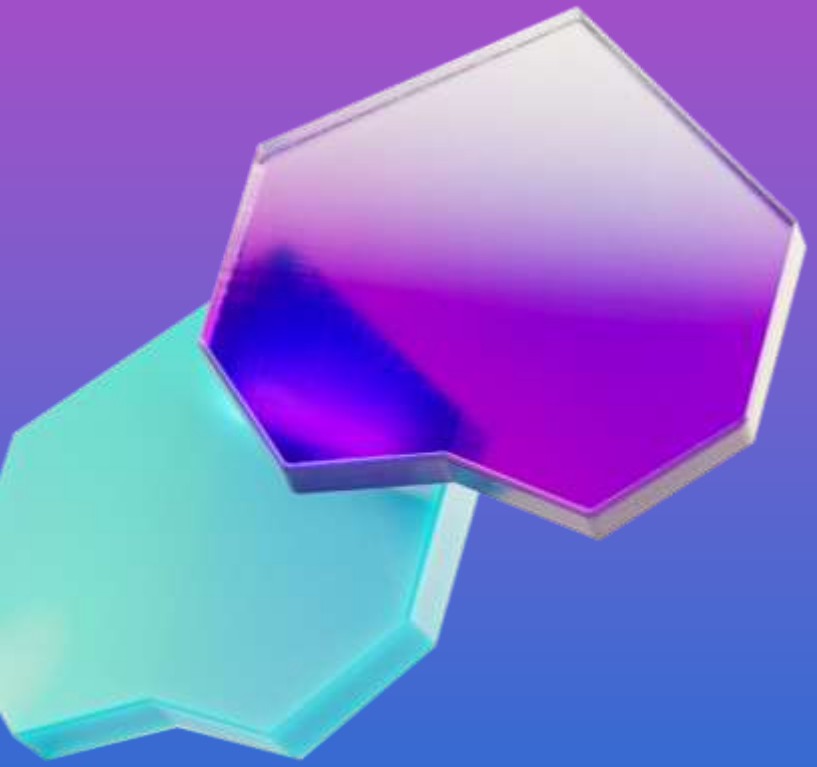
Filip Popović

Senior Program Manager

Belgrade, Serbia

- 12+ years of implementing analytics solutions using Microsoft stack prior to joining Microsoft
- Fabric Espresso host
- All about SQL performance

Logistics



- Restrooms
- Breaks and lunch
- Questions
- Workshop content

Agenda

01
Introduction to Fabric

02
OneLake

03
Medallion Architecture

04
Fabric Warehouse

05
Direct Lake

06
Warehouse Security

07
Migrations

08
Parking lot



Introduction to Fabric

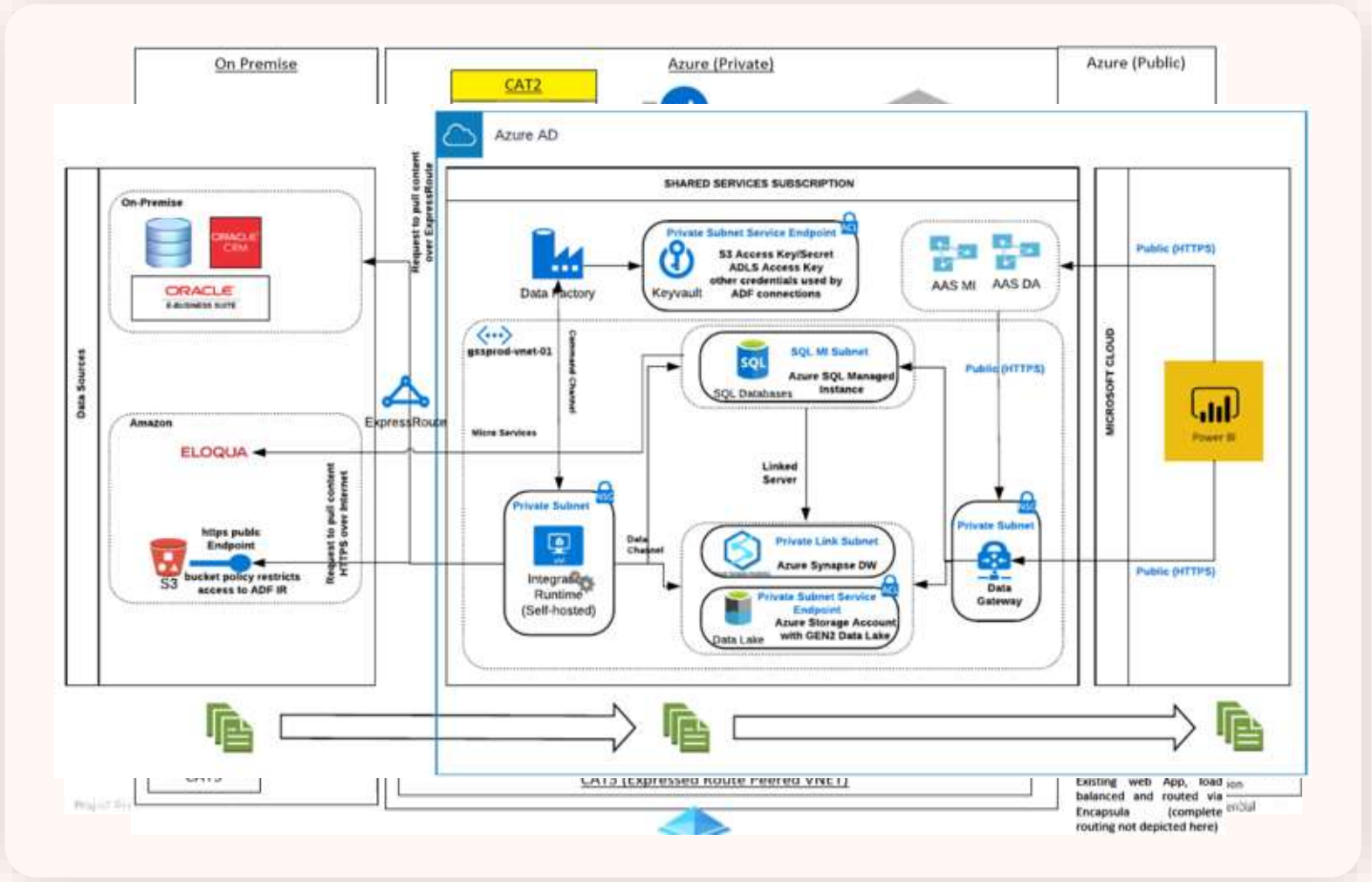
Analytics is complex and fragmented

Every project has many subsystems

Every subsystem need a different class of product

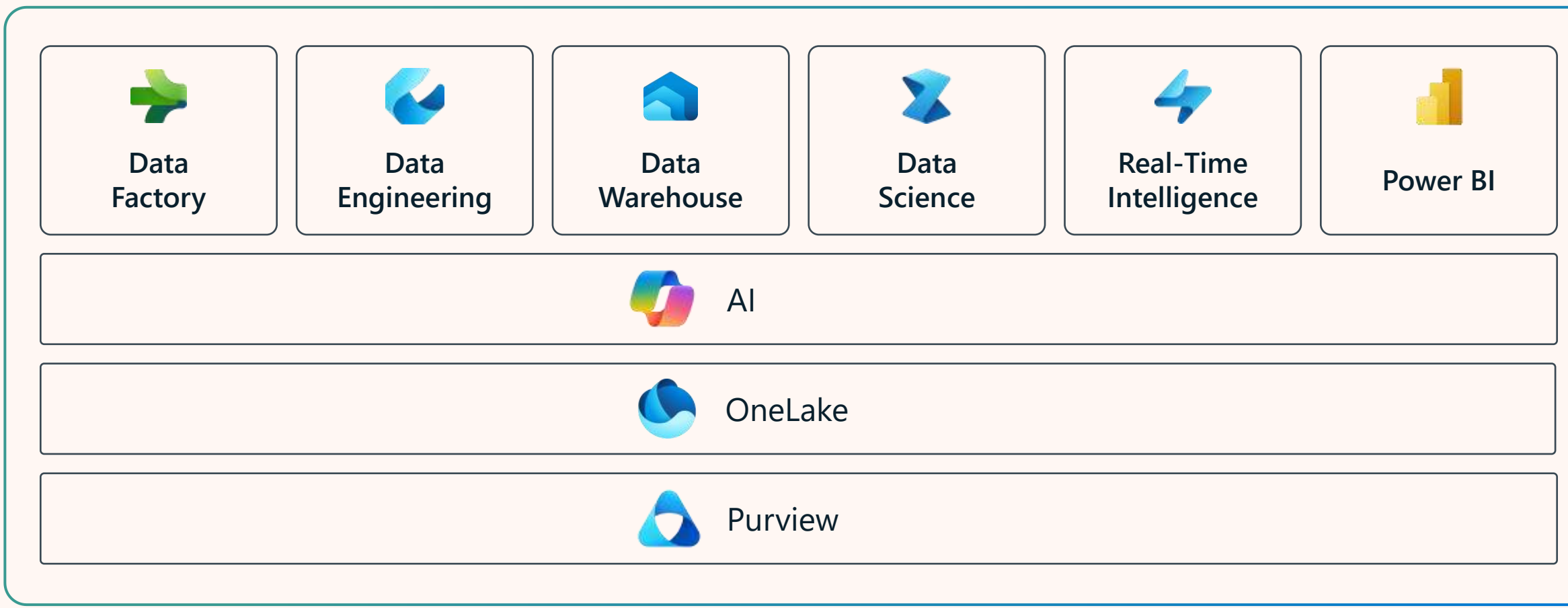
Products often comes from multiple vendors

Integration is complex, fragile and expensive



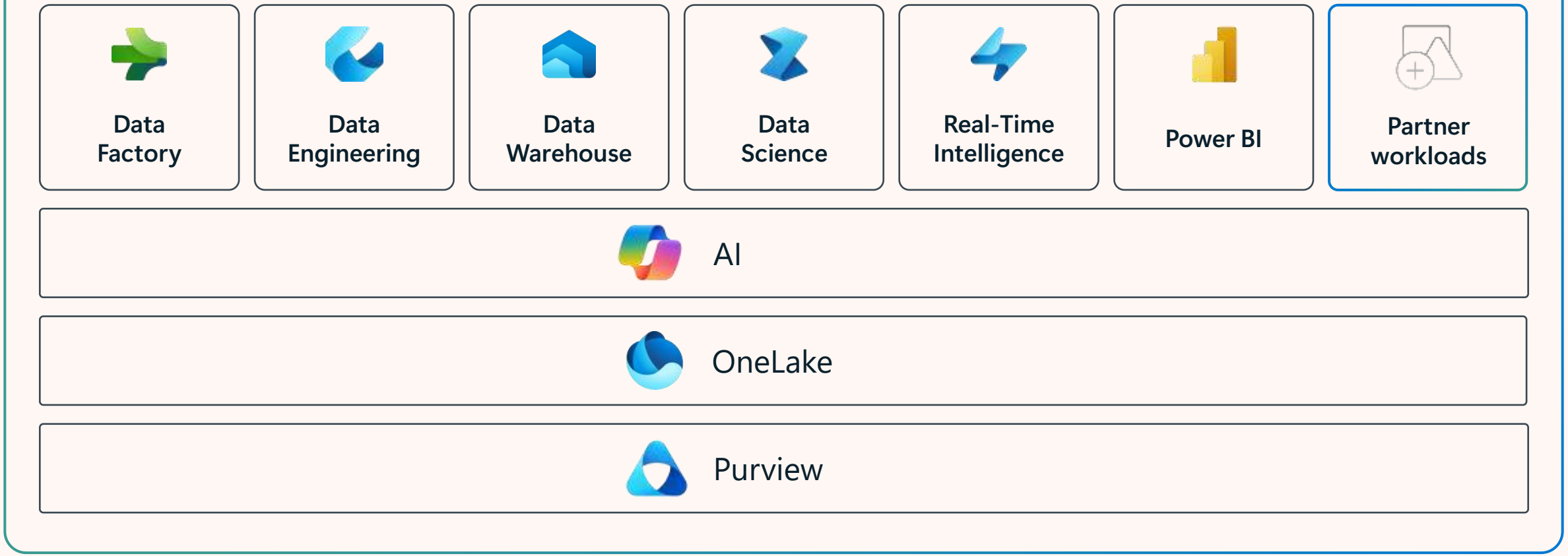
Microsoft Fabric

The unified data platform for the era of AI



Microsoft Fabric

The unified data platform for the era of AI



The data platform for the era of AI

Complete Analytics Platform

Unified product, experience, and architecture

Delivered as SaaS

Lake Centric and Open

Common SaaS data lake shared by all compute engines

Deep commitment for open formats and APIs

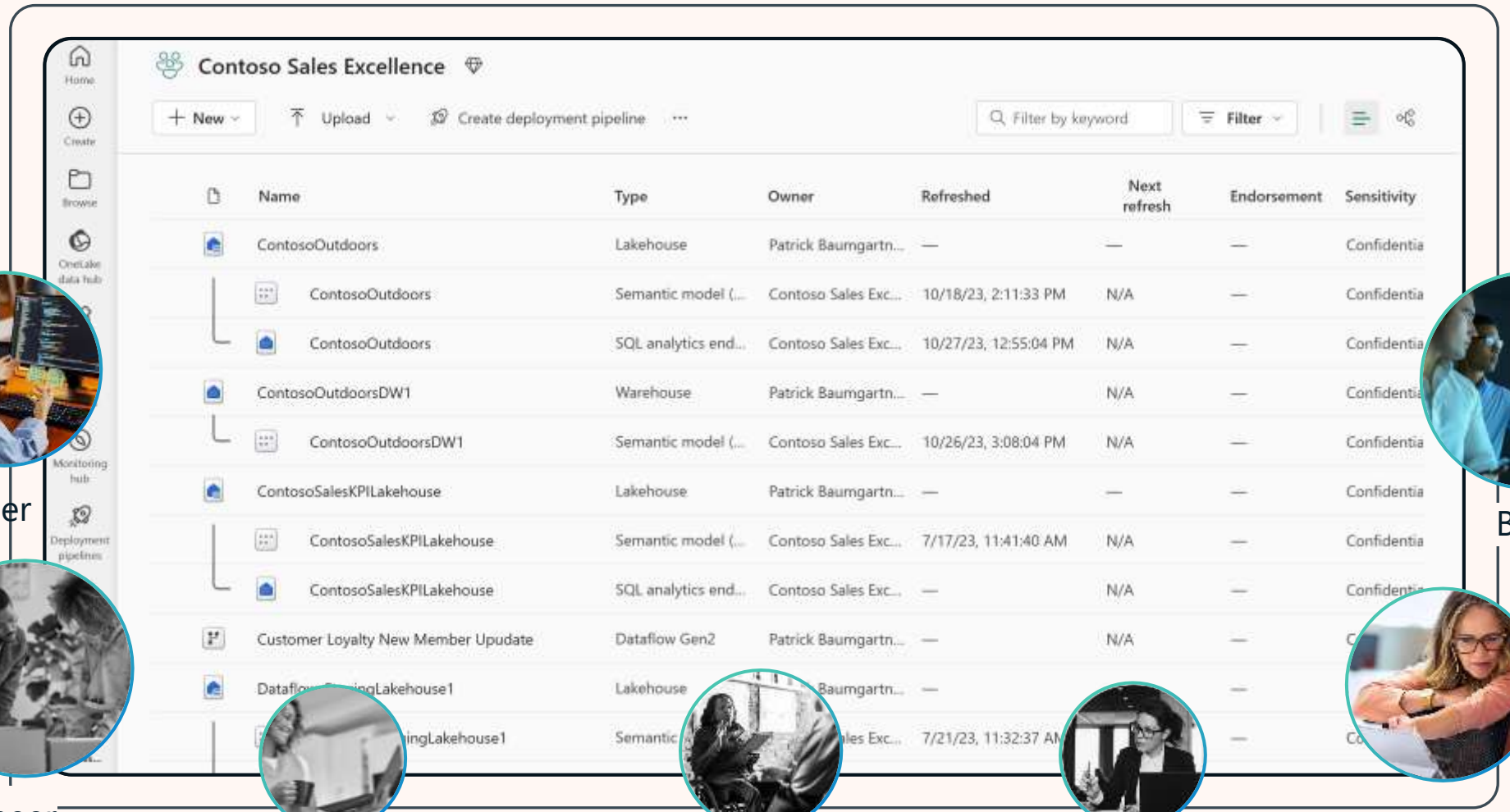
Empower Every Business User

Deliver data directly to the users in their favorite Office applications

AI Powered

Designed from the ground up for AI

Scales to the most demanding projects



SQL Developer

Business Analyst

Data Engineer

Data Scientist

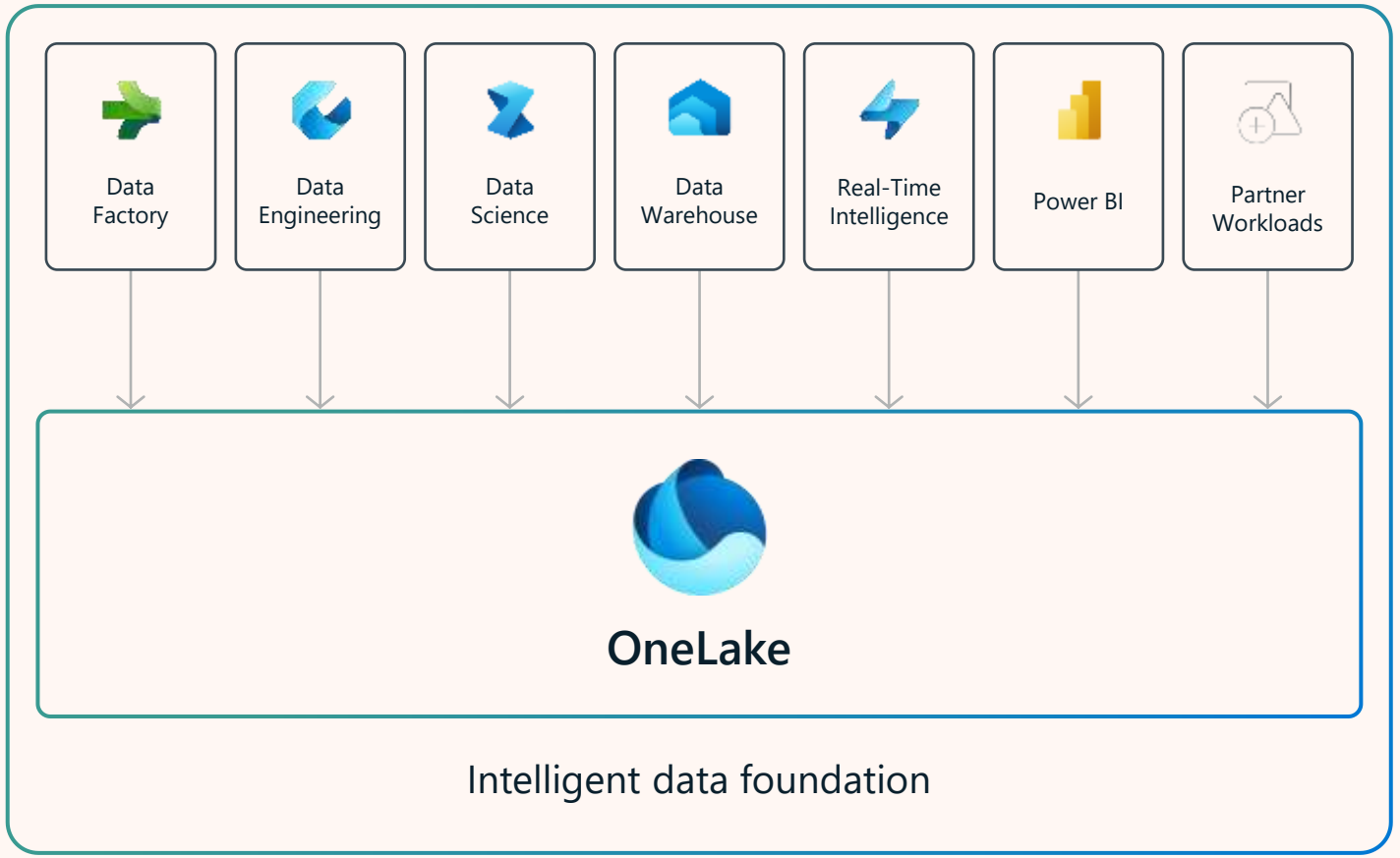
Database Admin

Developer

Project Manager

OneLake for All Data

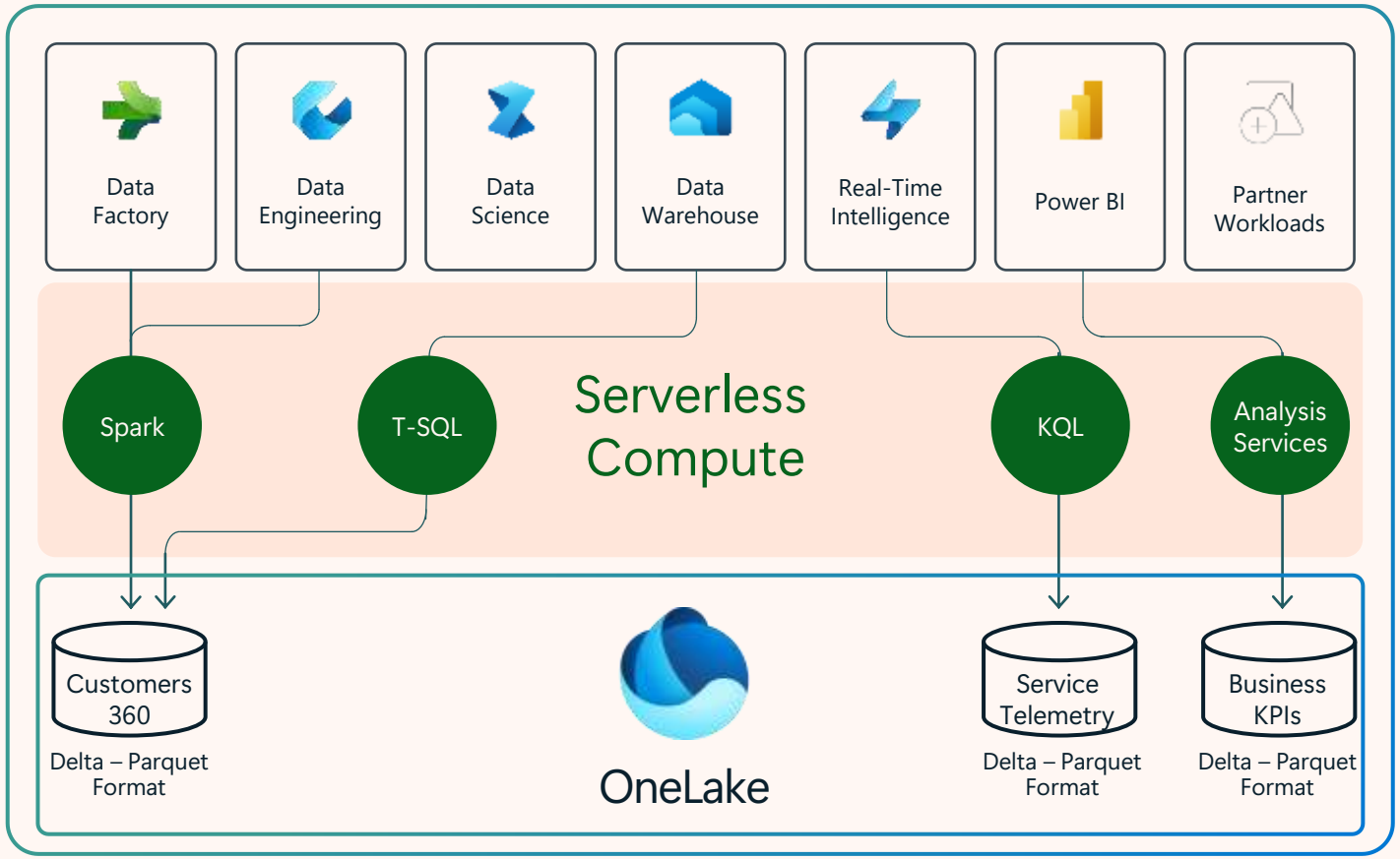
"The OneDrive for Data"



- A single SaaS lake for the whole organization
- Provisioned automatically with the tenant
- All workloads automatically store their data in the OneLake workspace folders
- All the data is organized in an intuitive hierarchical namespace
- The data in OneLake is automatically indexed for discovery, MIP labels, lineage, PII scans, sharing, governance and compliance

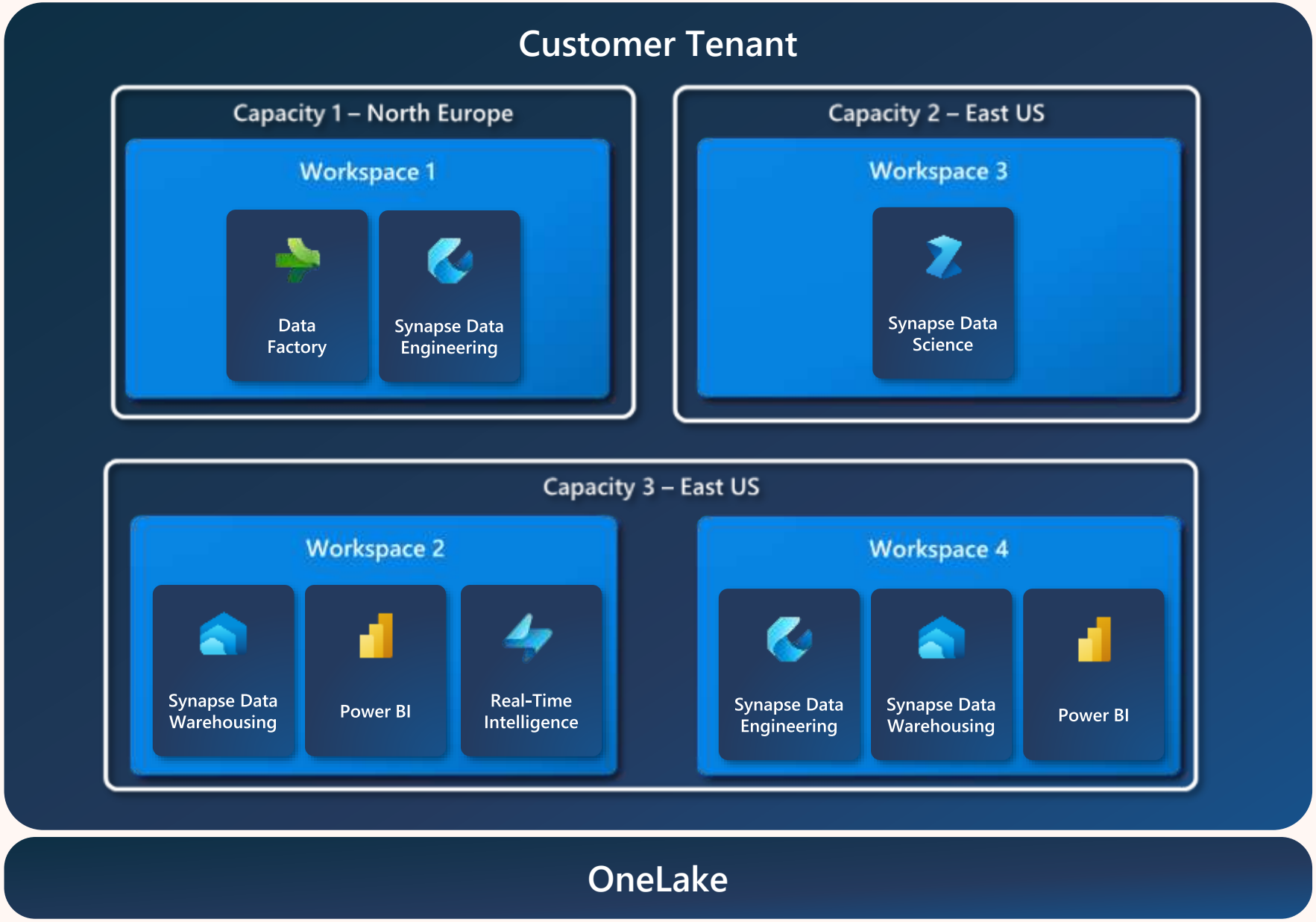
OneLake for All Data

"The OneDrive for Data"



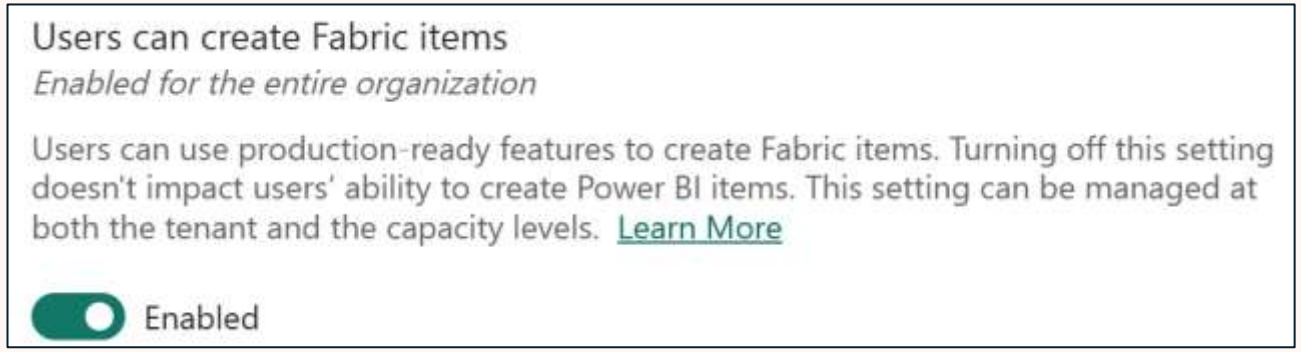
- A single SaaS lake for the whole organization
- Provisioned automatically with the tenant
- All workloads automatically store their data in the OneLake workspace folders
- All the data is organized in an intuitive hierarchical namespace
- The data in OneLake is automatically indexed for discovery, MIP labels, lineage, PII scans, sharing, governance and compliance

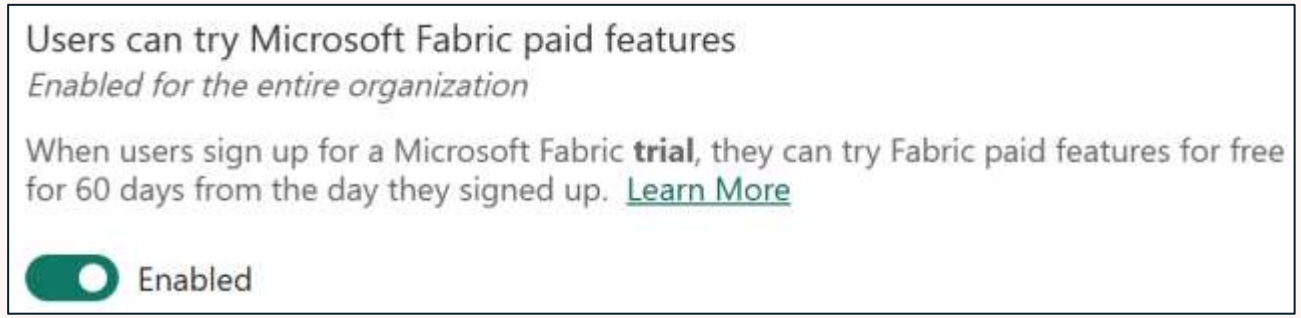
Fabric topology and organization


Customer Tenant
Capacity 1 – North Europe
Workspace 1
Data Factory
Synapse Data Engineering
Capacity 2 – East US
Workspace 3
Synapse Data Science
Capacity 3 – East US
Workspace 2
Synapse Data Warehousing
Power BI
Real-Time Intelligence
Workspace 4
Synapse Data Engineering
Synapse Data Warehousing
Power BI
OneLake

Getting started in your environment

1. Enable Fabric at the tenant level.
   - Override the tenant level setting at the capacity
   - Enable for only specific security groups
2. Get capacity
   - Start a 60-day free trial
   - Use existing P SKU
   - Provision an F SKU
3. Create a workspace and assign a capacity.
4. Build something cool!







Capacities

Capacities are to Fabric what CPUs are to PCs

Personal Computing

When you purchase a PC you choose the number of CPU cores. The more CPU cores the more load the PC can handle.



The CPU cores are dynamically shared across all applications with no need to pre-allocate by app.



The total consumption of the CPU across all the apps cannot exceed the number of cores. CPU overload causes a slowdown.



Fabric Capacities

In Fabric, you provision a Capacity with a number of "capacity units". The more capacity units provisioned, the more compute load handled.

Unlike the PC, capacity units can be scaled up or down as needed.

The capacity units are dynamically shared across all the Fabric workloads, with no pre-allocation necessary.

A single capacity can simultaneously drive BI, DW, Spark, ML and every other compute engine in Fabric

The total consumption of the capacity across all the workloads cannot exceed the capacity units provisioned.

Overloading the capacity will throttle it (slow down).

Auto scale can dynamically increase the available compute units avoiding the slowdown.

Capacities are a shared resource

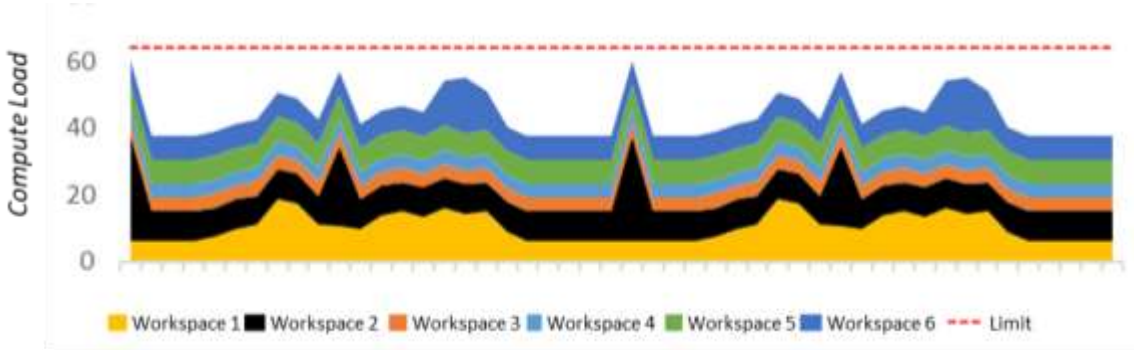
Shared across workloads
A single capacity is providing the compute power for all Fabric workloads.

There is no need to allocate compute for each workload separately.



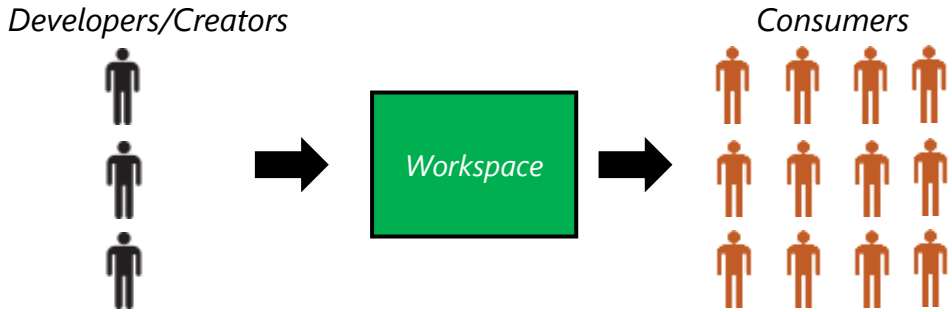
Shared Across Projects
A single capacity typically supports dozens of separate projects simultaneously, each managed in its own workspace.

It is rare to have a capacity dedicated to a single project



Shared across users
For each project, many developers will share a workspace where collaborative development and consumption at scale is managed.

Each creator can provision any artifact and run any job without the need for any pre-approval or planning

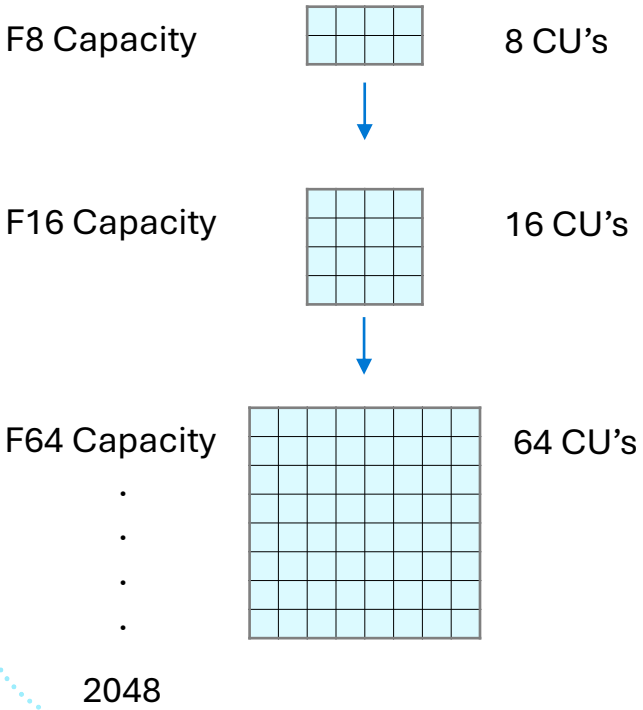


Capacities are flexible building blocks for growth

Capacities can be configured in endless ways to meet scale, usage and governance requirements while tuning to minimize TCO and performance goals

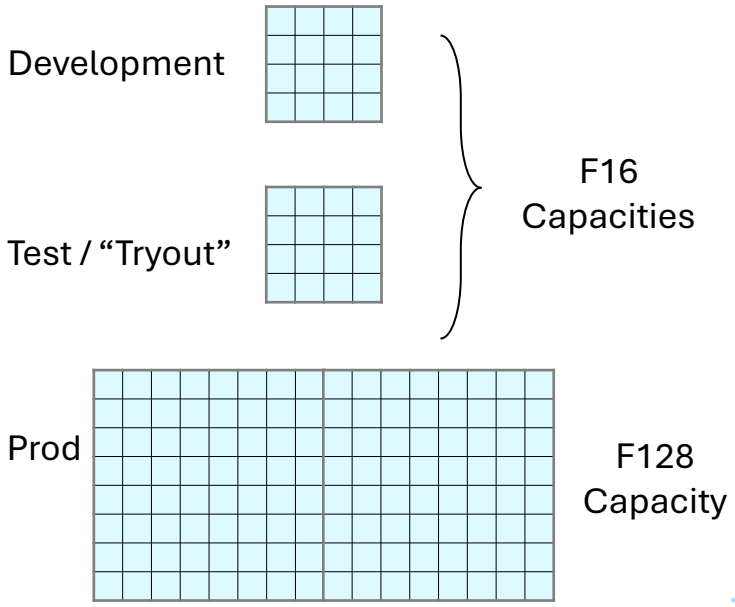
Scale Vertically

Increased capacity size provides more throughput



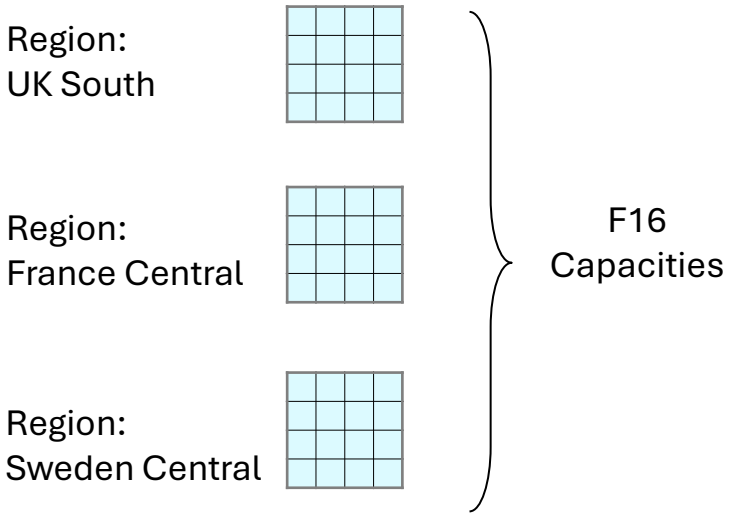
Scale Horizontally

Scale horizontally using the benefits of modular design for hardened isolation and governance



Regional Availability

Use different capacities for different regions to support GDPR / Data residency requirements



Smoothing intro and benefits

Load stabilization	Smoothing helps capacities self-stabilize by flattening large spikey loads into a smooth load profile, eliminating temporal spikes
Background operations	Large/scheduled Jobs usage (not execution) are smoothed over 24 hours, eliminating the need to decide the timing and order of job execution
Interactive operations	Interactive operations smoothed over several minutes, preventing a single user with a very demanding query from hogging the entire capacity



What is Bursting?

Job acceleration

Bursting provides extra compute resources to jobs and queries to accelerate their completion

Go beyond

The extra resources of bursting allow jobs to **utilize far more resources than "face value"**

Instead of running a job on 64 CU and completing in 60 seconds, bursting could use 256 CUs to complete the job in 15 seconds.

Same amount of work, just completed faster

No hassle, No overload

Bursting is automatic when the system reasons it can accelerate the job by applying extra resources. No settings are required.

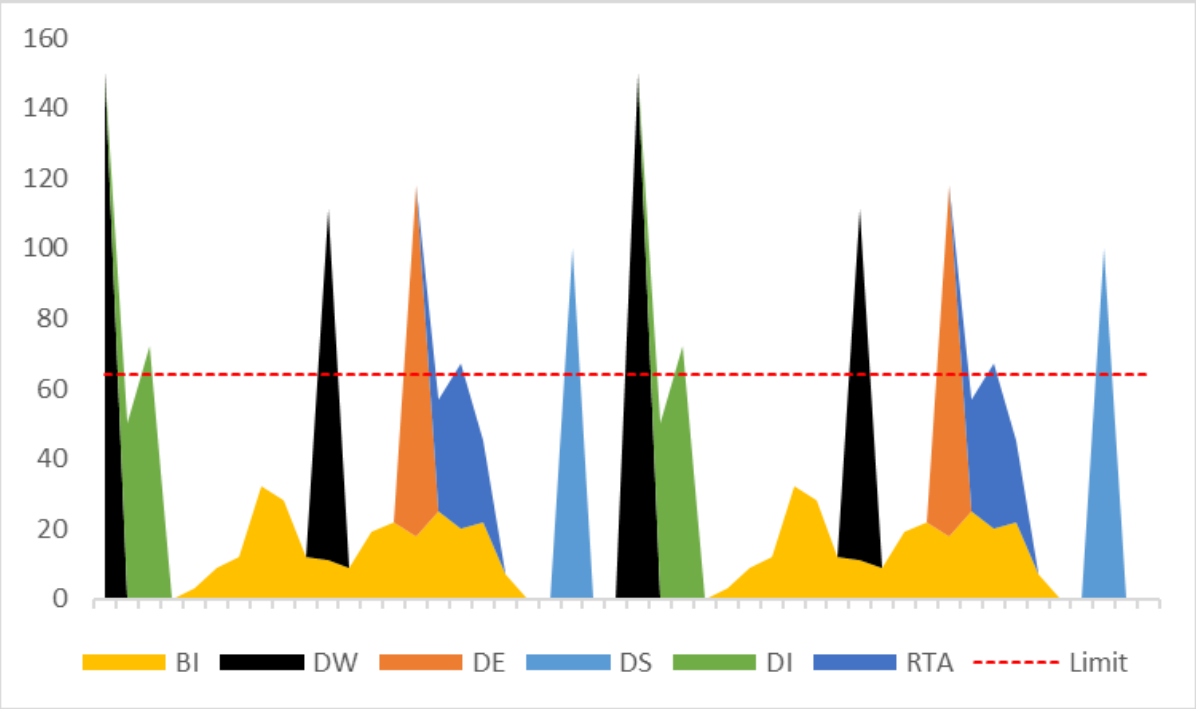
Bursting never causes an overload as the *smoothing* mechanism will always flatten the resource burst

Bursting and smoothing | before and after

Looking at an example of a capacity, running multiple workloads over a couple of days…

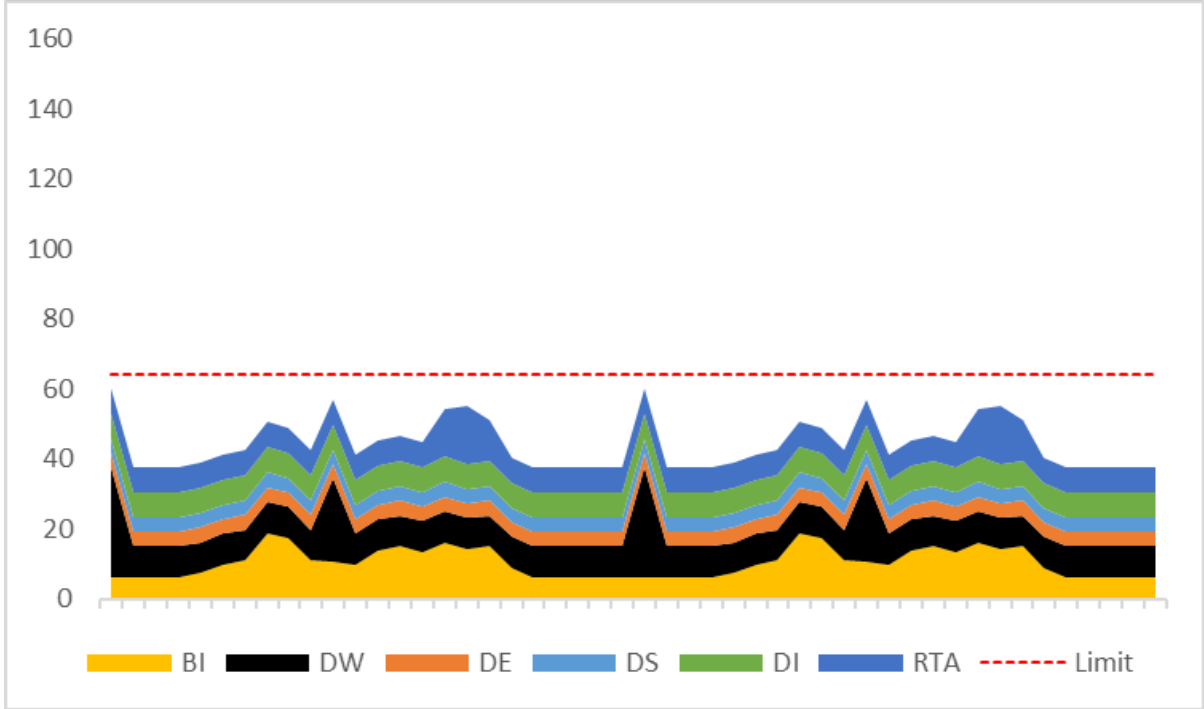
Before Smoothing

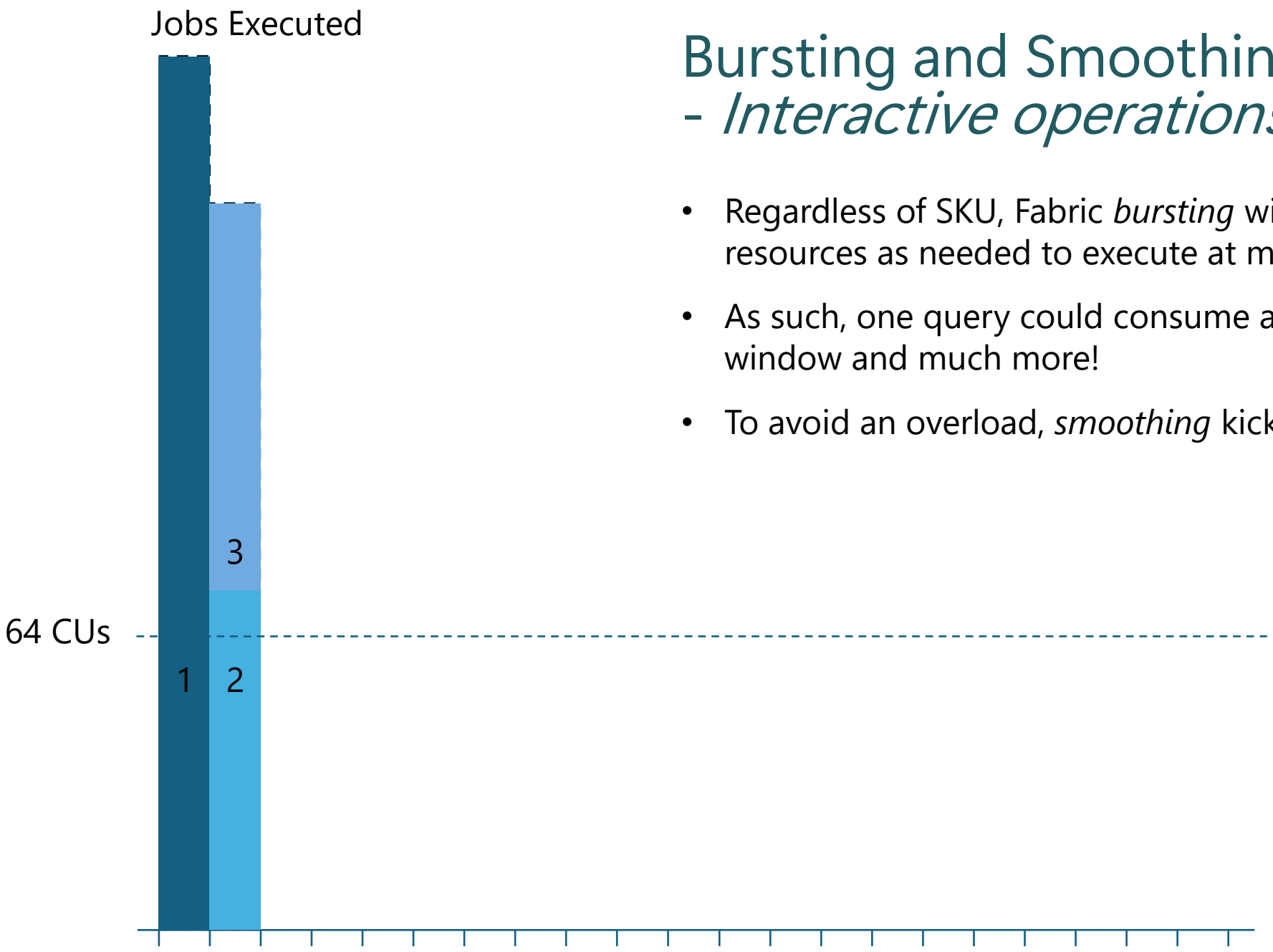
- Actual load as executed on the capacity before smoothing
- *Bursting* accelerates jobs execution by resource boosting
- The capacity could be overloaded 25% of the time
- Some of the overloads are more than 2x the limit
- There are periods of no/low usage



After Smoothing

- Shows the reported load (not runtime execution) against the capacity limits
- There is NO overload, and consumption is more stable
- The smoothing of usage fills in gaps



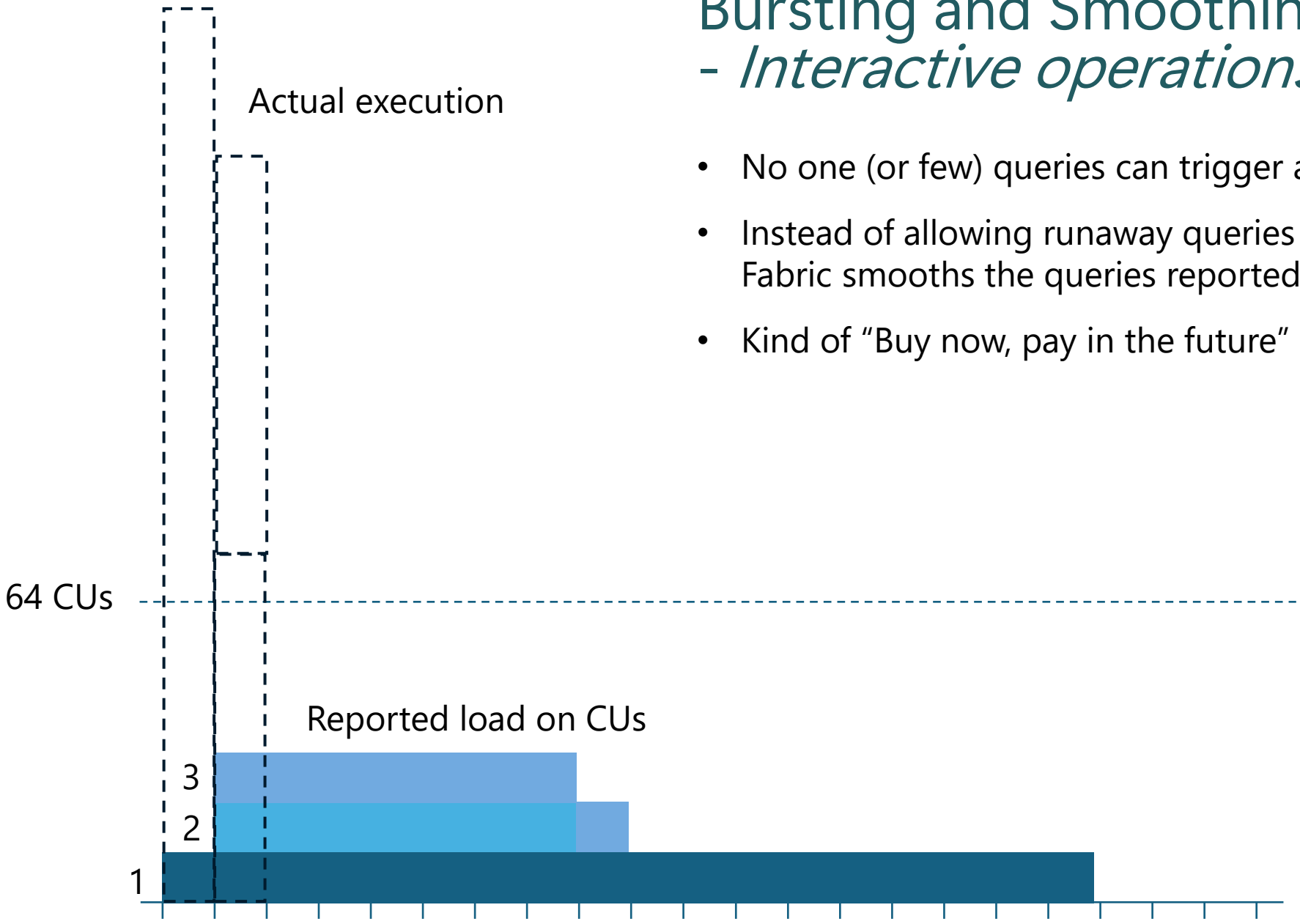


Bursting and Smoothing - *Interactive operations*

- Regardless of SKU, Fabric *bursting* will automatically allocate resources as needed to execute at maximum performance
- As such, one query could consume all the quota of a single time window and much more!
- To avoid an overload, *smoothing* kicks in

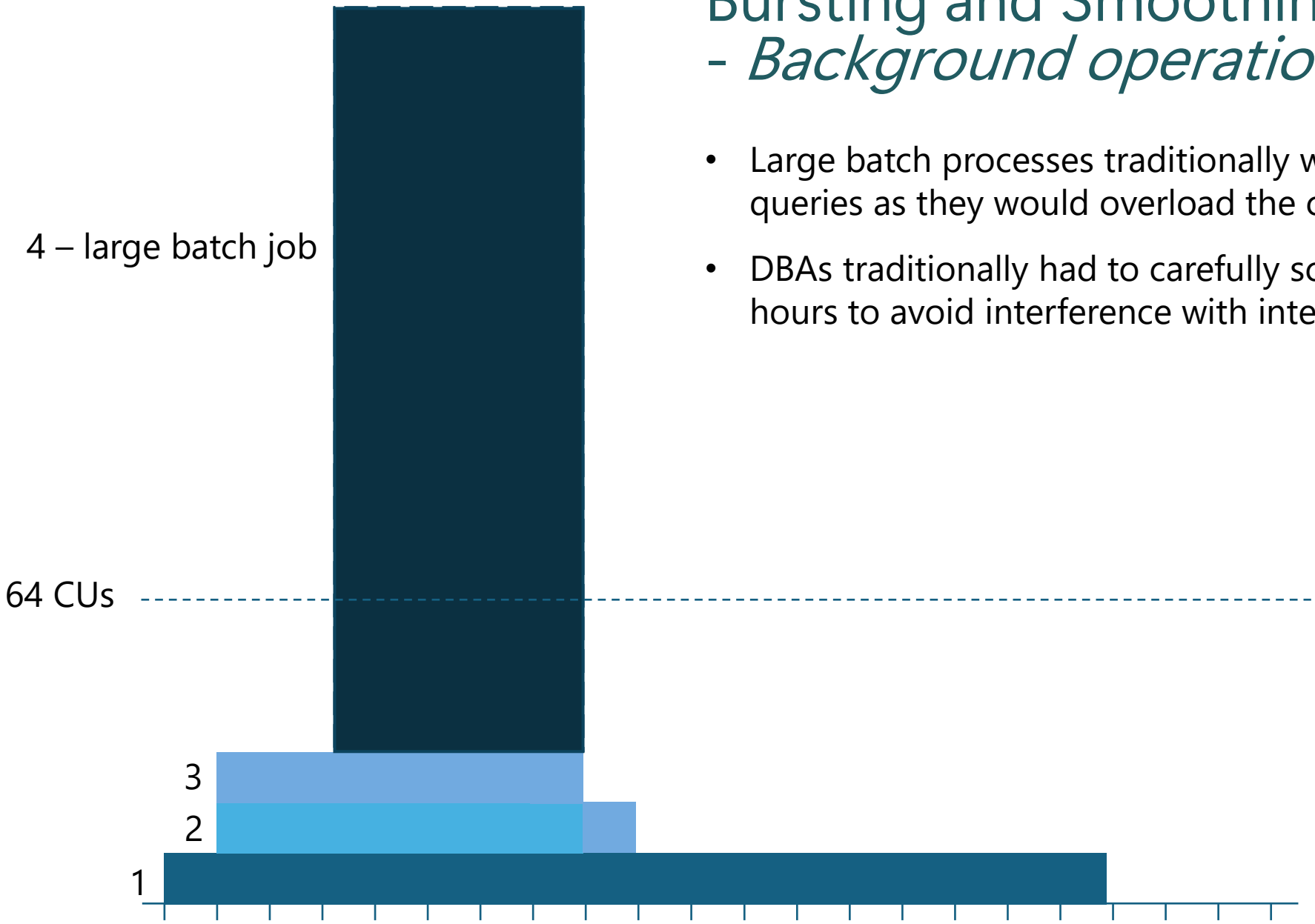
Bursting and Smoothing - *Interactive operations*

- No one (or few) queries can trigger an overload
- Instead of allowing runaway queries to create a local overload, Fabric smooths the queries reported usage to future time windows
- Kind of “Buy now, pay in the future” installment plan



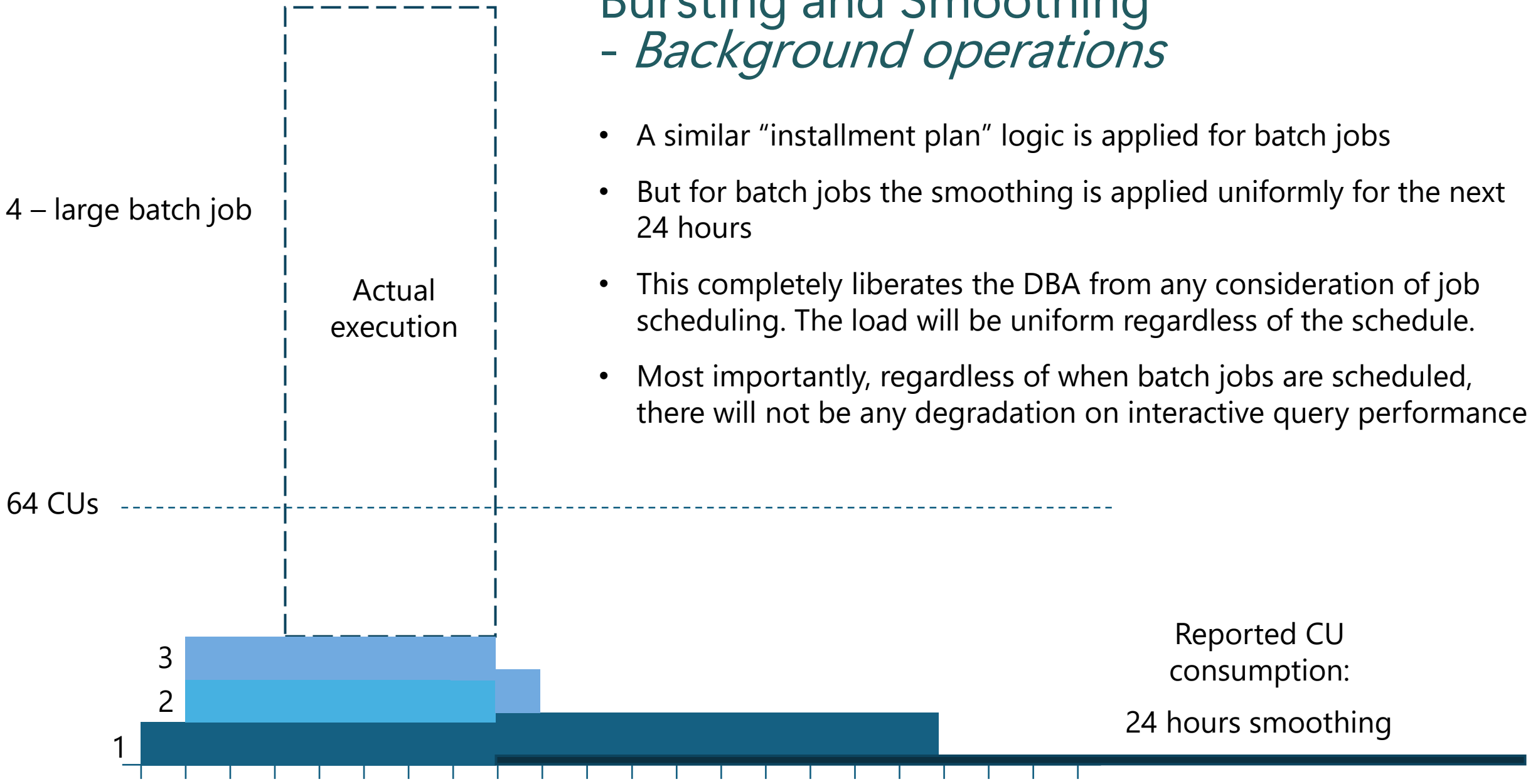
Bursting and Smoothing - *Background operations*

- Large batch processes traditionally were a threat to interactive queries as they would overload the compute resource
- DBAs traditionally had to carefully schedule these jobs to off-hours to avoid interference with interactive user experiences



Bursting and Smoothing - *Background operations*

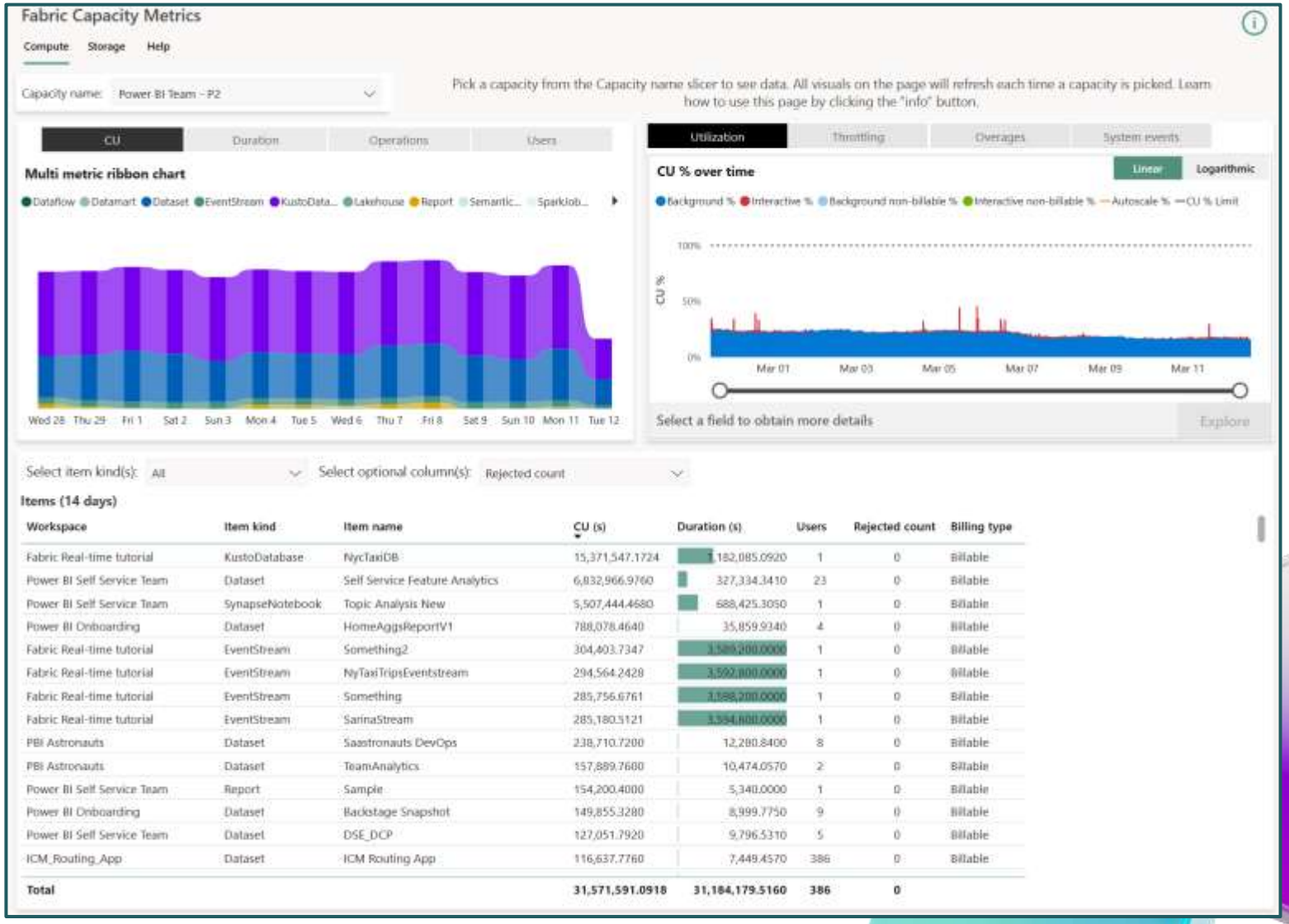
- A similar “installment plan” logic is applied for batch jobs
- But for batch jobs the smoothing is applied uniformly for the next 24 hours
- This completely liberates the DBA from any consideration of job scheduling. The load will be uniform regardless of the schedule.
- Most importantly, regardless of when batch jobs are scheduled, there will not be any degradation on interactive query performance



Capacity Metrics

Monitor Capacities and Plan capacity scale-up with confidence

- Tenant wide visibility into capacity usage for all Fabric experiences
- Identify resource usage trends and their impact to autoscale & throttling
- View preview workload usage alongside production workloads to make data-driven capacity sizing decisions



Warehousing capacity details

- Warehouse operations are background operations, except Modeling (Power BI) operations
- Ingestion and Query operations are isolated
- Throttling policies

Policy Limits		
Usage <= 10 minutes	Overage protection	Jobs can consume 10 minutes of future capacity use without throttling.
10 minutes < Usage <=60 minutes	Interactive Delay	User-requested interactive jobs are delayed 20 seconds at submission.
60 minutes < Usage <= 24 hours	Interactive Rejection	User requested interactive type jobs are rejected.
Usage > 24 hours	Background Rejection	All new jobs are rejected from execution.

Warehousing capacity details

- Warehouse operations are background operations, except Modeling (Power BI) operations
- Ingestion and Query operations are isolated
- Throttling policies

- Documentation
  - Burstable capacity
  - Pause and resume in Synapse Data Warehouse
  - Smoothing and throttling in Fabric Data Warehousing
  - Workload management

Getting access to Fabric user for workshop

Notice this Fabric user will be removed after the workshop

–

make your own copy of content, you want to keep after the workshop

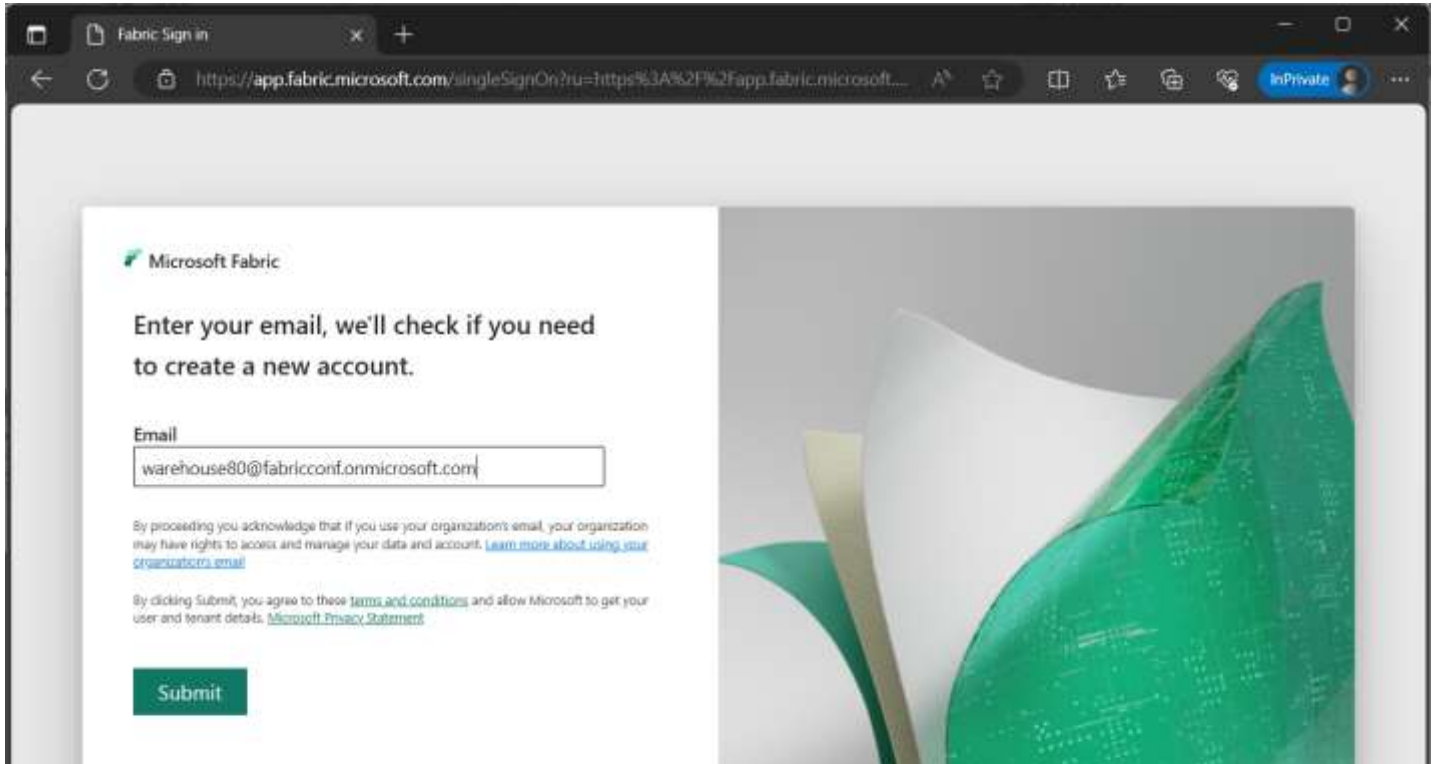
aka.ms/DPNextStep

HUONE Kastrup

Pw: youareawesome

Get started

- Create a new profile in your browser, or use InPrivate window
- Go to https://fabric.microsoft.com



Initial login

- Login with the email address *<first name>.<last name>@fabricconf.onmicrosoft.com*
  E.g. john.doe@fabricconf.onmicrosoft.com or
  karen.marie.smith@*fabricconf.onmicrosoft.com*
  Password is Hell0Fabric (0 is the number zero)

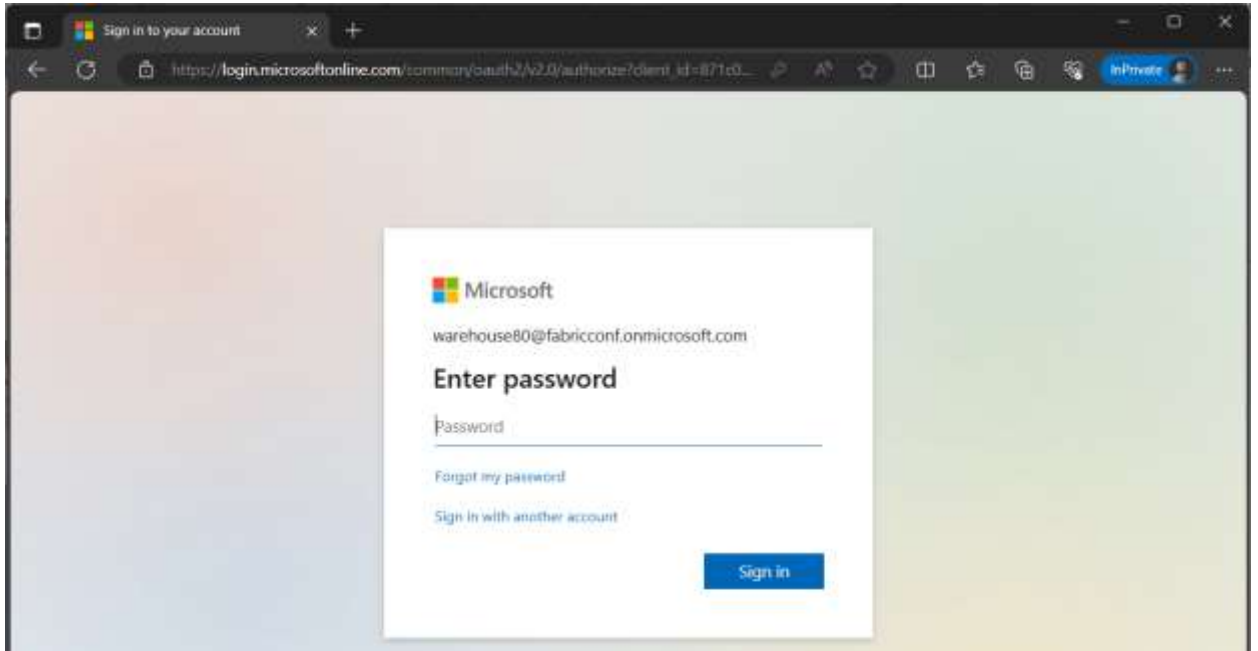
Special characters:

Æ = A

Ø = O

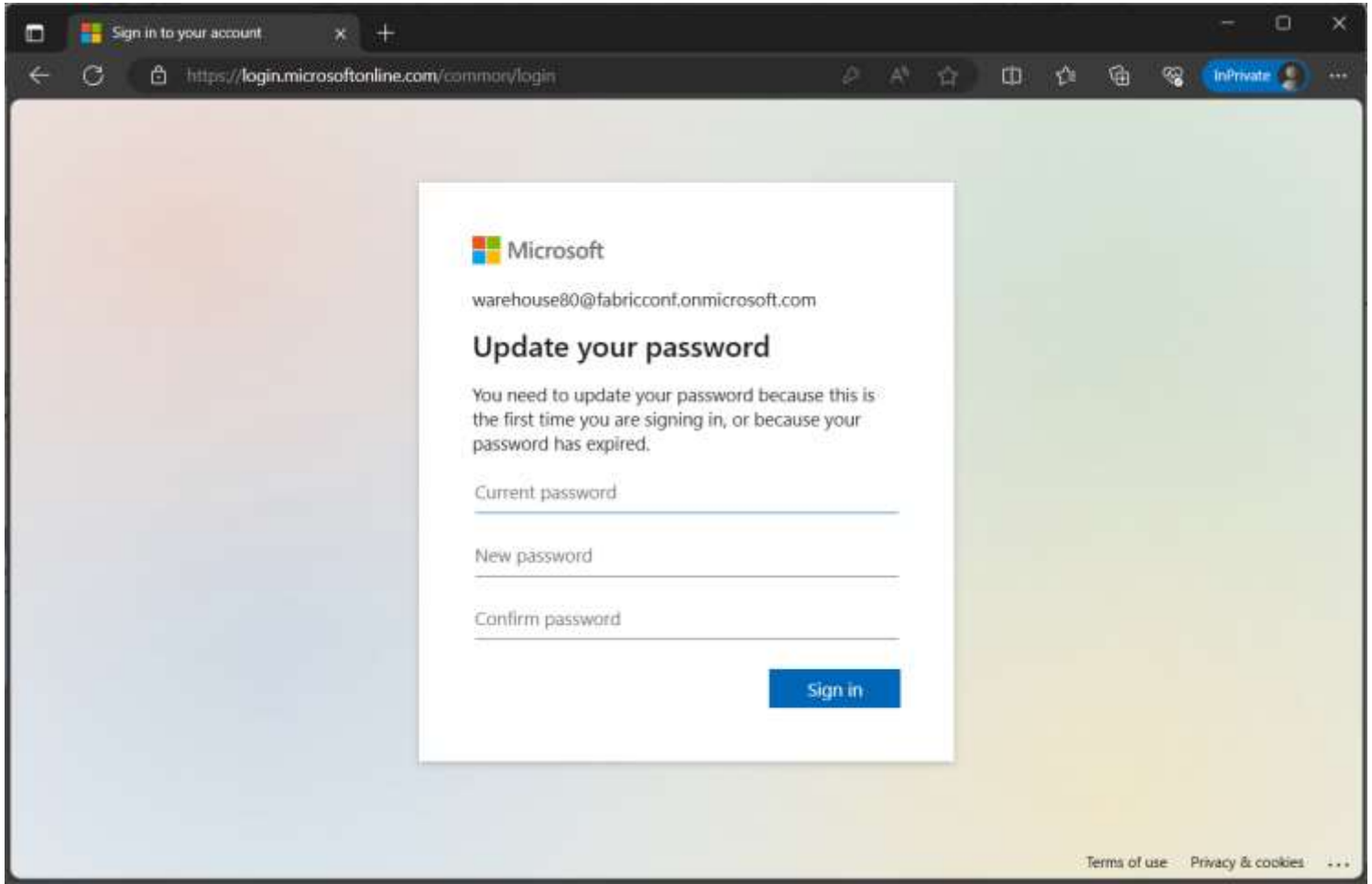
Ä = A

É = E



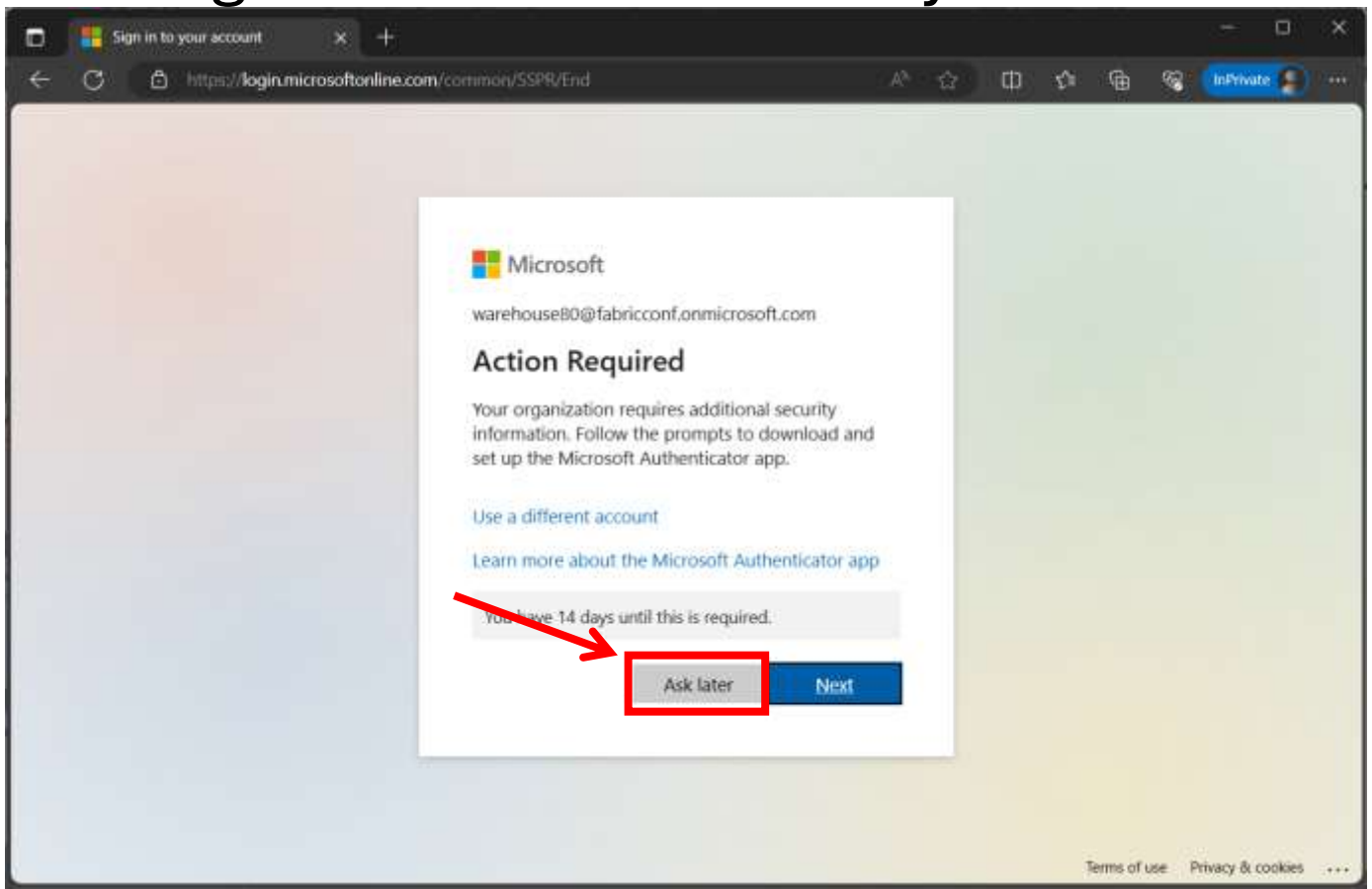
Change password

- The first time you login, you are prompted to change the password



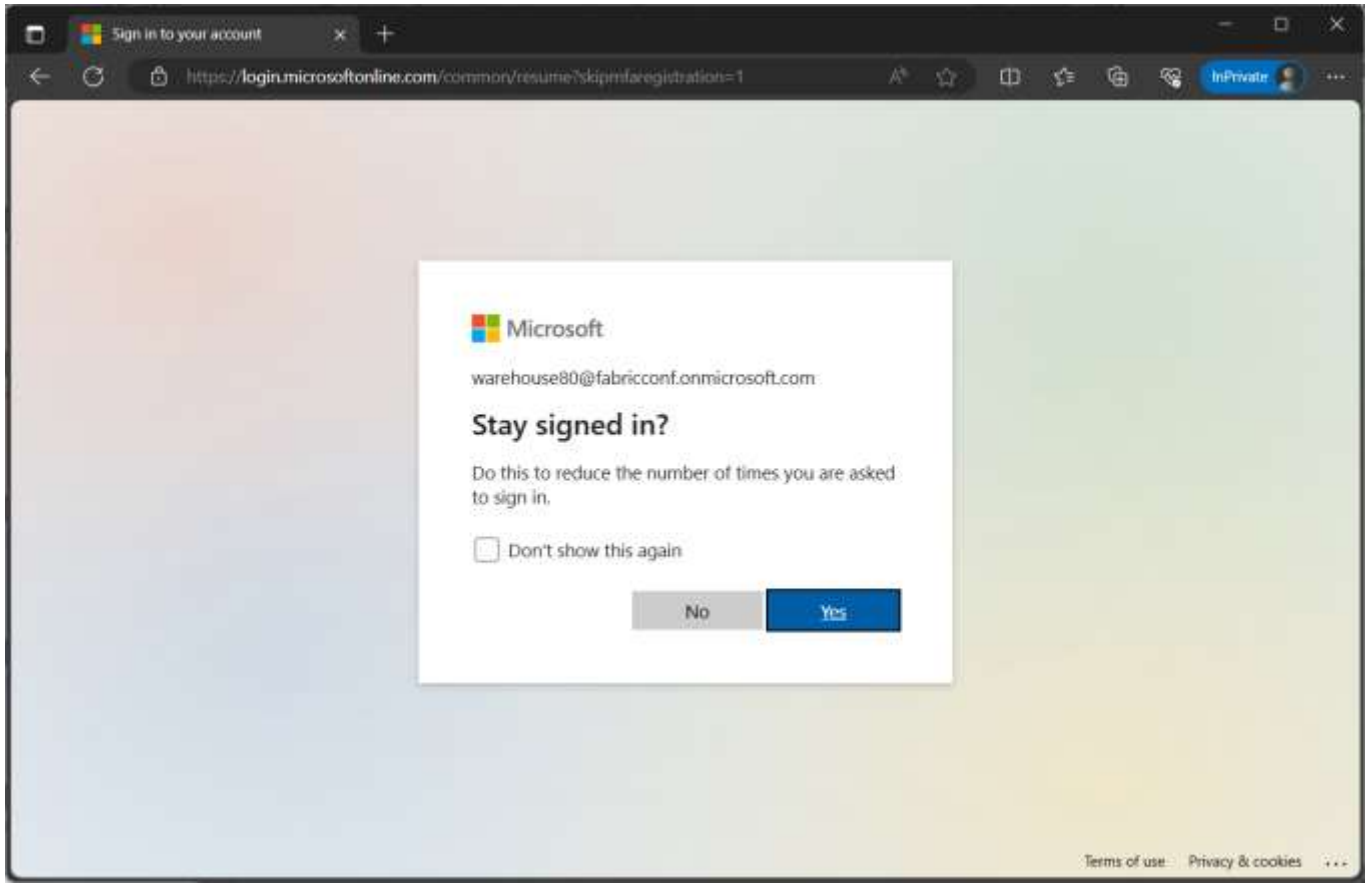
Do not setup additional security information

- Once you selected your own password, you should click “Ask later” when asking for additional security information



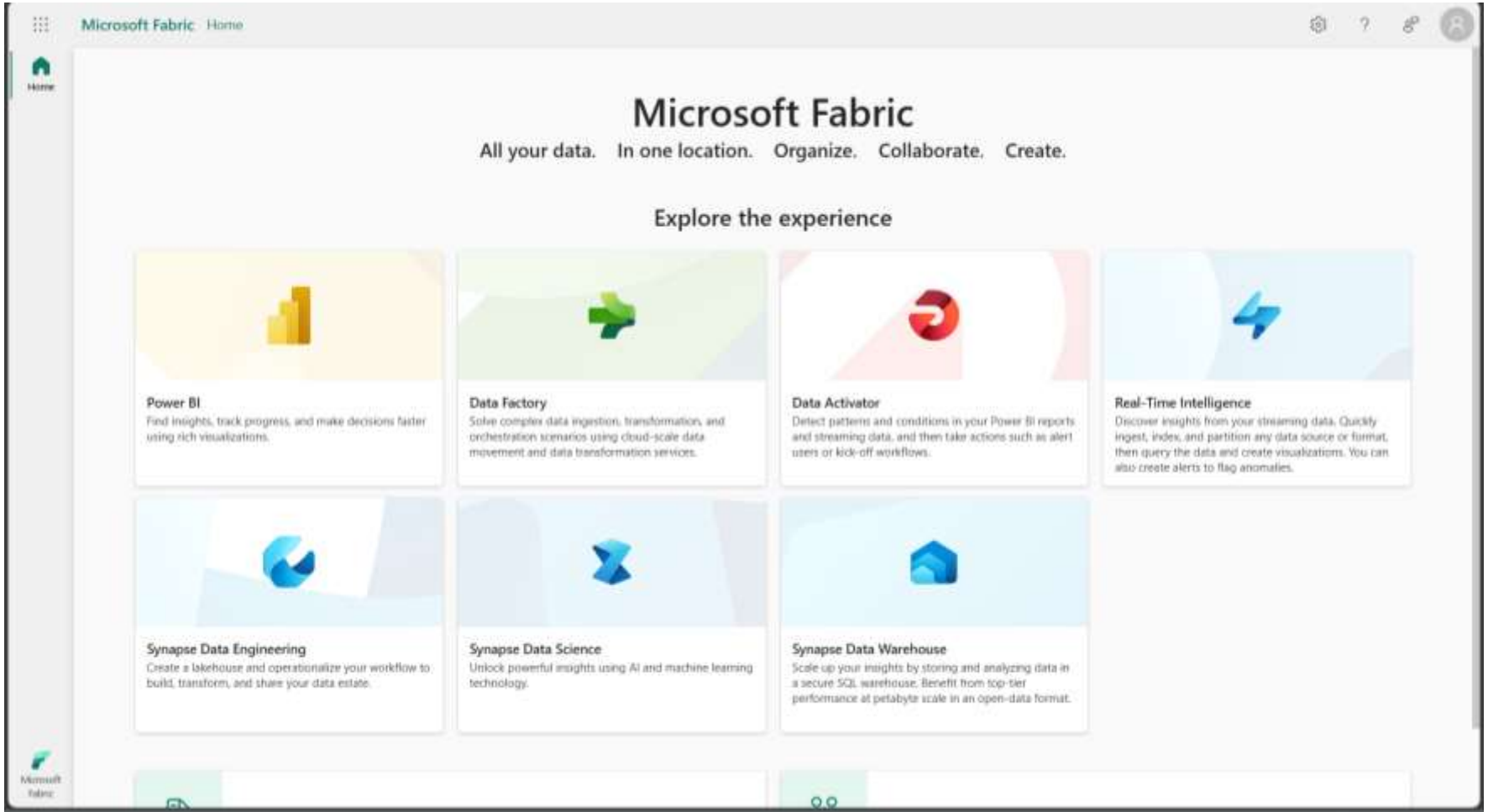
Stay signed in?

- You may be asked to stay signed in and you can select whatever option you prefer



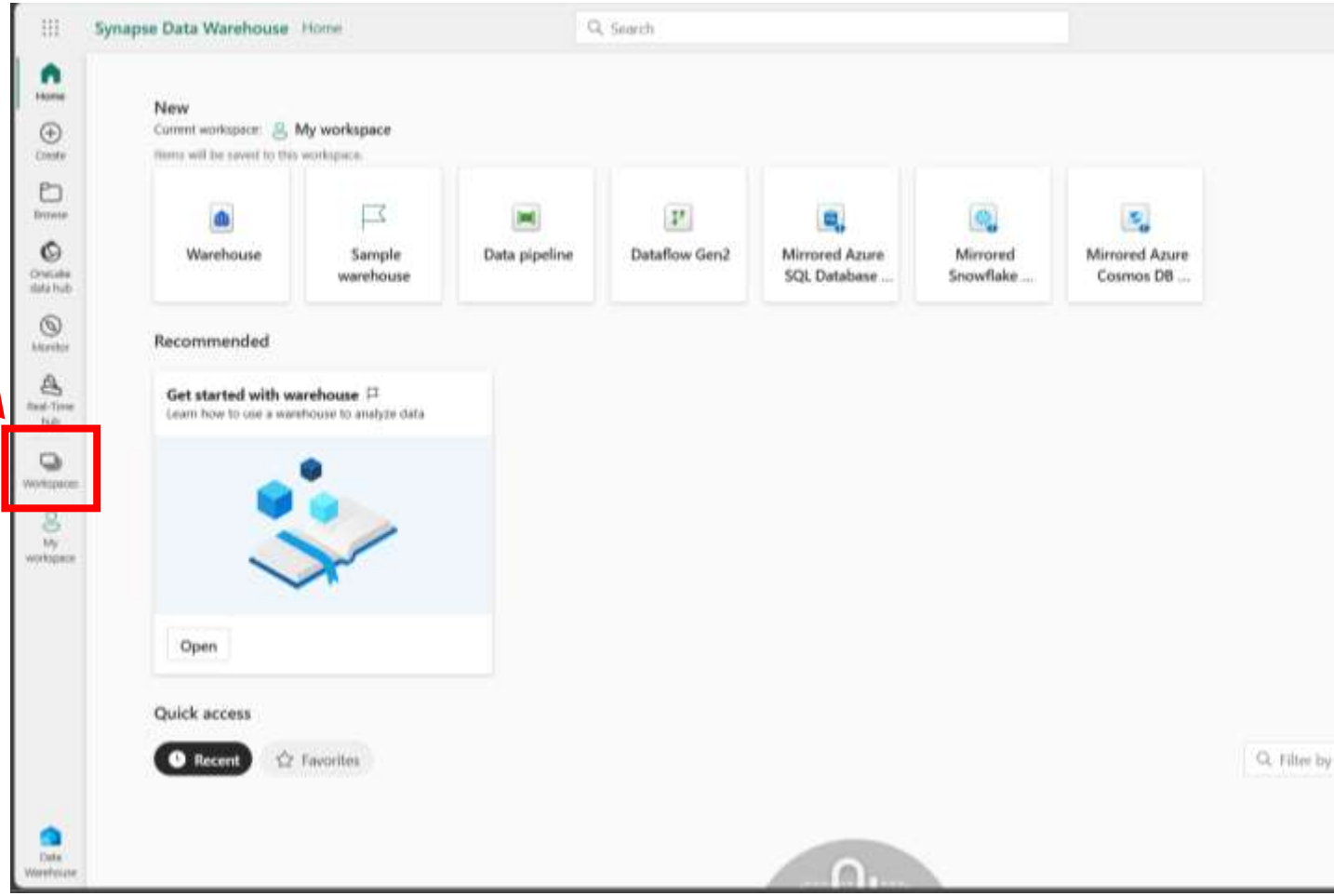
Ready to start working in Fabric

- Now you are ready to work with Fabric



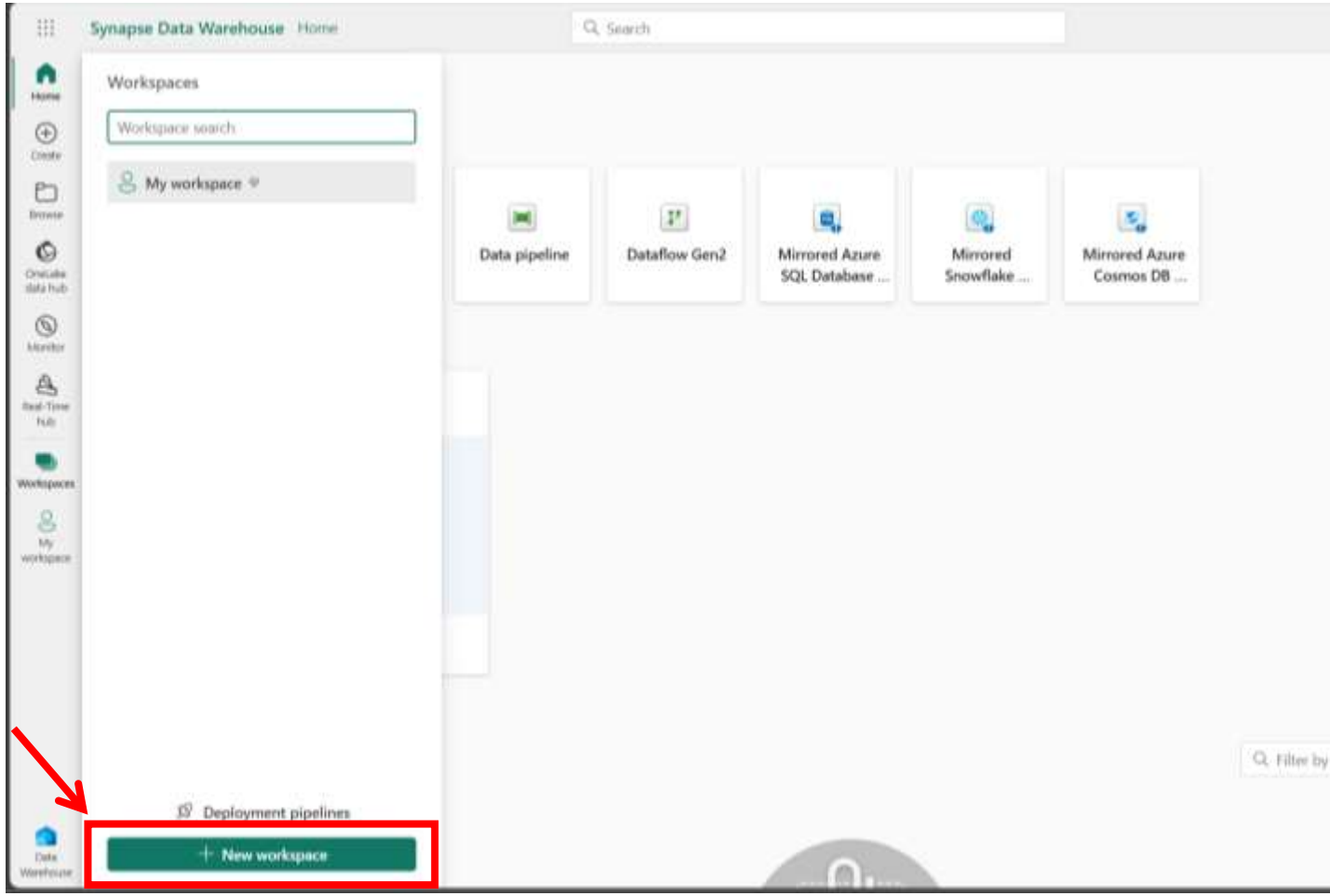
Create workspace, DO NOT use My Workspace

- Navigate to Synapse Data Warehouse
- Select *Workspaces*



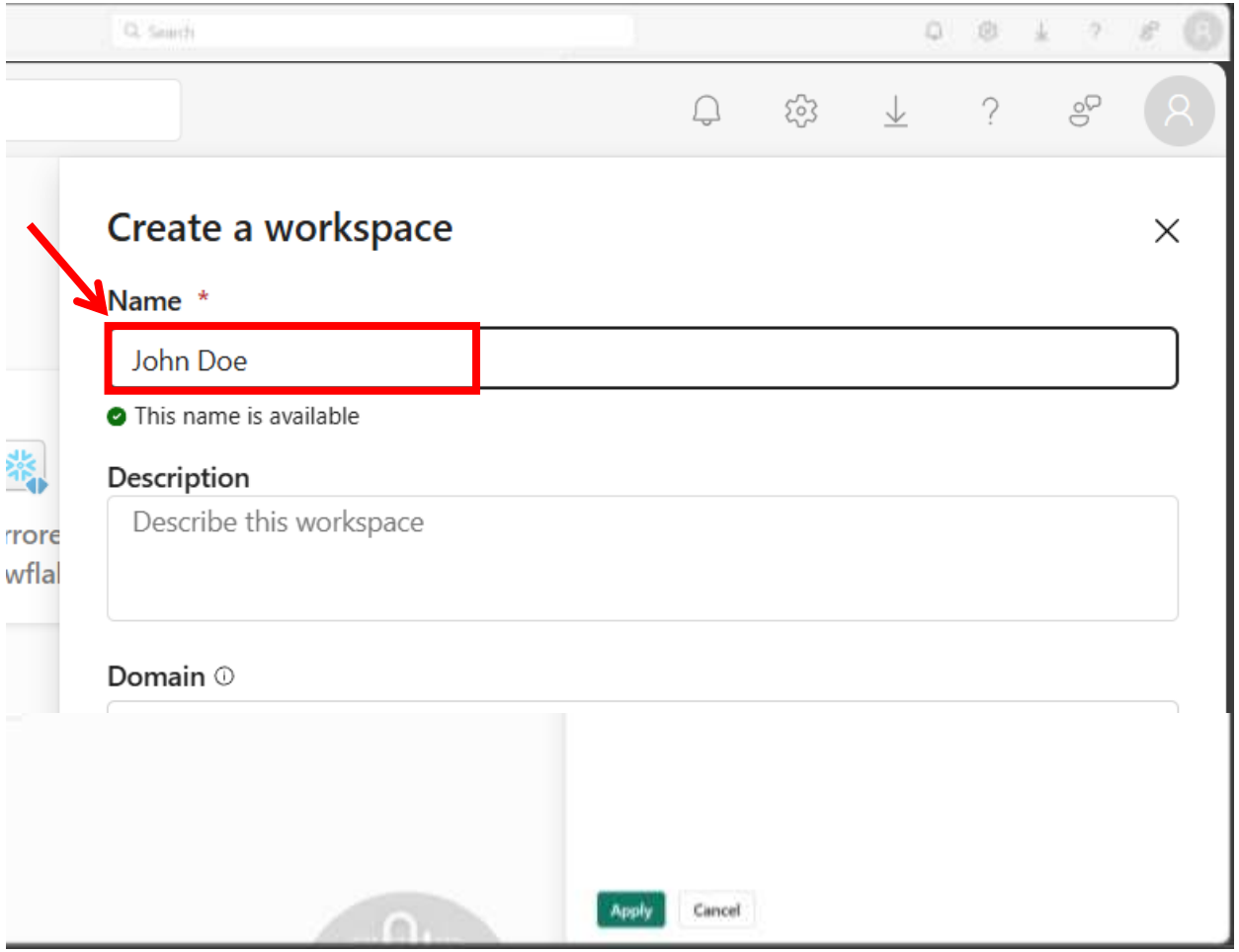
Create workspace

- Navigate to Synapse Data Warehouse
- Select *Workspaces*
- Create *New workspace*



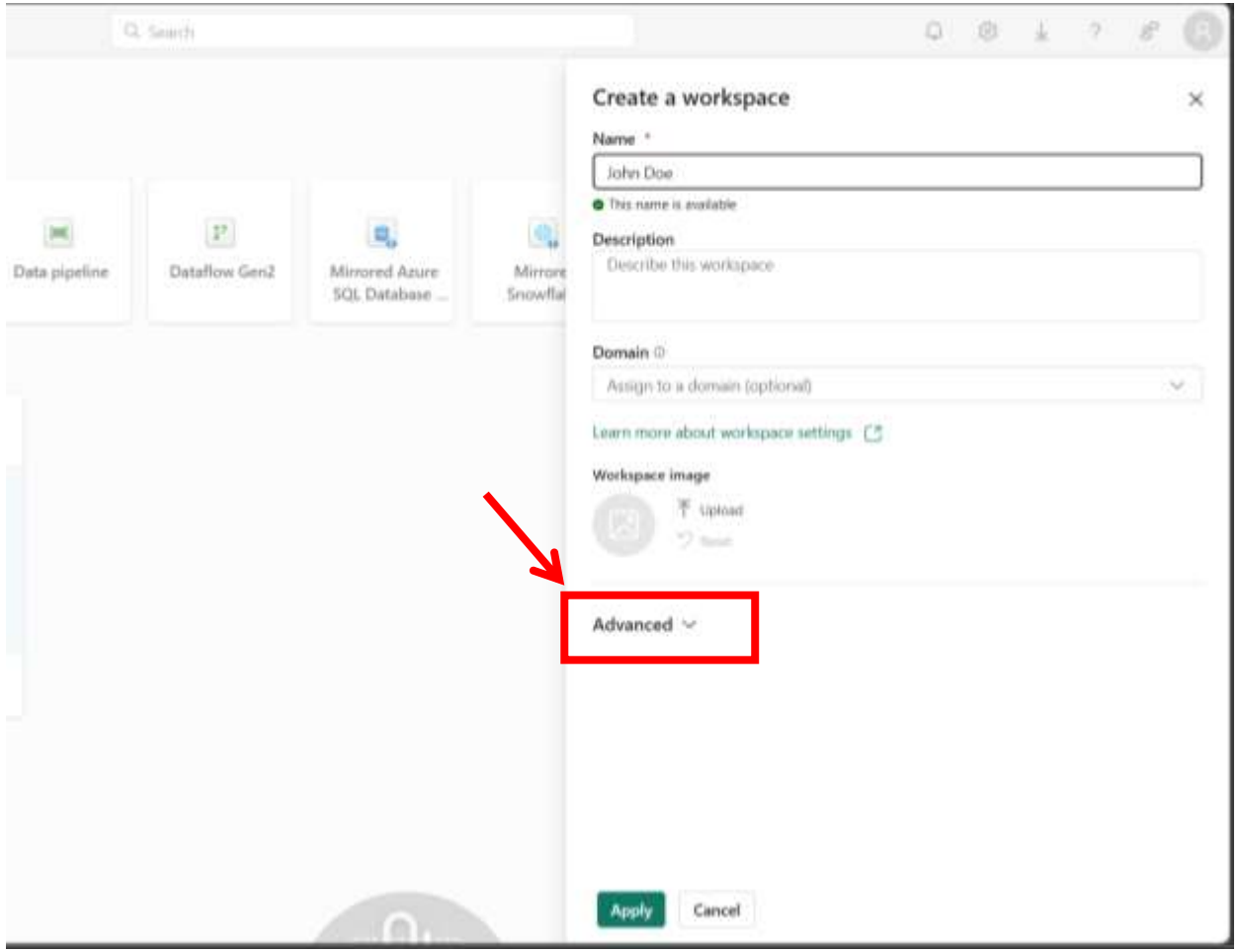
Create workspace

- Navigate to Synapse Data Warehouse
- Select *Workspaces*
- Create *New workspace*
- Name the workspace with your own name



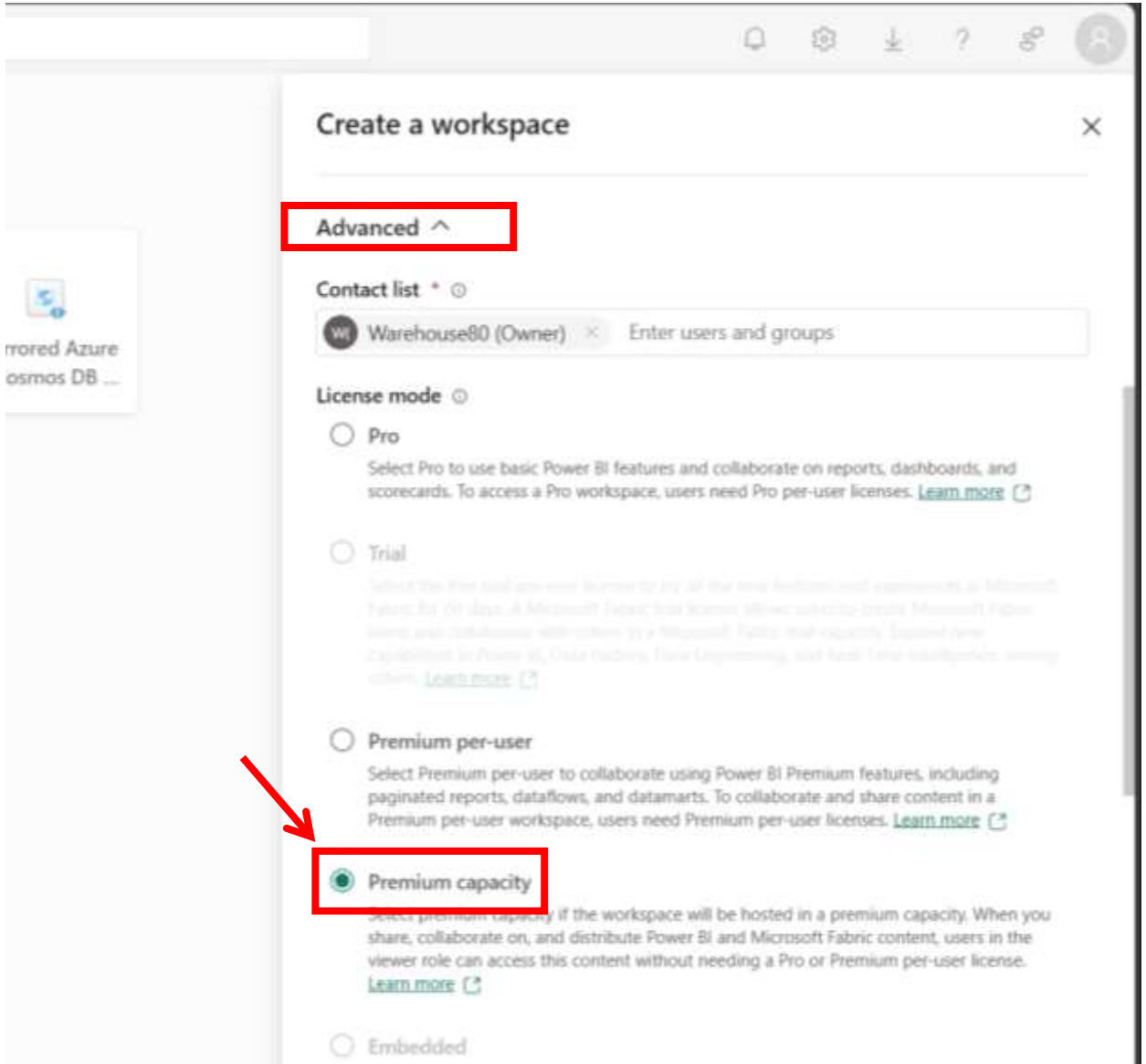
Create workspace – capacity

- Under Advanced in *Create a workspace*, you can verify that the workspace is assigned to Premium capacity called *Warehouse – North Europe*



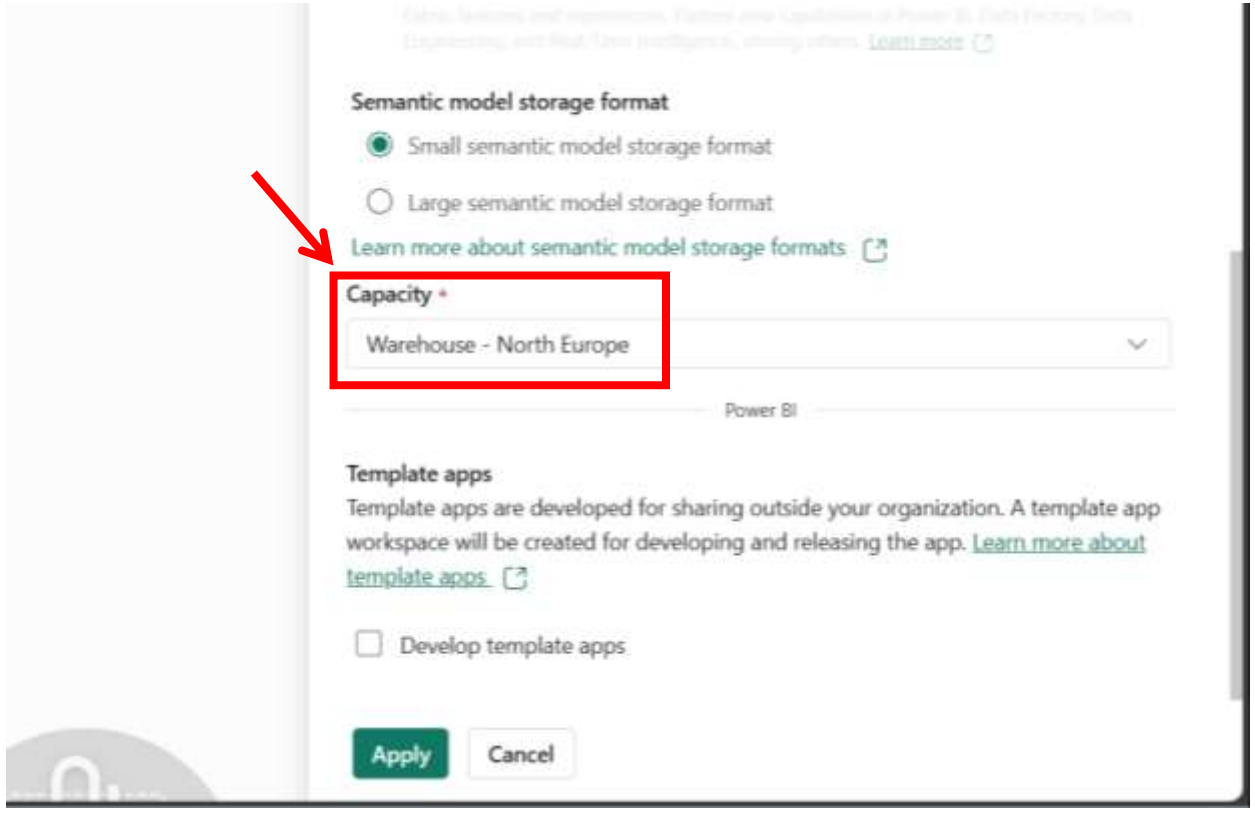
Create workspace – capacity

- Under Advanced in *Create a workspace*, you can verify that the workspace is assigned to Premium capacity called *Warehouse – North Europe*



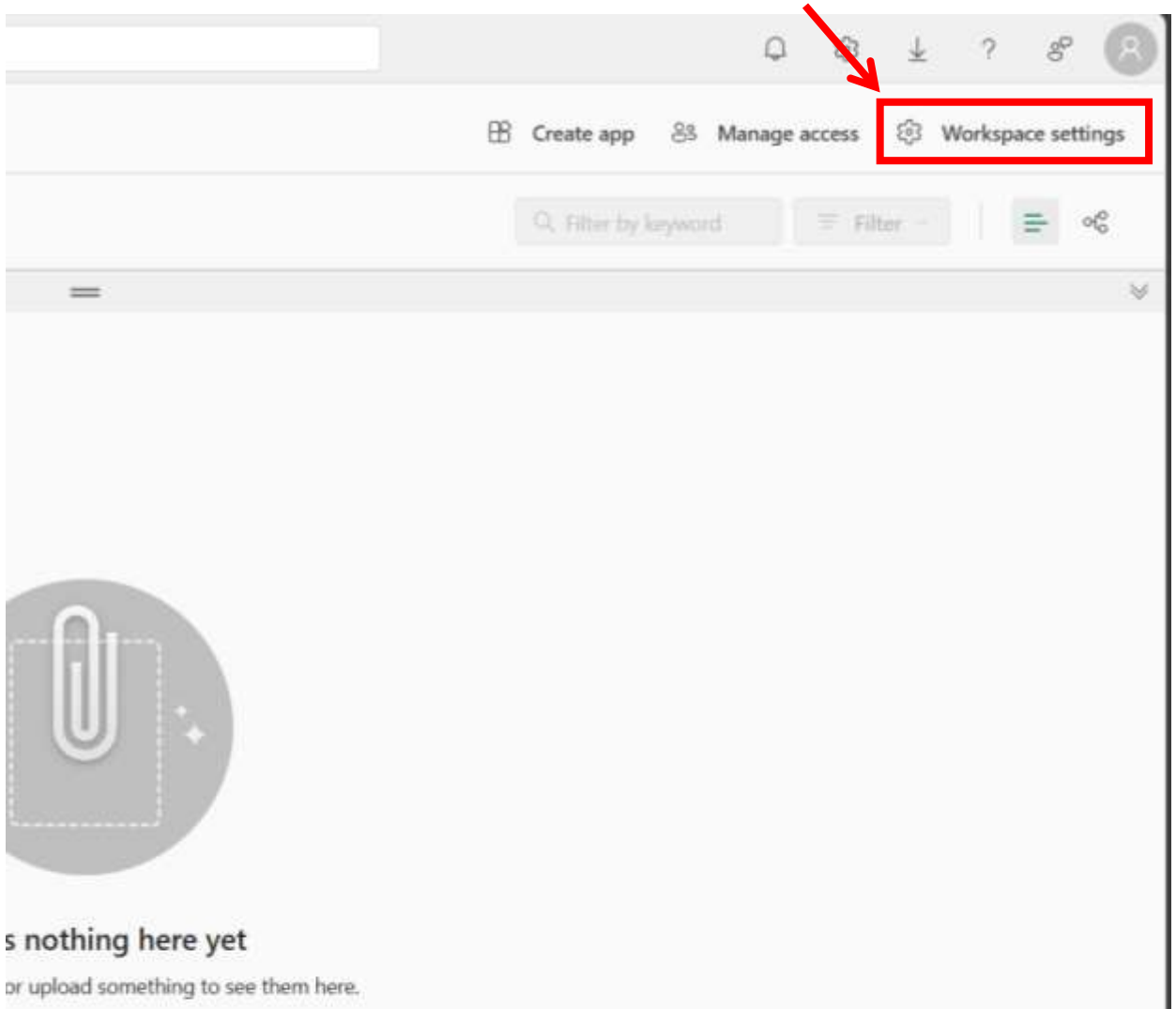
Create workspace – capacity

- Under Advanced in *Create a workspace*, you can verify that the workspace is assigned to Premium capacity called *Warehouse – North Europe*



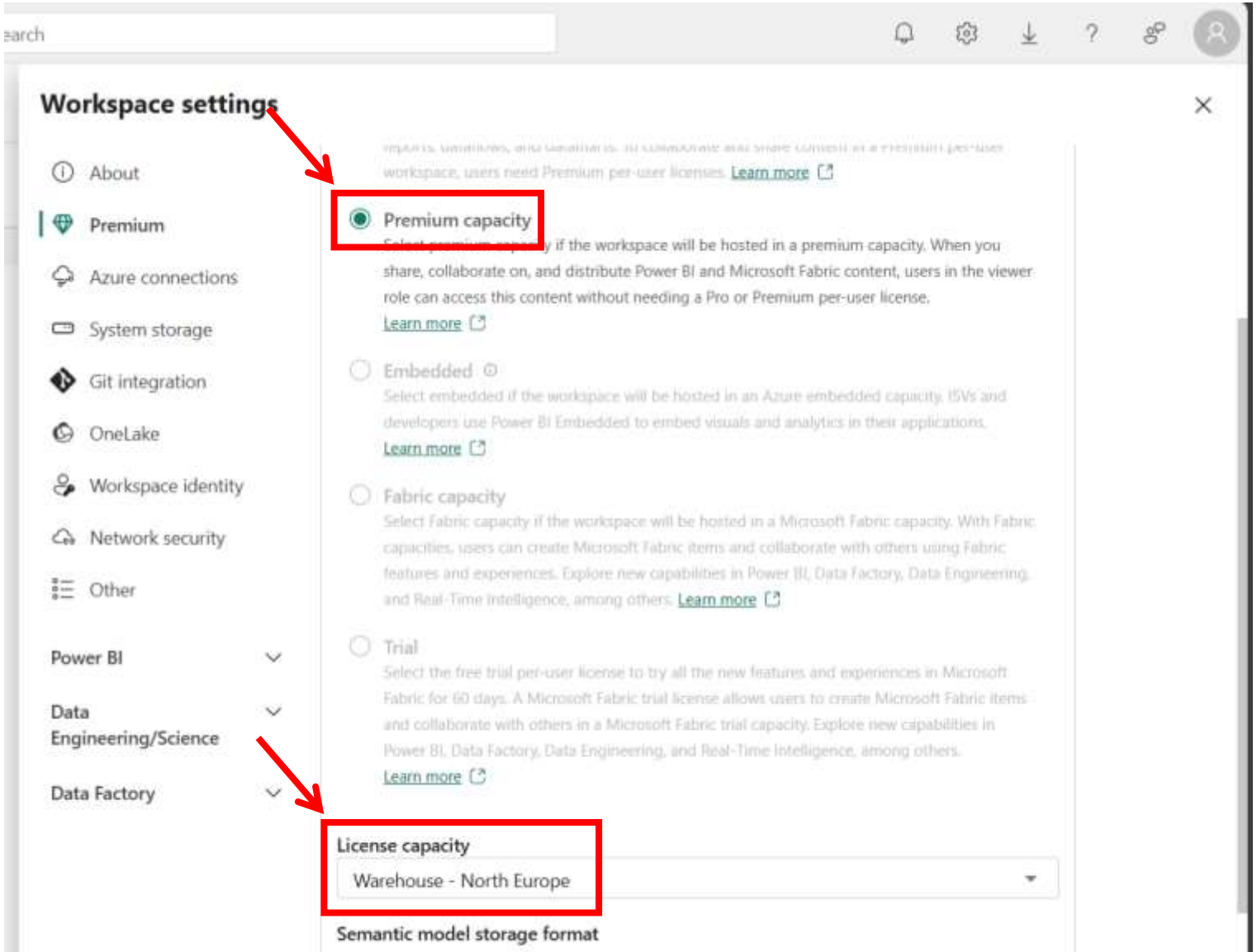
Create workspace – capacity

- If workspace was created before verifying Premium capacity, go to *Workspace settings* to verify capacity



Create workspace – capacity

- If workspace was created before verifying Premium capacity, go to *Workspace settings* to verify capacity



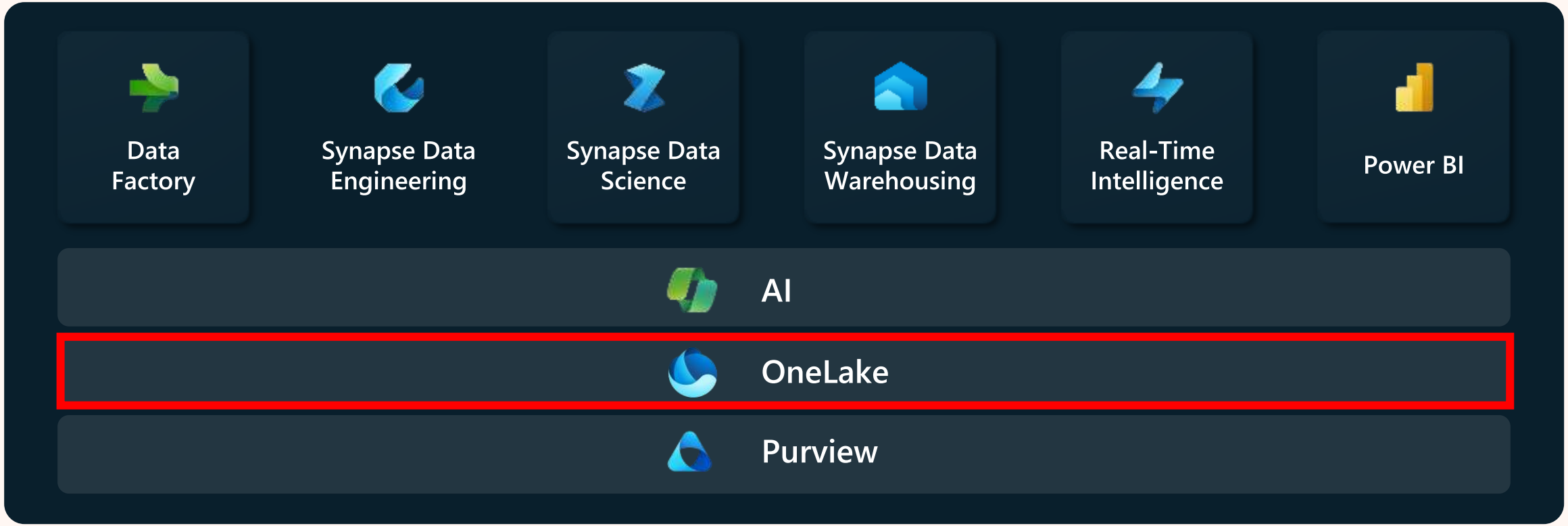
Now you are ready to create Fabric content





OneLake

Microsoft Fabric



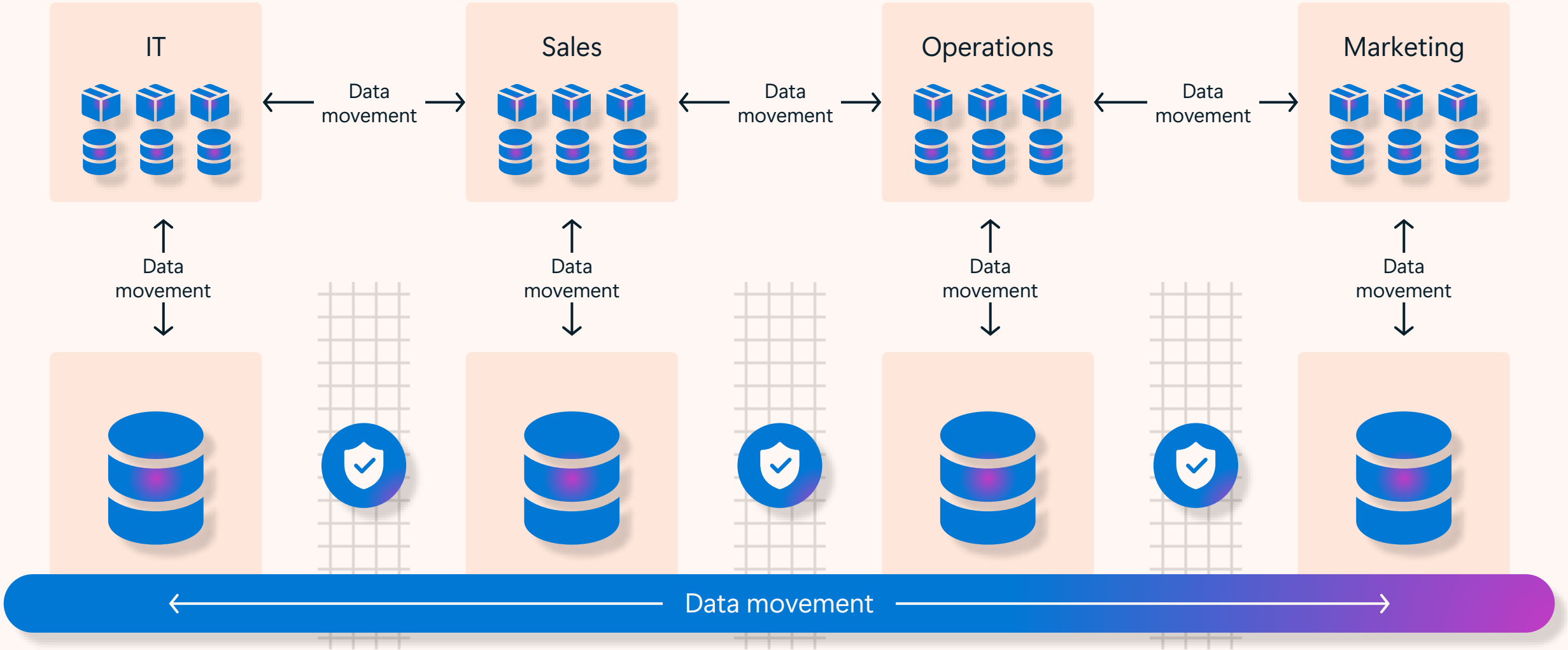
Reality: DIY data lakes using storage

You buy storage and build a complicated data lake solution on top of it.

- Getting the desired value from one data lake is hard. Business groups get frustrated with the pace of change when having to coordinate through a central team.
- Shifting to a data mesh pattern enables business groups to work independently with multiple business domain driven lakes. However, requires each team to manage their own lake.



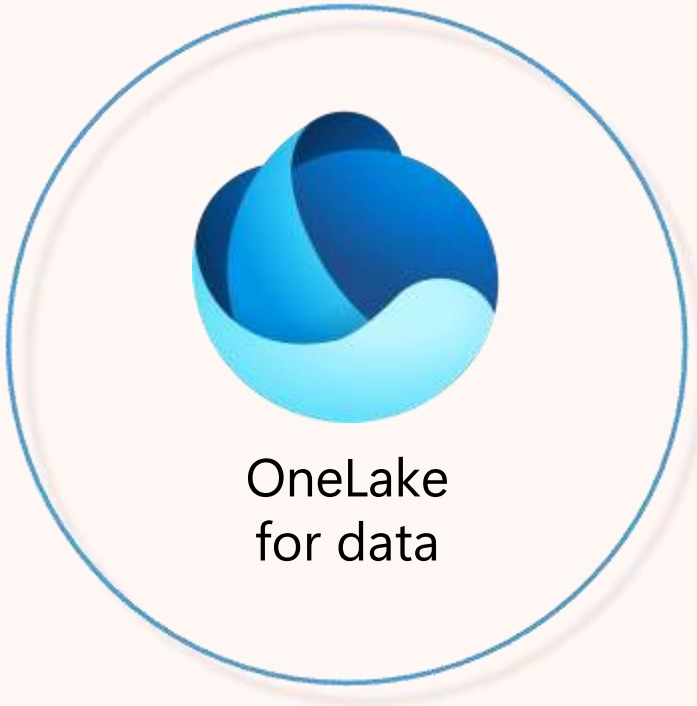
Today: Multiple siloed lakes with lots of duplication



OneLake for all data

"The OneDrive for data"





OneLake provides a data lake as a service without you needing to build it

OneLake for all data

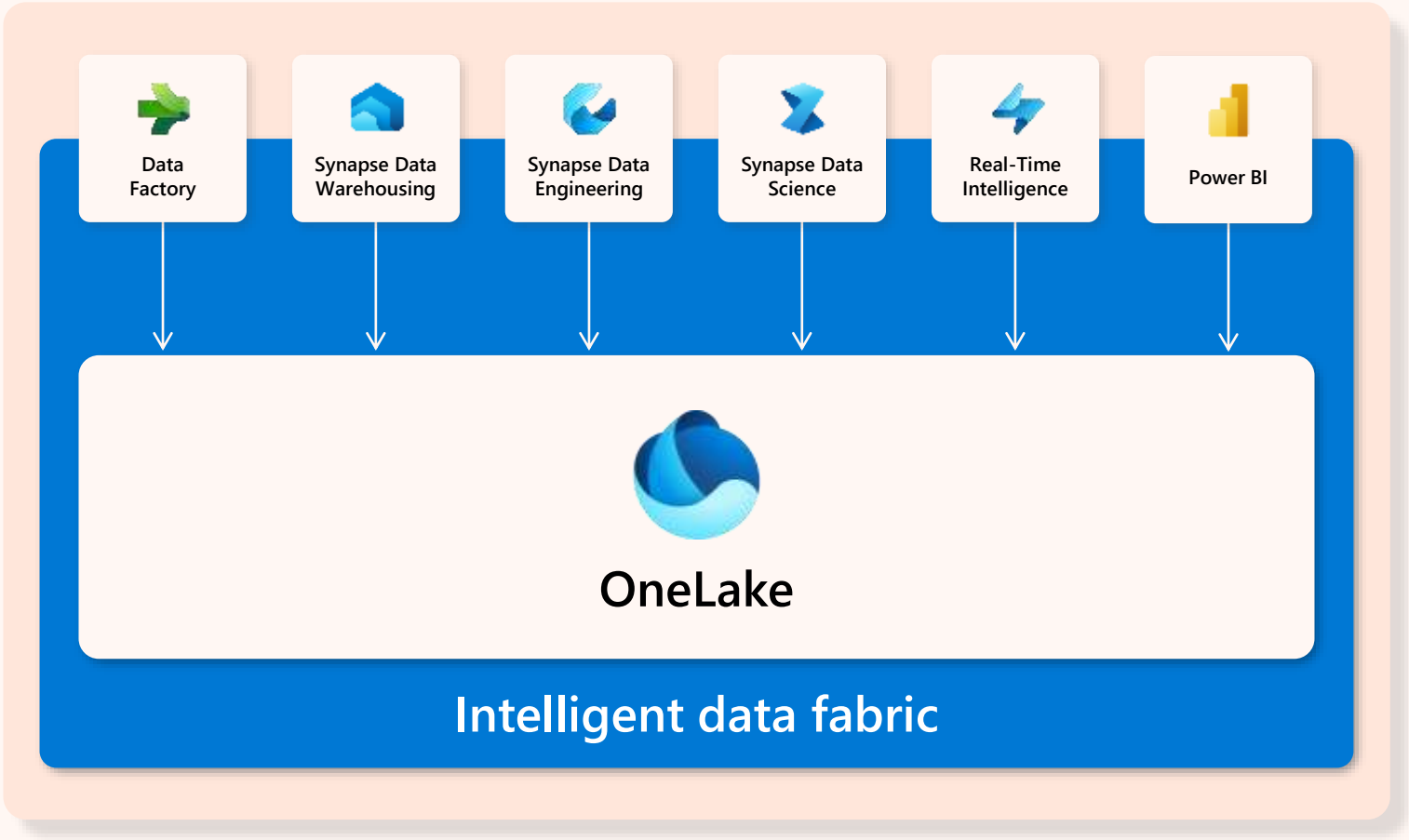
"The OneDrive for data"

OneLake

One Copy

One Security

OneLake Data Hub



OneLake for all data

"The OneDrive for data"

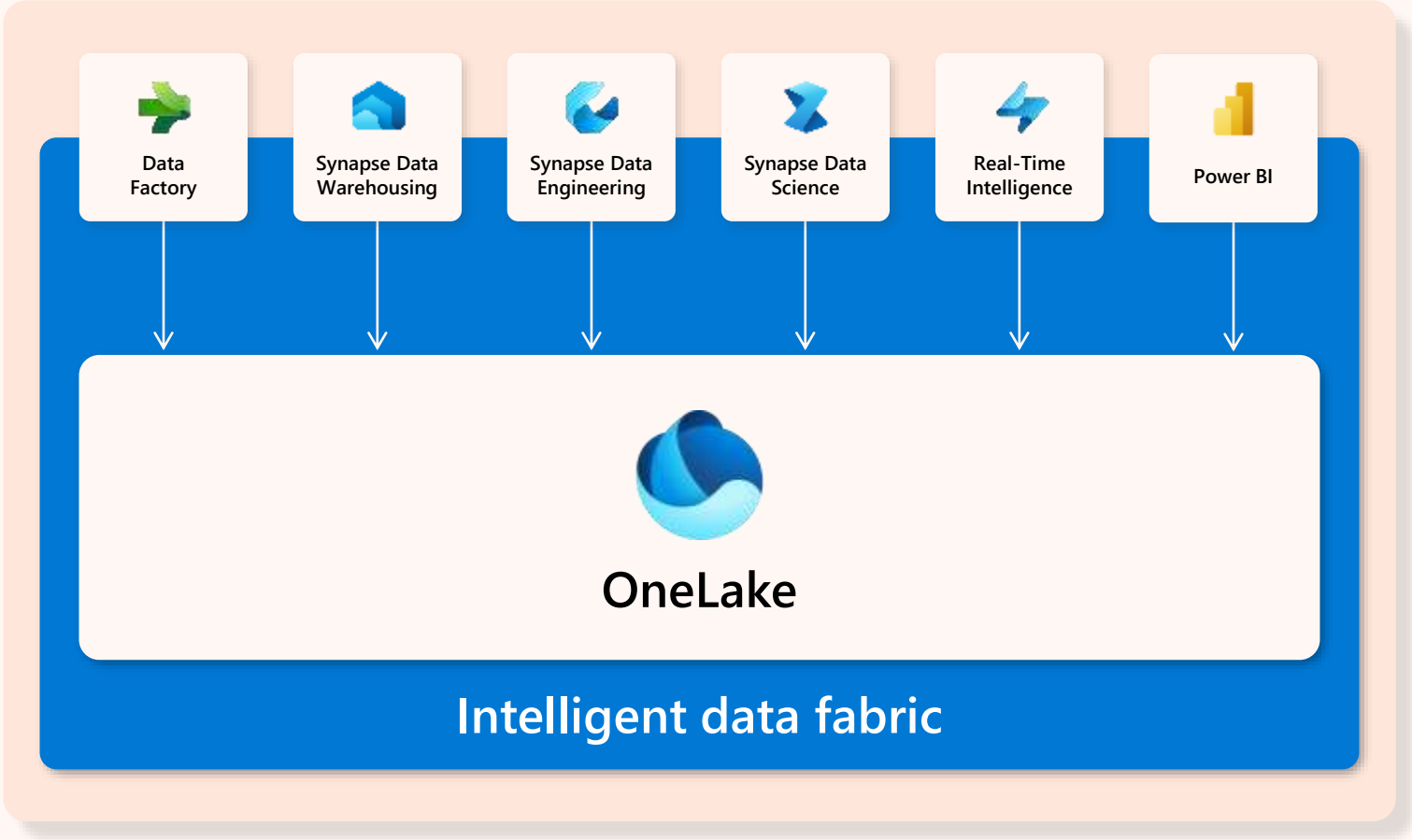
OneLake

- A single unified logical SaaS data lake for the whole organization (no silos)
- Organize data into domains
- Foundation for all Fabric data items
- Provides full and open access through industry standard APIs and formats to any application (no lock-in)

One Copy

One Security

OneLake Data Hub



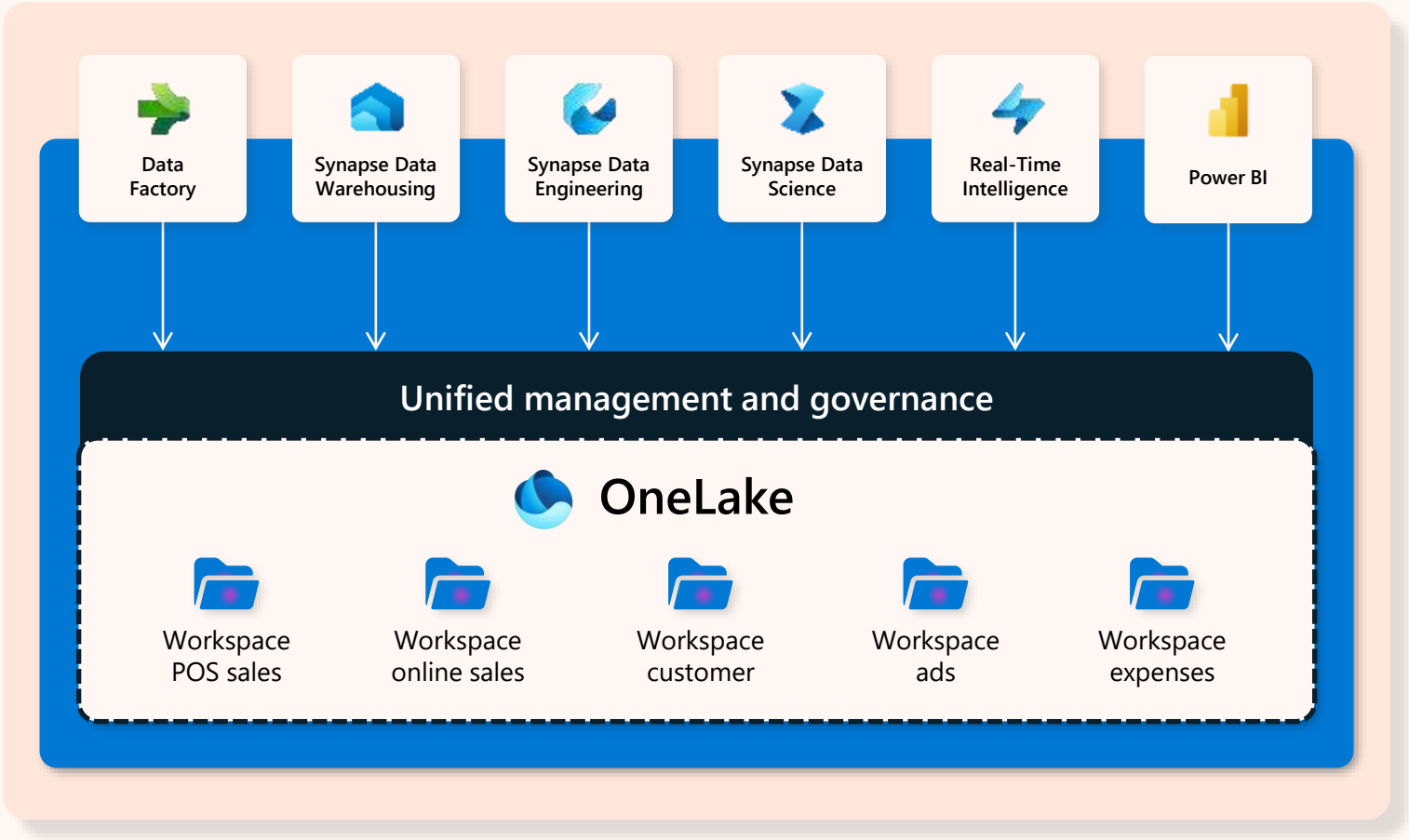
A single unified SaaS data lake

"No silos"

OneLake comes automatically provisioned with every Fabric tenant with no infrastructure to manage.

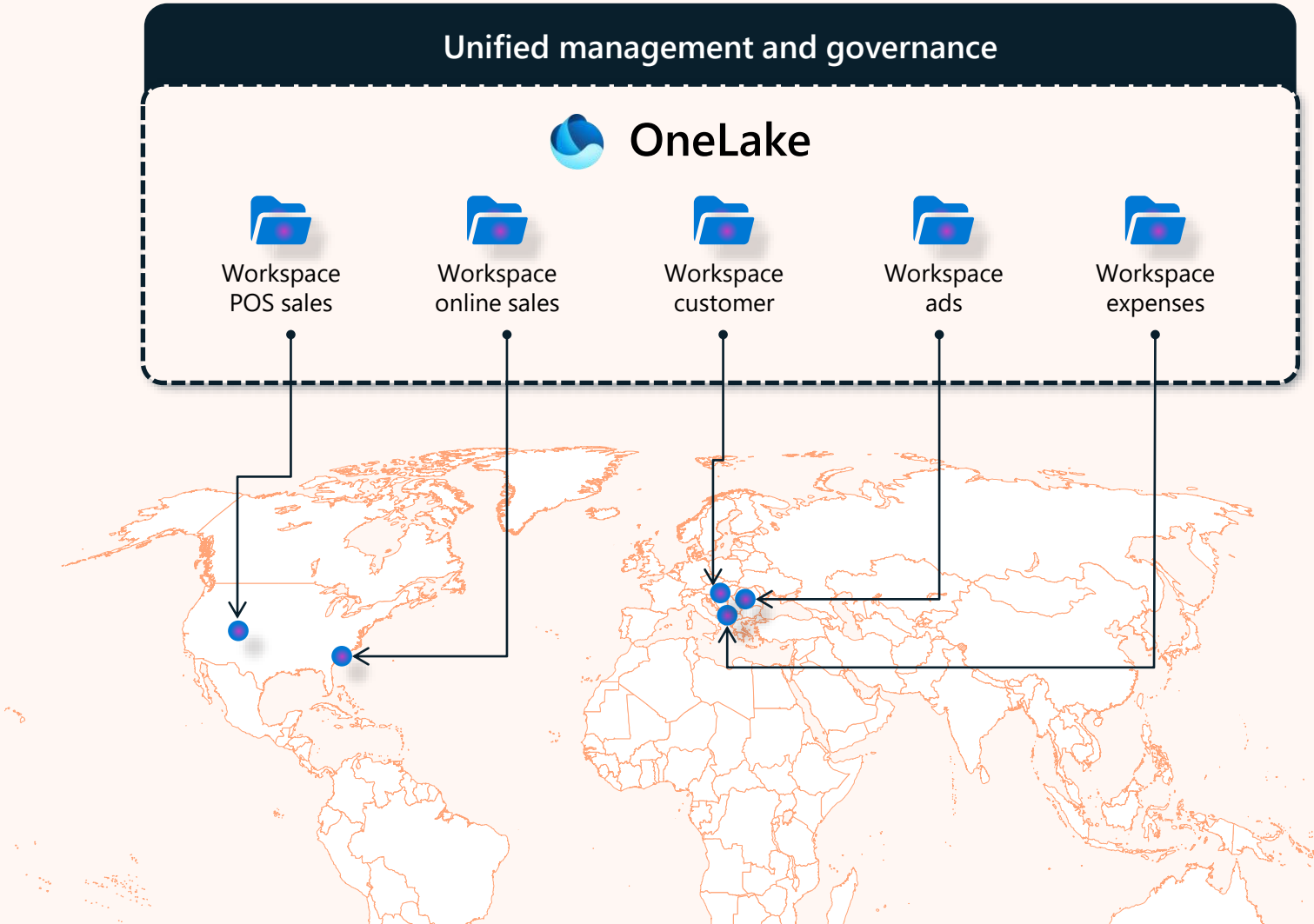
Any data in OneLake works with out-of-the-box governance such as data lineage, data protection, certification, catalog integration, etc. All data is ultimately under the control of a tenant admin.

OneLake enables distributed ownership. Different workspaces allow different parts of the organization to work independently while still contributing to the same data lake. Each workspace can have its own administrator, access control, region and capacity for billing.



OneLake which logically spans the world

Workspaces can reside in different regions around the world while still being part of the same data lake.



The foundation of all Fabric data items

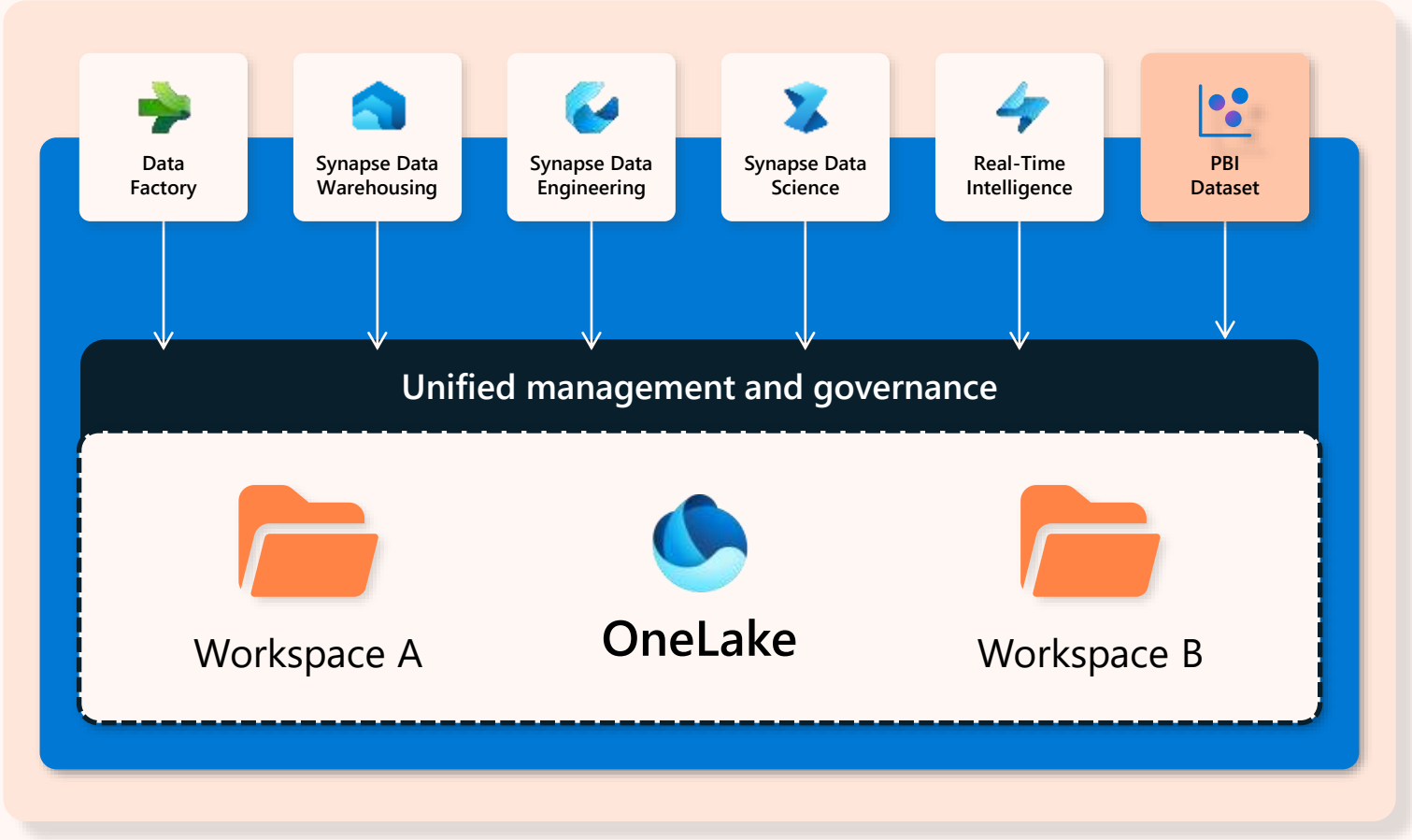
Fabric data items are prewired to store data in OneLake in open formats

All data is stored in a Fabric data item.

Today, users interact with Power BI via items like datasets.

In Fabric, different items give tailored experiences for each persona and scenario when accessing OneLake.

Like how Office stores Word, Excel, and PowerPoint documents in OneDrive.



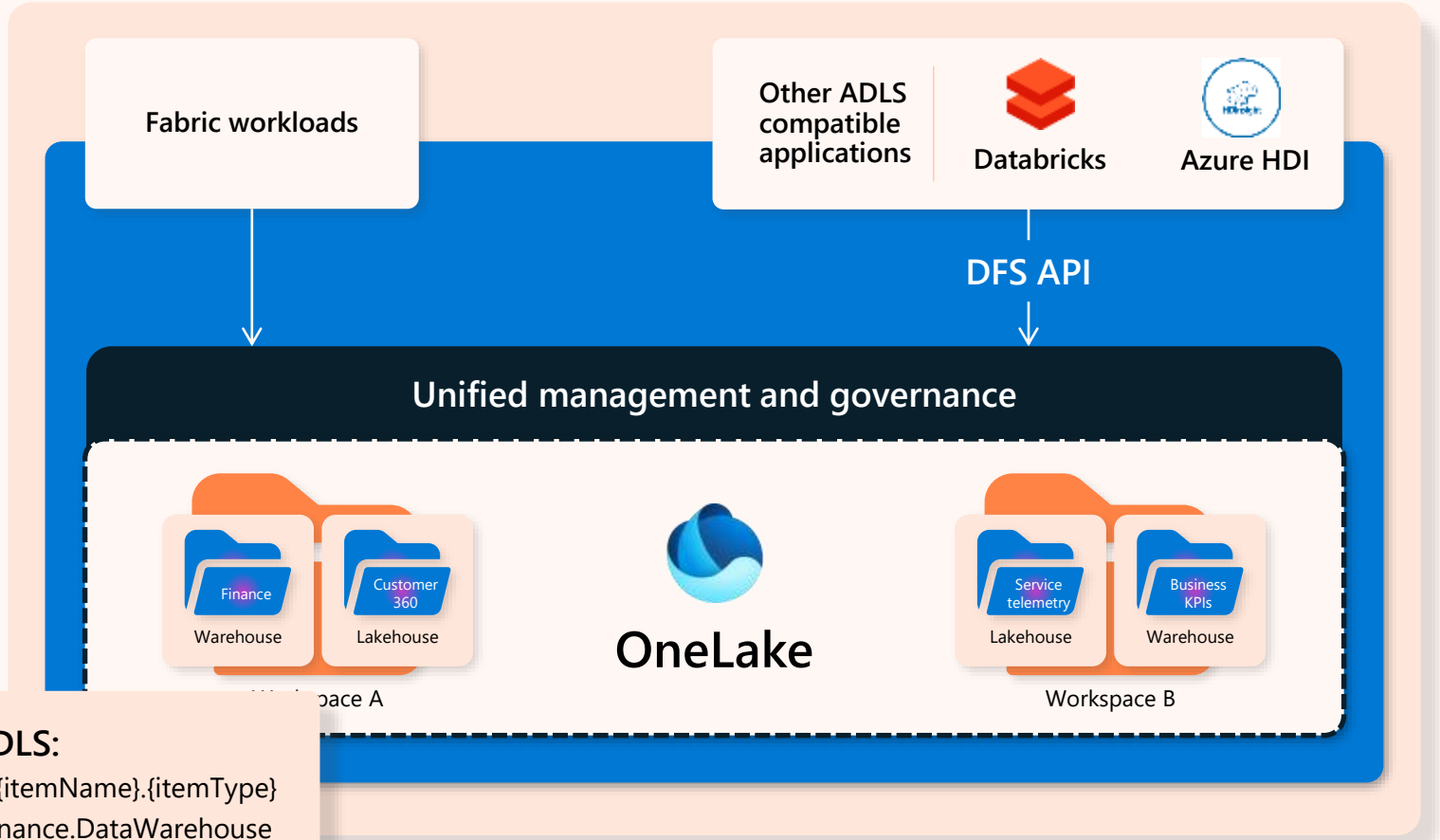
Open access to data in OneLake

No lock-in with industry standard APIs and open file formats

By supporting the ADLS Gen2 DFS APIs and SDKs, OneLake is compatible with existing ADLS applications

Tenants will appear as one big storage account with different workspaces appearing as different containers with data organized into folders

Underlying physical storage is virtualized away. OneLake ensures proper scale and performance.



Addressing Fabric workspaces and items as ADLS:

https://onelake.dfs.fabric.microsoft.com/{workspaceName}/{itemName}.{itemType}

Ex. https://onelake.dfs.fabric.microsoft.com/Workspace A/Finance.DataWarehouse

OneLake

Demo



Home
Create
Browse
Data hub
Workspaces
01_Contoso_KPIWorks...
Data Warehouse

New

New items saved to : 01_Contoso_KPIWorkspace

Show less

Warehouse

Data pipeline

Recommended

You frequently open this

01_Contoso_Workspace

You frequently open this

01_Contoso_KPIWorkspace

You frequently open this

#Contoso_Development

You frequently open this

My workspace

Getting started

Learn how to store and

Quick access

Recent | Favorites | Shared with me

Filter by keyword

Filter

Name	Type	Opened	Owner	Endorsement	Sensitivity	Workspace
01_Contoso_KPIWorkspace	Workspace	now	—	—	—	Workspaces
BusinessKPIs	Warehouse	6 minutes ago	Matthew Hicks	—	—	01_Contoso_KPIWor...
#Contoso_Campaigns	Workspace	19 minutes ago	—	—	—	Workspaces
LH	Lakehouse	19 minutes ago	Matthew Hicks	—	—	#Contoso_Campaigns
LH	Lakehouse	19 minutes ago	Matthew Hicks	—	—	#Contoso_Strategy

OneLake for all data

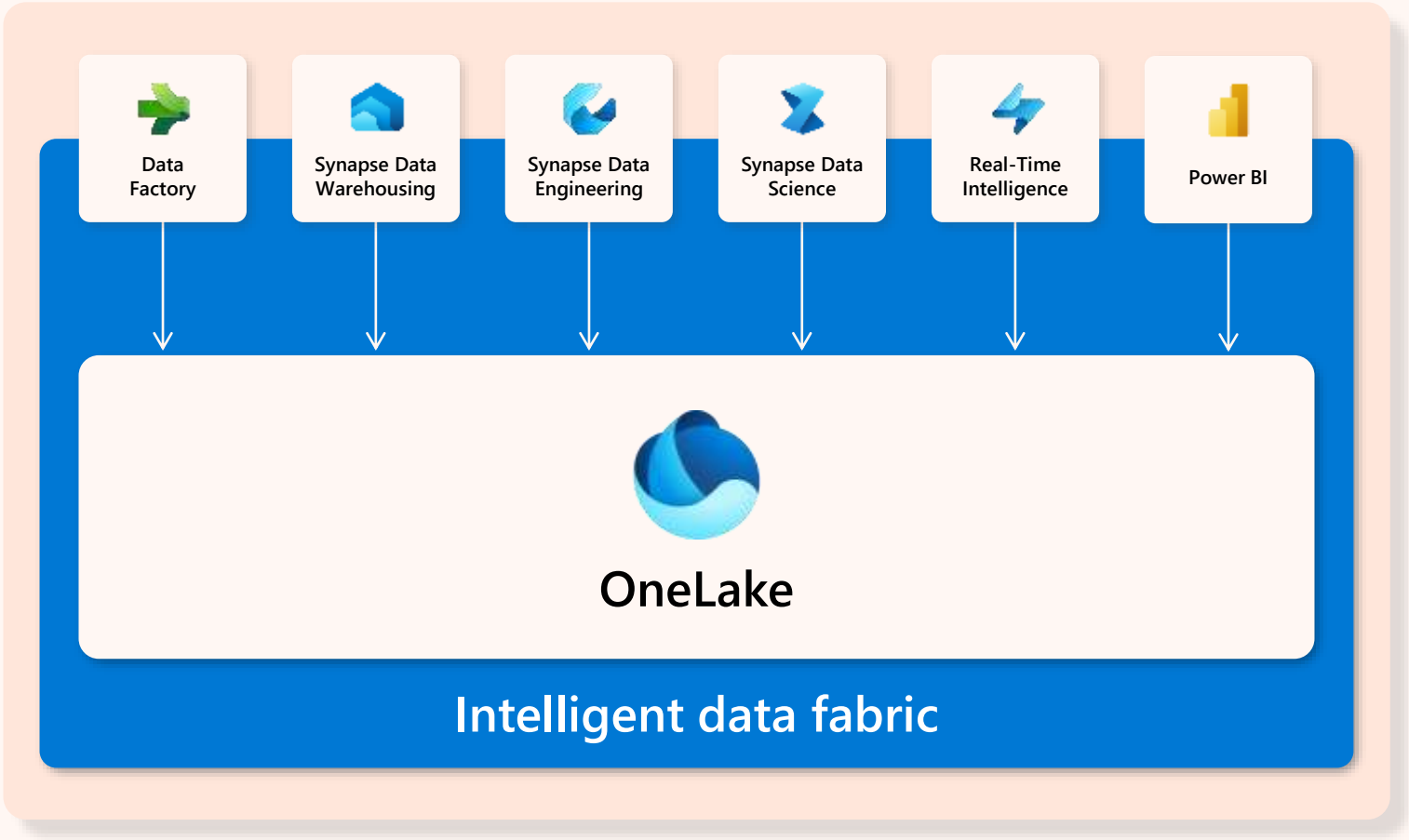
"The OneDrive for data"

OneLake

One Copy

- Virtualize data across domains and clouds into a single logical lake with shortcuts
- The One Copy of data for all the analytical engines of Fabric without moving or duplicating data

OneLake Data Hub



Shortcuts virtualize data across domains and clouds

No data movements or duplication

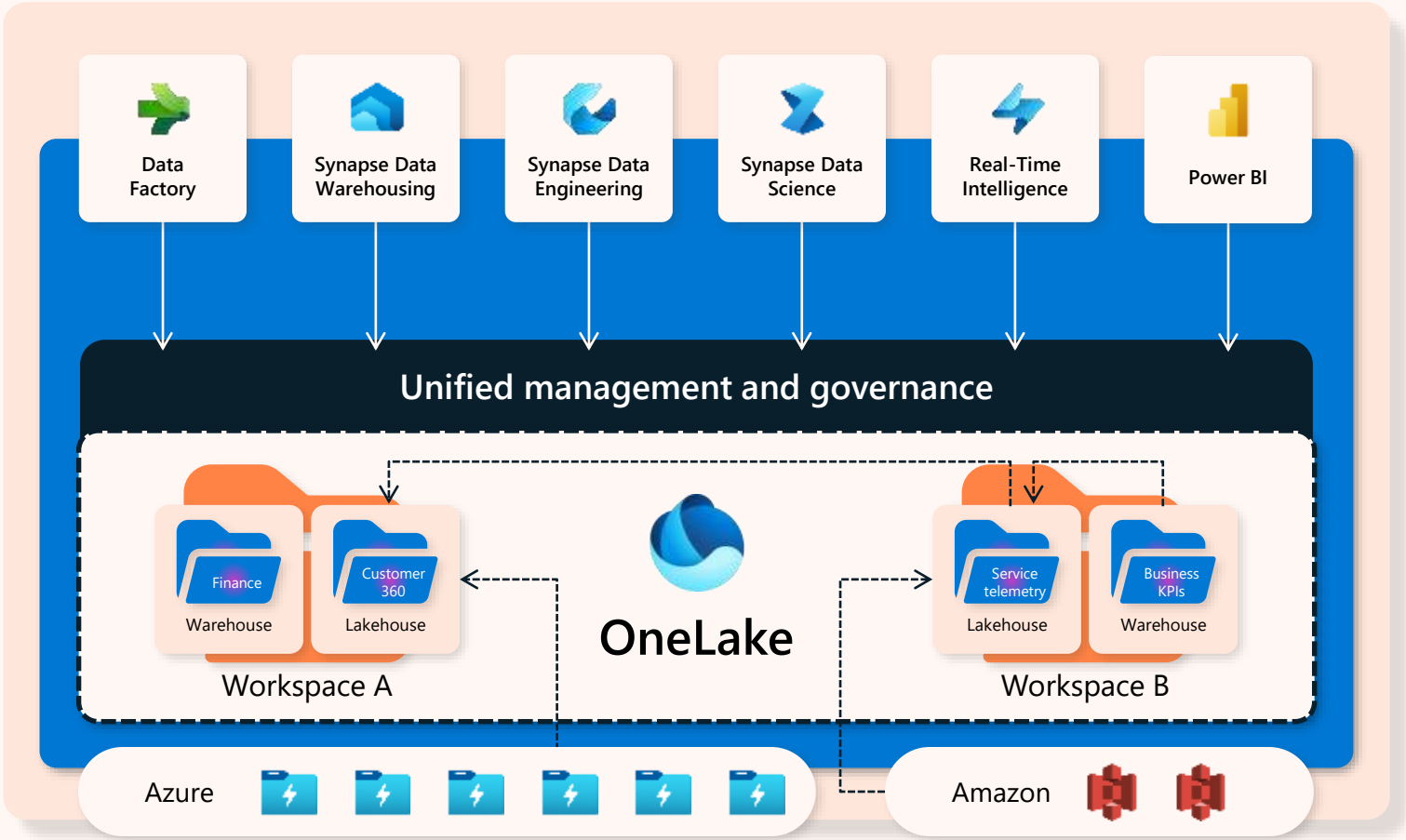
A shortcut is a symbolic link which points from one data location to another

Create a shortcut to make data from a warehouse part of your lakehouse

Create a shortcut within Fabric to consolidate data across items or workspaces without changing the ownership of the data. Data can be reused multiple times without data duplication.

Existing ADLS gen2 storage accounts and Amazon S3 buckets can be managed externally to Fabric and Microsoft while still being virtualized into OneLake with shortcuts

All data is mapped to a unified namespace and can be accessed using the same APIs including the ADLS Gen2 DFS APIs

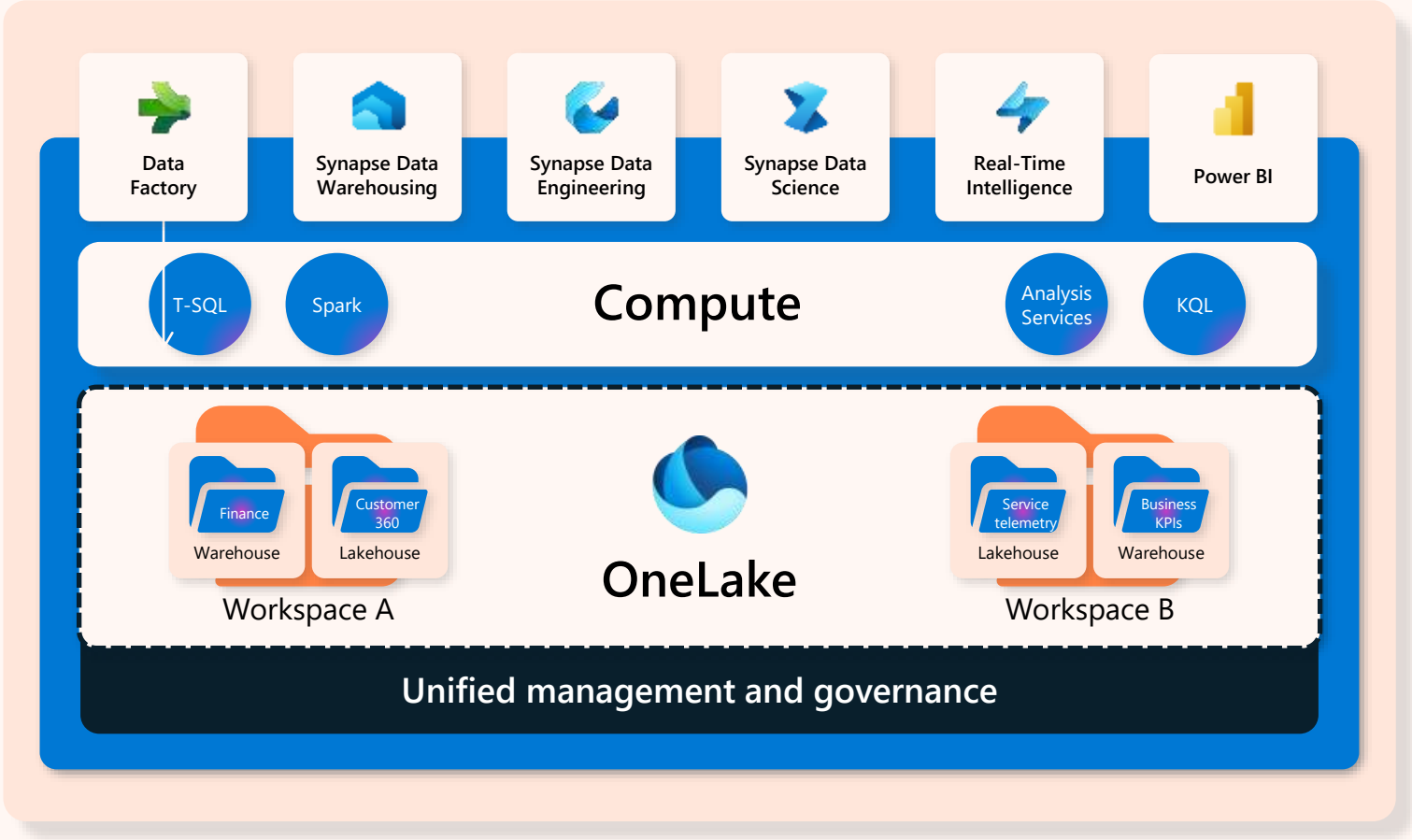


One Copy for all computes

Real separation of compute and storage

Compute powers the applications and experiences in Fabric. The compute is separate from the storage.

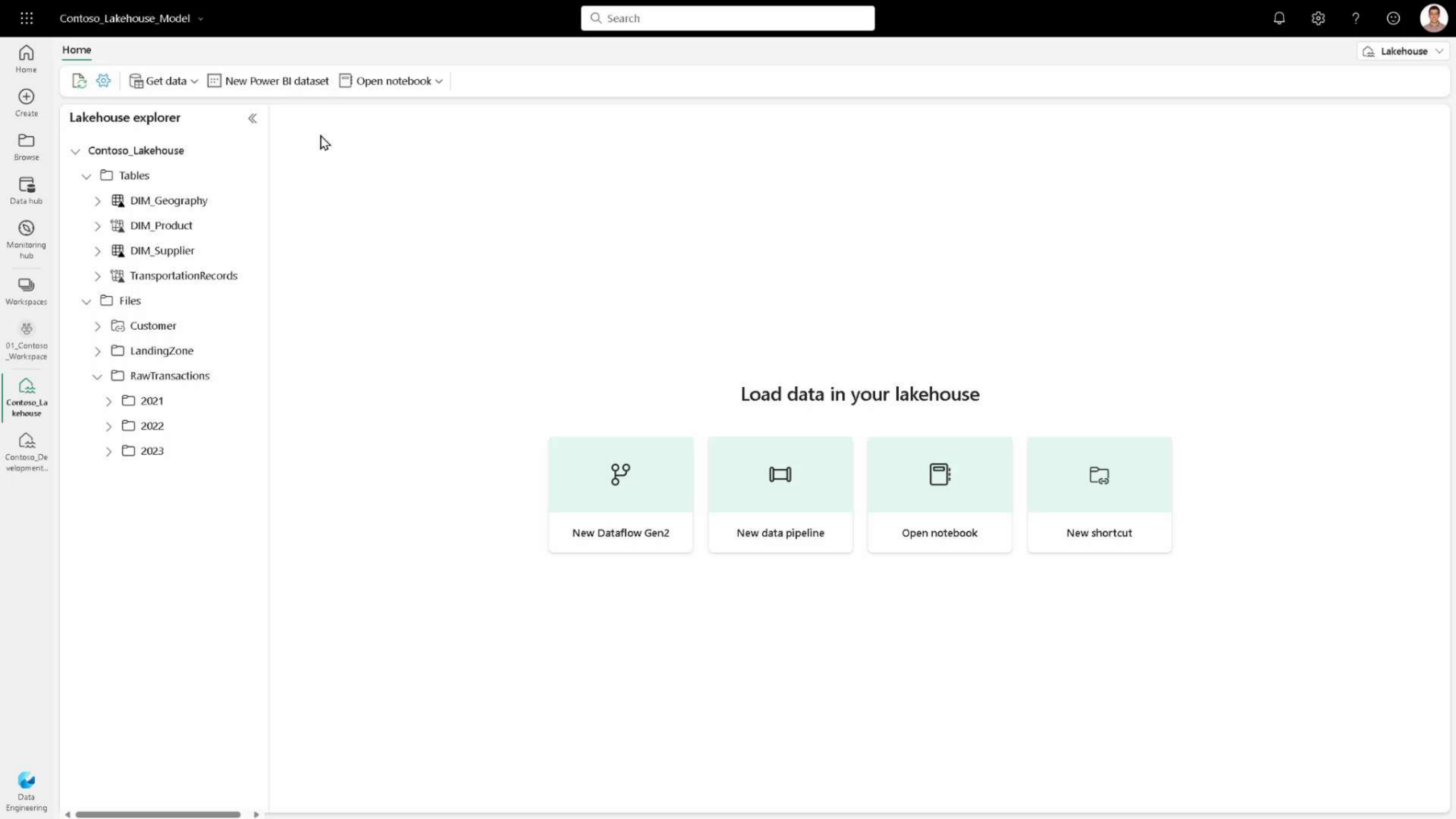
Multiple compute engines are available, and all engines can access the same data without needing to import or export it. You are able to choose the right engine for the right job.



OneCopy

Demo




Contoso_Lakehouse_Model
Search
Home
Lakehouse
Get data
New Power BI dataset
Open notebook
Lakehouse explorer
Contoso_Lakehouse
Tables
DIM_Geography
DIM_Product
DIM_Supplier
TransportationRecords
Files
Customer
LandingZone
RawTransactions
2021
2022
2023
Load data in your lakehouse
New Dataflow Gen2
New data pipeline
Open notebook
New shortcut
Home
Create
Browse
Data hub
Monitoring hub
Workspaces
01_Contoso_Workspace
Contoso_Lakehouse
Contoso_Development...
Data Engineering

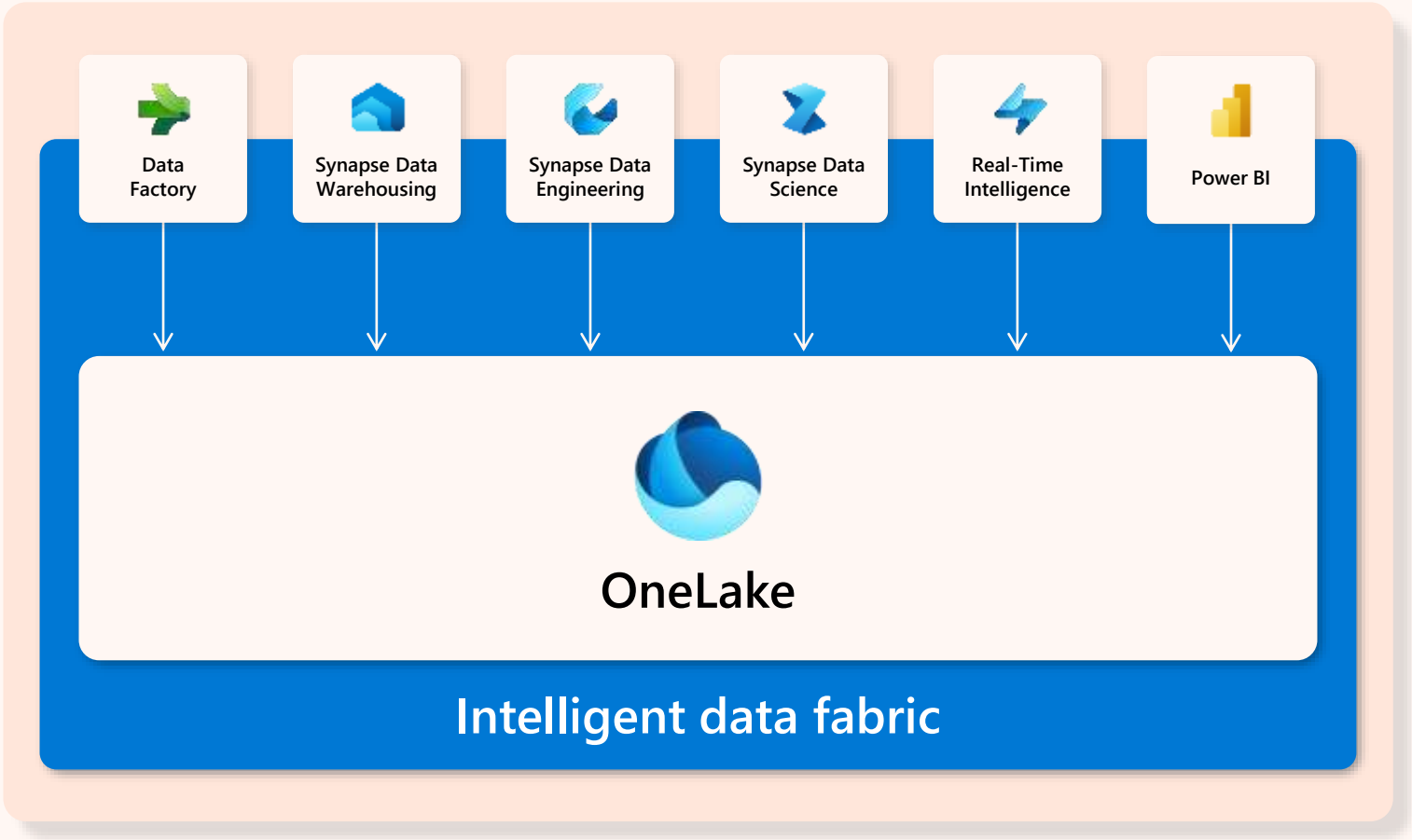
OneLake for all data

"The OneDrive for data"

OneLake

One Copy

OneLake Data Hub



OneLake Data Hub

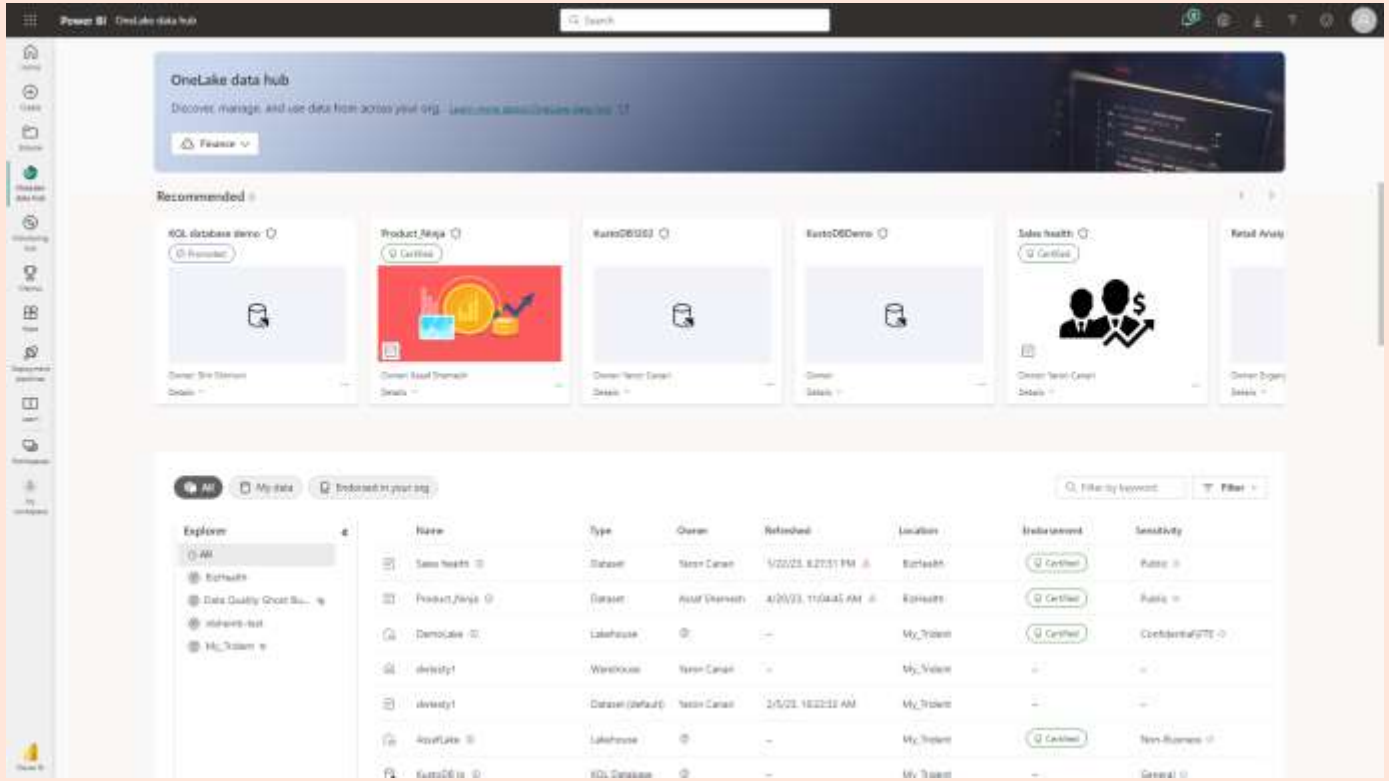
Discover, manage and use data in one place

Central location within Fabric to discover, manage, and reuse data

Data can be easily discovered by its domain (e.g. Finance) so users can see what matters for them

Efficient data discovery using search, filter and sort

Explorer capability to easily browse and find data by its folder (workspace) hierarchy



Home
Create
Browse
OneLake data hub
Monitoring hub
Metrics
Apps
Deployment pipelines
Learn
Workspaces
My workspace

OneLake data hub

Discover, manage, and use data from across your org. Learn more about OneLake data hub

All domains

Recommended

DemoLake	LHDemo1	DemoLake2	DemoLakehouse2	DemoLake3	KustoDB lo
Certified	Certified	Certified	Certified		
Owner:	Owner: Yaron Canari	Owner:	Owner: Omer Sade	Owner: Yaron Canari	Owner:
Details	Details	Details	Details	Details	Details

All | My data | Endorsed in your org

Filter by keyword | Filter

Explorer

- All
- My Workspace
- !_impact_analysis_1
- akj deployment 2
- Alexandr Gavrishev's w...
- Assafshemeshfsdfsdfsg...
- bbash
- BizHealth
- Boba Inventory
- CDS for Apps raw data

Name	Type	Owner	Refreshed	Location	Endorsement	Sensitivity
DemoLake	Lakehouse		-	My_Fabric	Certified	Confidential\FTE
Sales health	Dataset	Yaron Canari	1/22/23, 6:27:31 PM	BizHealth	Certified	Public
DemoLake3	Lakehouse	Yaron Canari	-	My_Fabric	-	General
Account Revenue	Dataset	Yaron Canari	1/22/20, 9:32:08 AM	Yaron demo WS2	Certified	-
LHDemo1	Lakehouse	Yaron Canari	-	Yaron Demo WS1	Certified	Confidential\FTE
DemoLake2	Lakehouse		-	My_Fabric	Certified	Non-Business
DemoLakehouse2	Lakehouse	Omer Sade	-	Compact view	Certified	Highly Confidential...



Time to build together!

Lab 1

Create LH and
shortcuts to the source

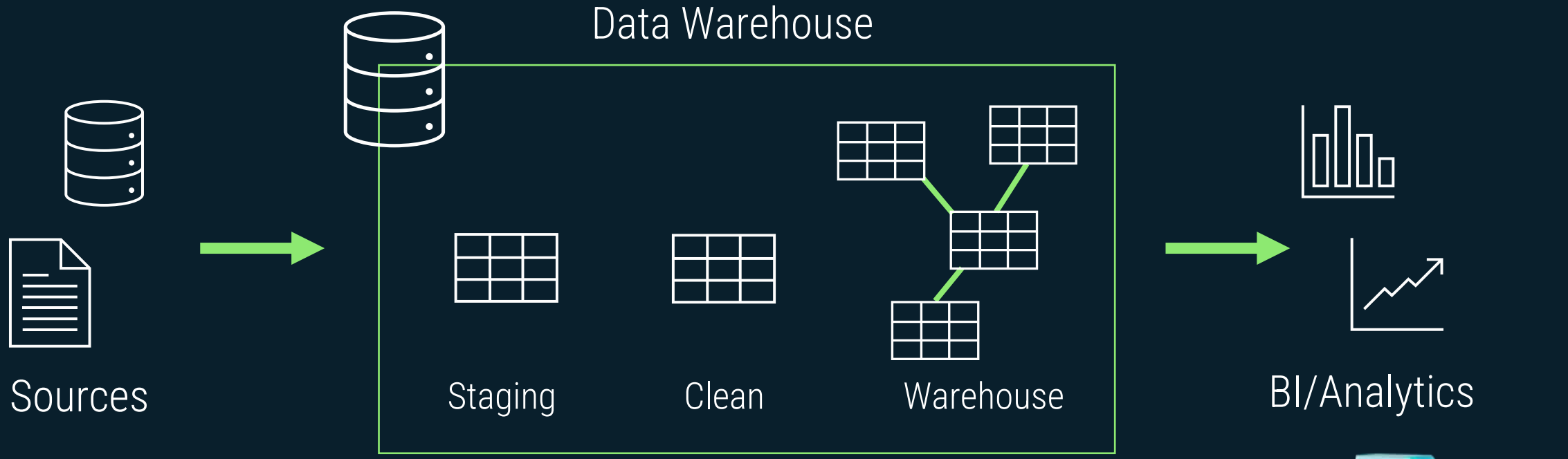


Medallion Architecture

Medallion Architecture



Data Warehousing



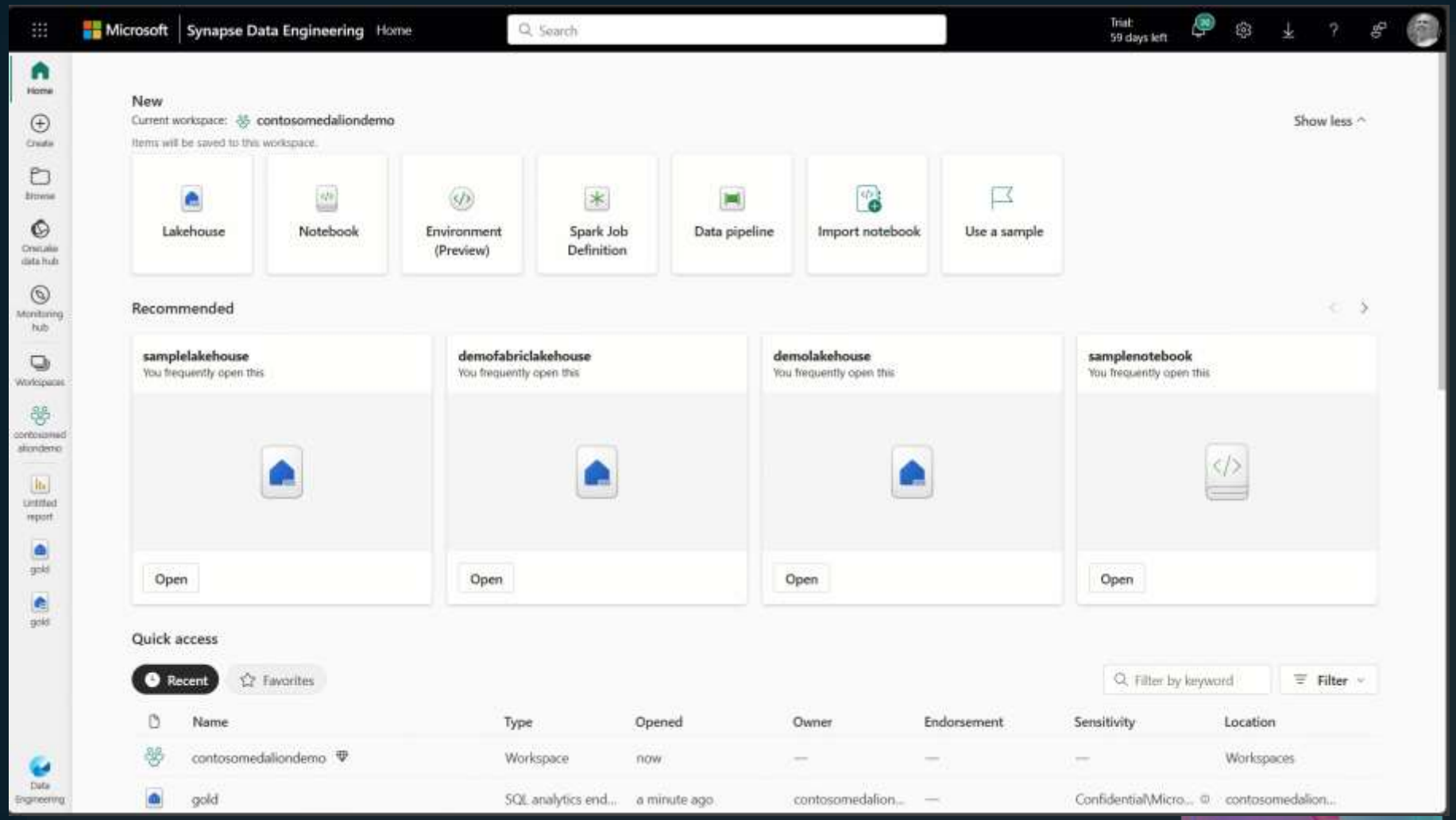


The Data Lakehouse



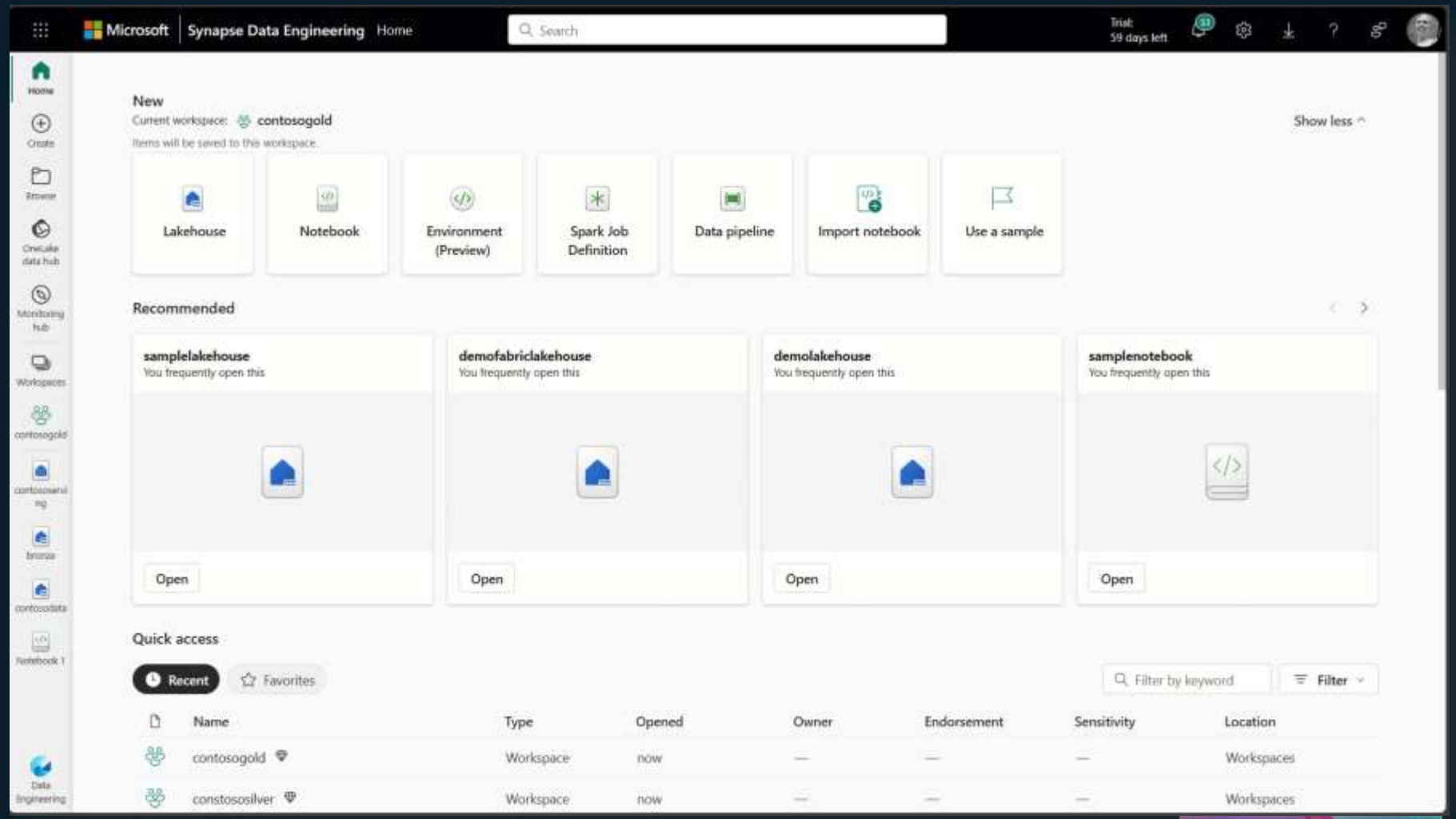
Multi-database architecture in Fabric

- Easy navigation
- Data joins
- Sharing



Multi-workspace architecture in Fabric

- Layer segregation
- Access control
- Capacity management



Bronze



Bronze

- First ingestion from source
- Raw state
- Incremental Growth

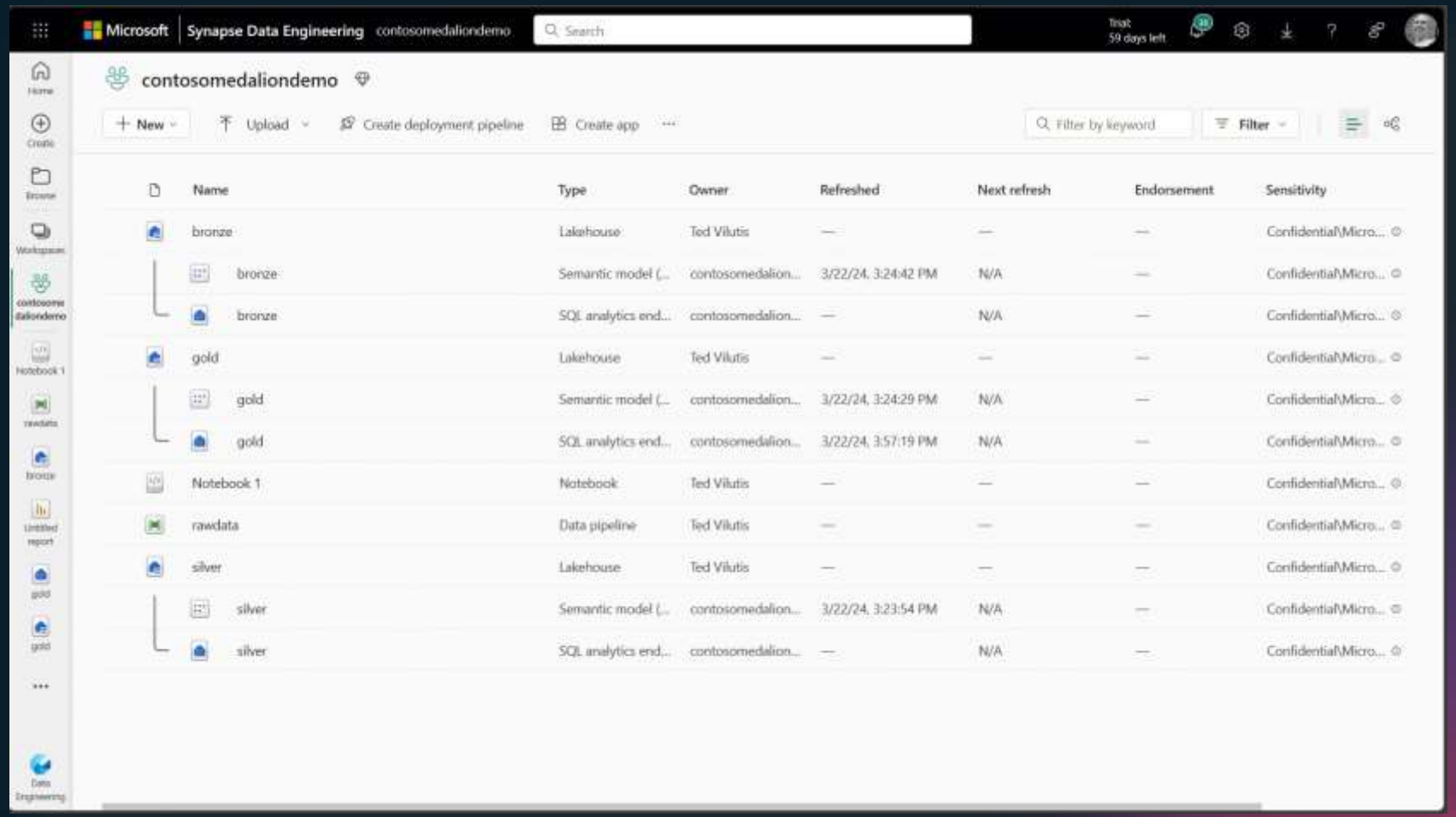
Who

Data Engineer



Getting data to Bronze layer in Fabric

- Copy tool in Pipelines
- Shortcuts
- Event stream
- Notebook
- Dataflows



Silver



Silver

- Validate & Deduplicate
- Trusted Source
- Cleaned & Usable

Who

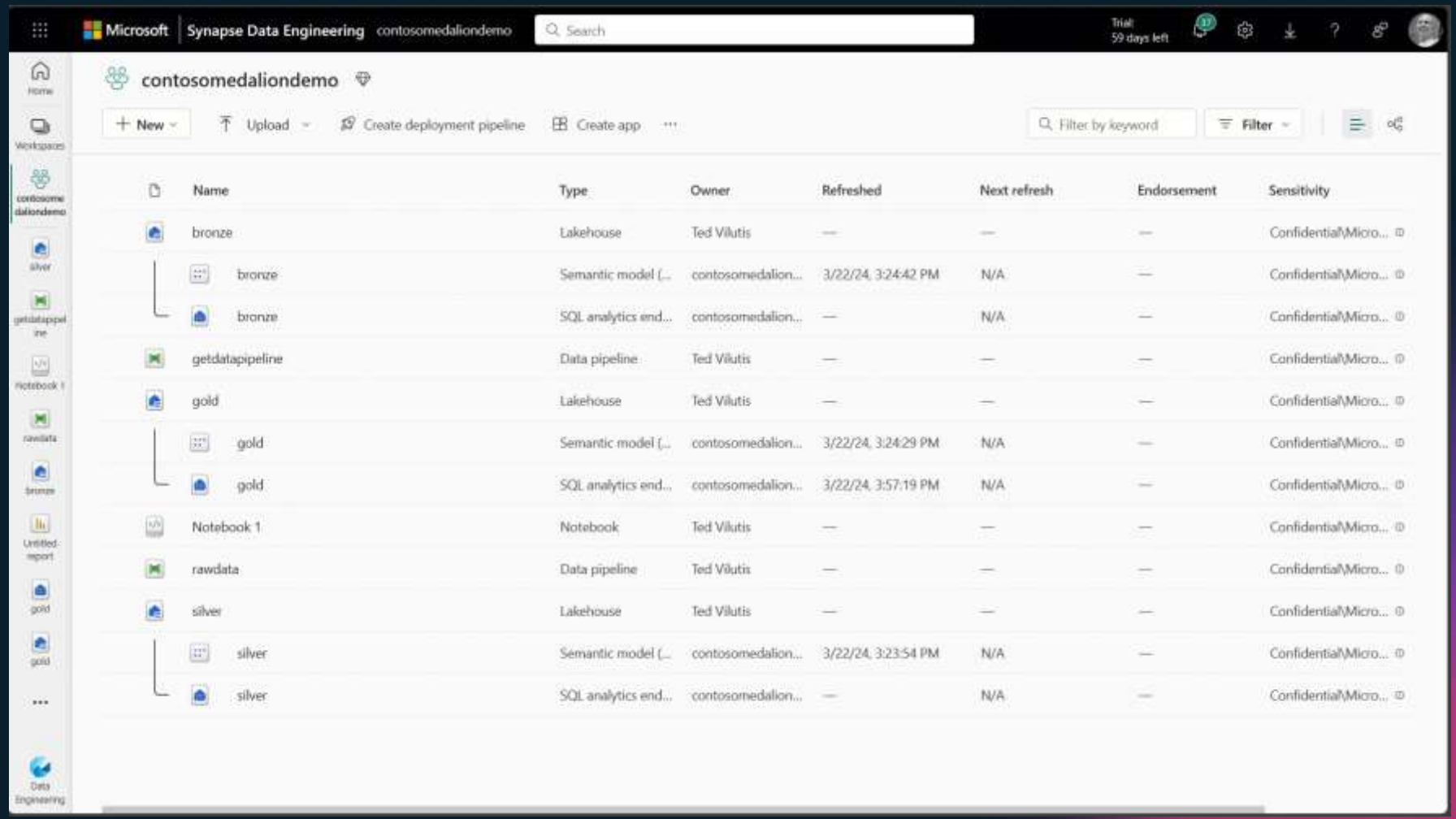
Data Scientist

Data Analyst/Modeller



Moving and transforming data to Silver layer

- Dataflows
- Notebooks



Gold



Gold

- Transformations & aggregations
- Modelled data
- End-user ready

Who

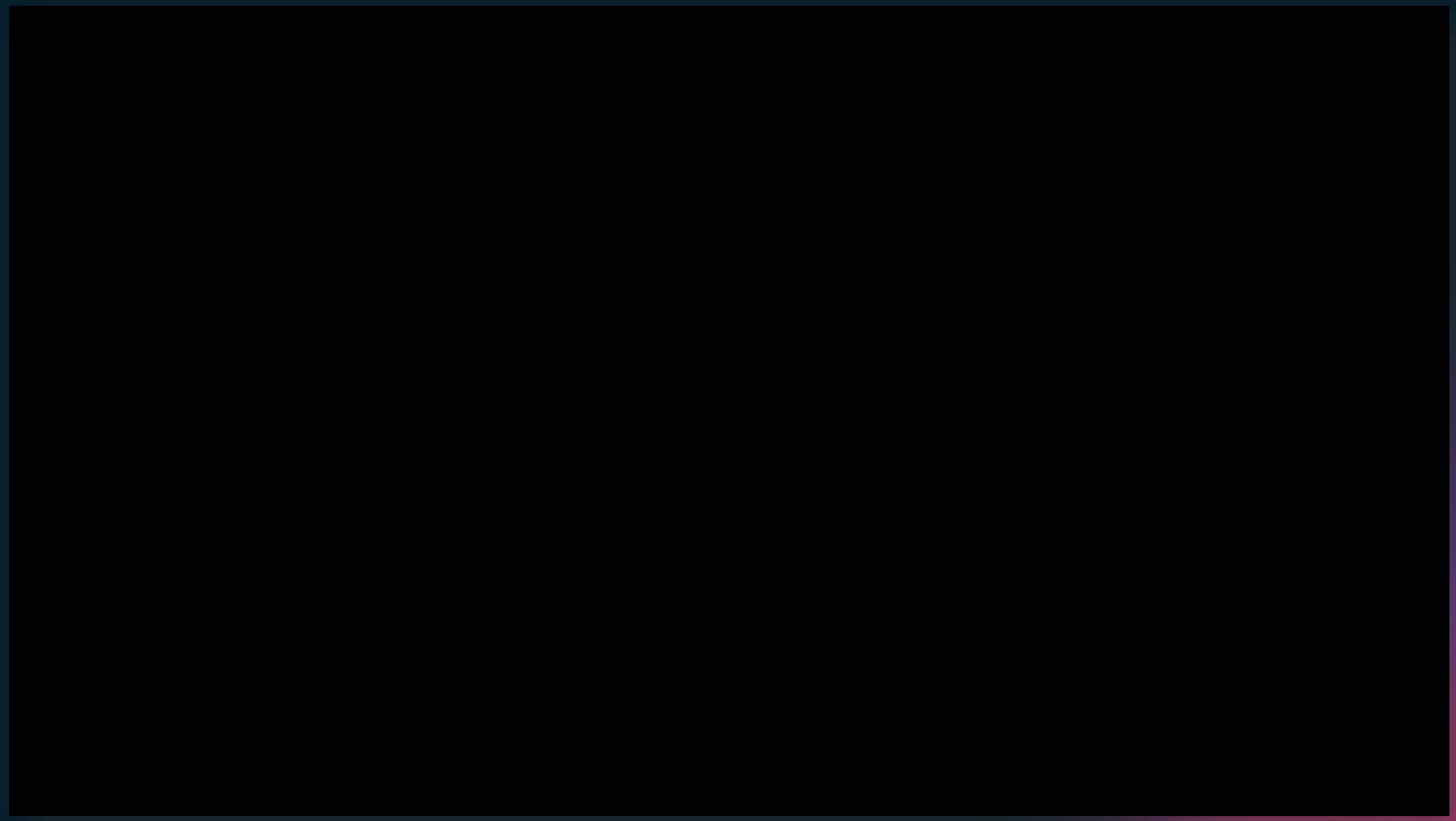
Report Writer/End User

Data Analyst/Modeller



Moving data to Gold layer and Serving

- Lakehouse or data Warehouse
- SQL analytics endpoint
- Semantic model
- Sharing



What about?



Landing

- Immutable Copy
- Allows Bronze to be Schema validation
- Independently secured



Platinum

- Pushing the medallion nomenclature a bit far
- Allows Gold to be enrichment of datasets



Lab

- Separation of data science work
- Can include layers for Exhibit & Model

The problem with the Medallion Architecture

- Confusion with Data Quality categories
- Name your layers whatever you want

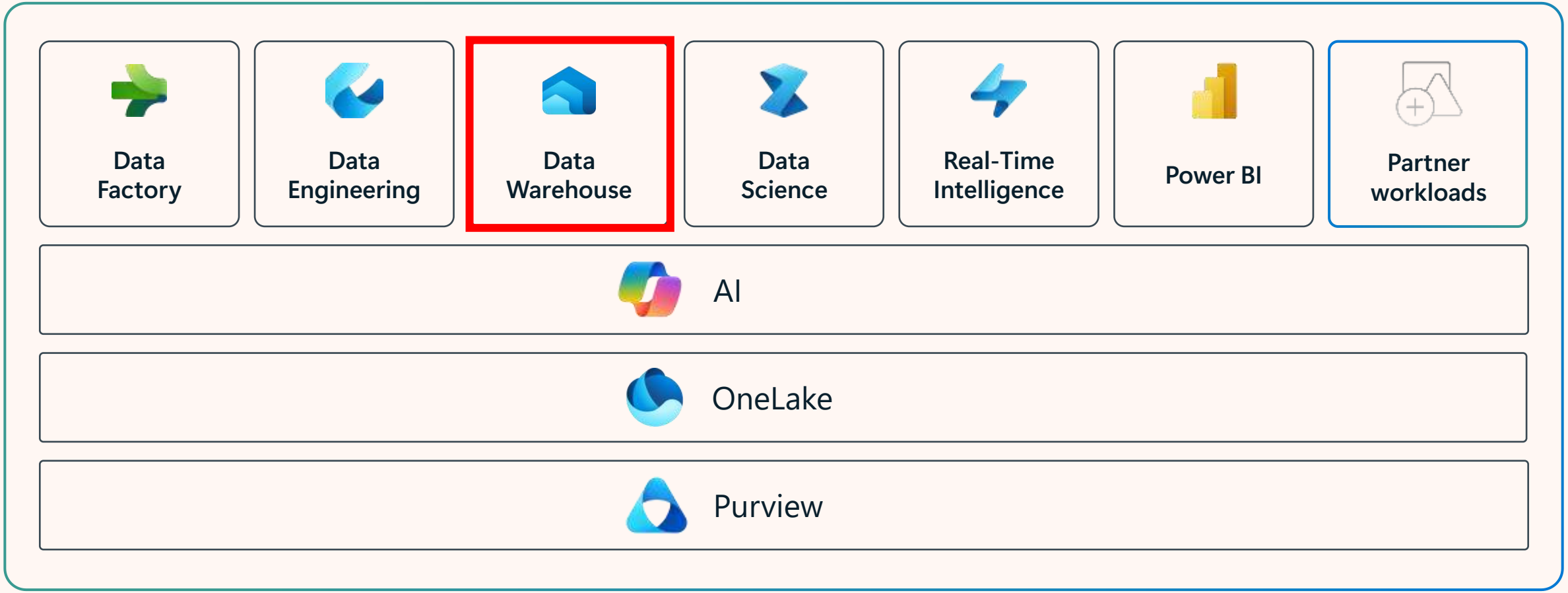
Bronze	Silver	Gold
Raw	Base	Curated
Stage	Conformed	Delivery





Data Warehouse

Microsoft Fabric



What is a data warehouse?

A data warehouse is a central repository of information that can be analyzed to make more informed decisions. Data flows into a data warehouse from transactional systems, relational databases, and other sources, typically on a regular cadence. Business analysts, data engineers, data scientists, and decision makers access the data through business intelligence tools, SQL clients, and other analytics applications.

Data Warehouse key capabilities

Transaction guarantees on open Delta format, industry-leading performance and granular security for easy analysis and reporting

Simple and intuitive Warehouse experiences

Integrated experiences for the Citizen and Pro Dev

Virtual warehouses and cross database querying

Fully integrated Power BI semantic layer

Lake-centric Warehouse

Data in OneLake in open Delta format

No knobs performance at any scale

Easy data recovery and management

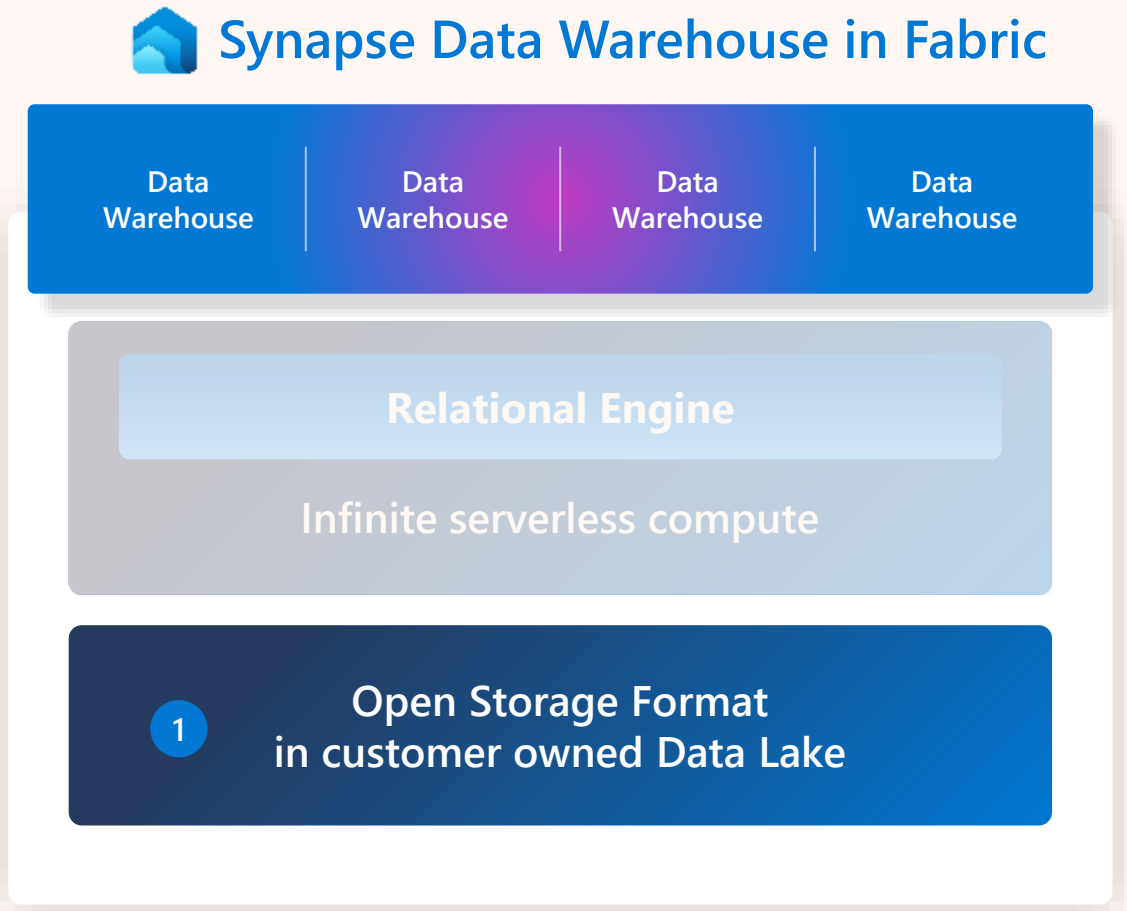
Enterprise-ready platform

End-to-end performance and usage visibility

Always governed

Secure by default

Synapse Data Warehouse
Infinitely scalable and open

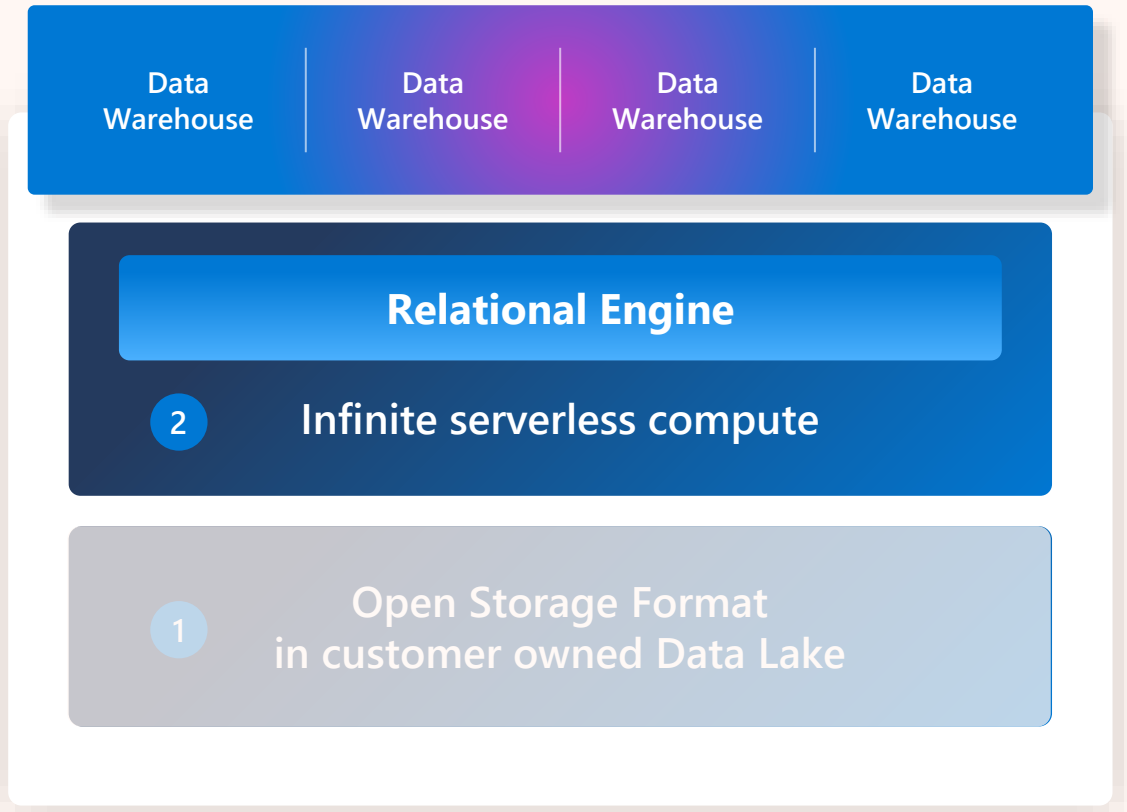


1 Open standard format in an open data lake replaces proprietary formats as the native storage

- First transactional data warehouse natively embracing an open standard format
- Data is stored in Delta – Parquet with no vendor lock-in
- Is auto-integrated and auto-optimized with minimal knobs
- Extends full SQL ecosystem benefits

Synapse Data Warehouse
Infinitely scalable and open

Synapse Data Warehouse in Fabric

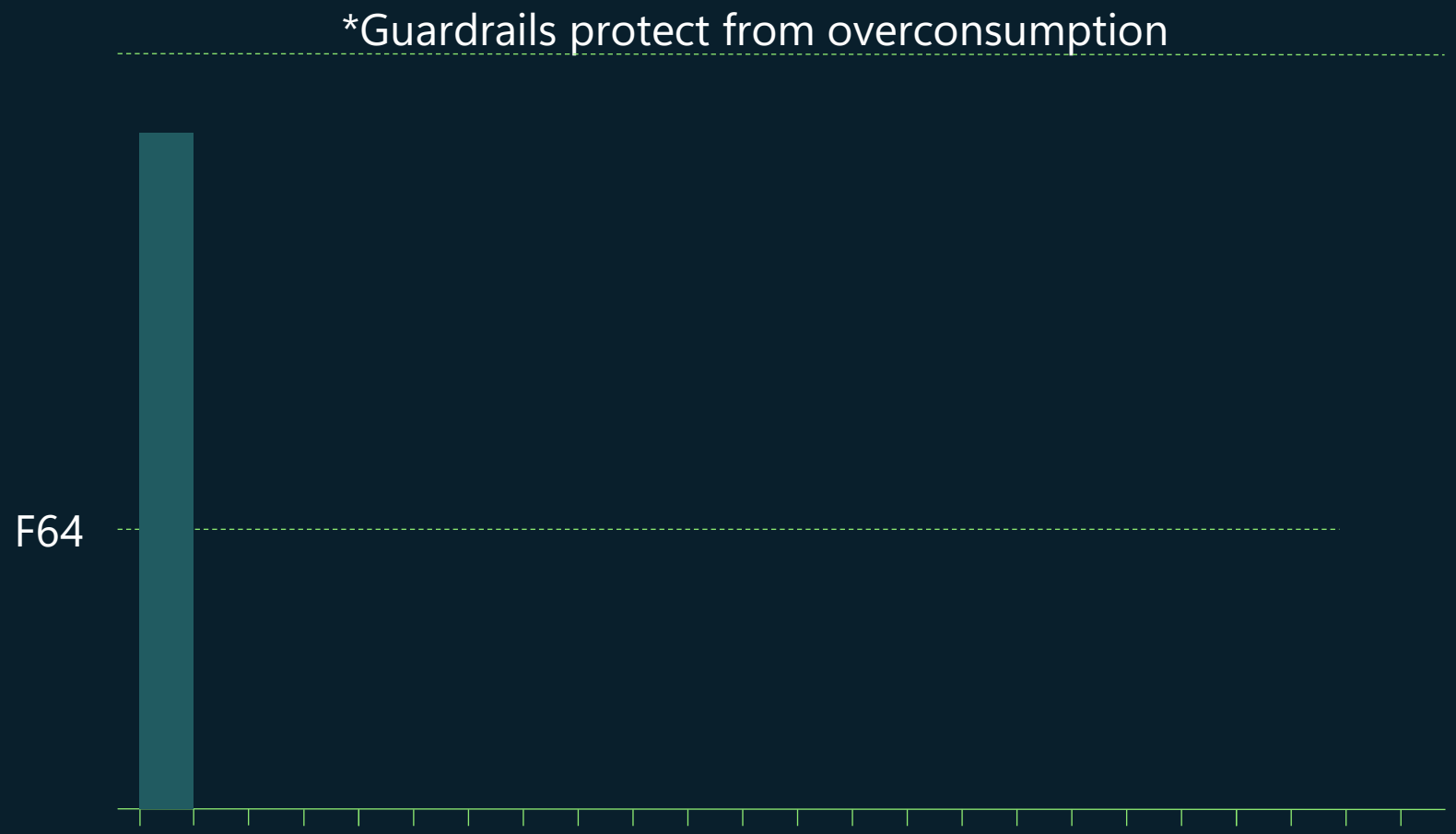


2 Dedicated clusters are replaced by serverless compute infrastructure

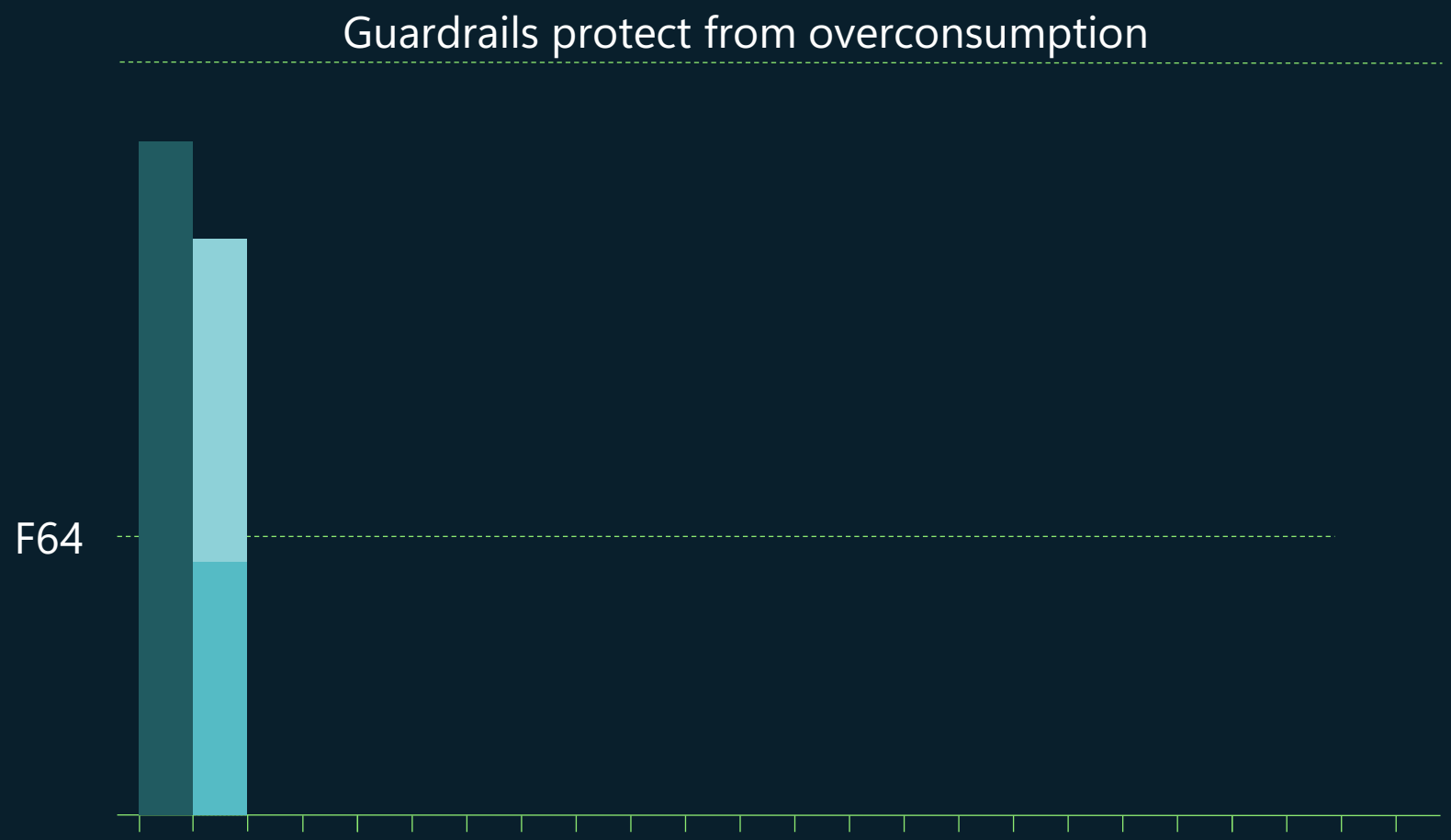
- Physical compute resources assigned within milliseconds to jobs
- Infinite scaling with dynamic resource allocation tailored to data volume and query complexity
- Instant scaling up/down with no physical provisioning involved
- Resource pooling providing significant efficiencies and pricing

Always optimal query performance

- Fabric bursting automatically allocates resources as needed to achieve optimal performance
- Query performance is always the same regardless of SKU*
- F2 capacity can run 1TB workloads thanks to bursting!

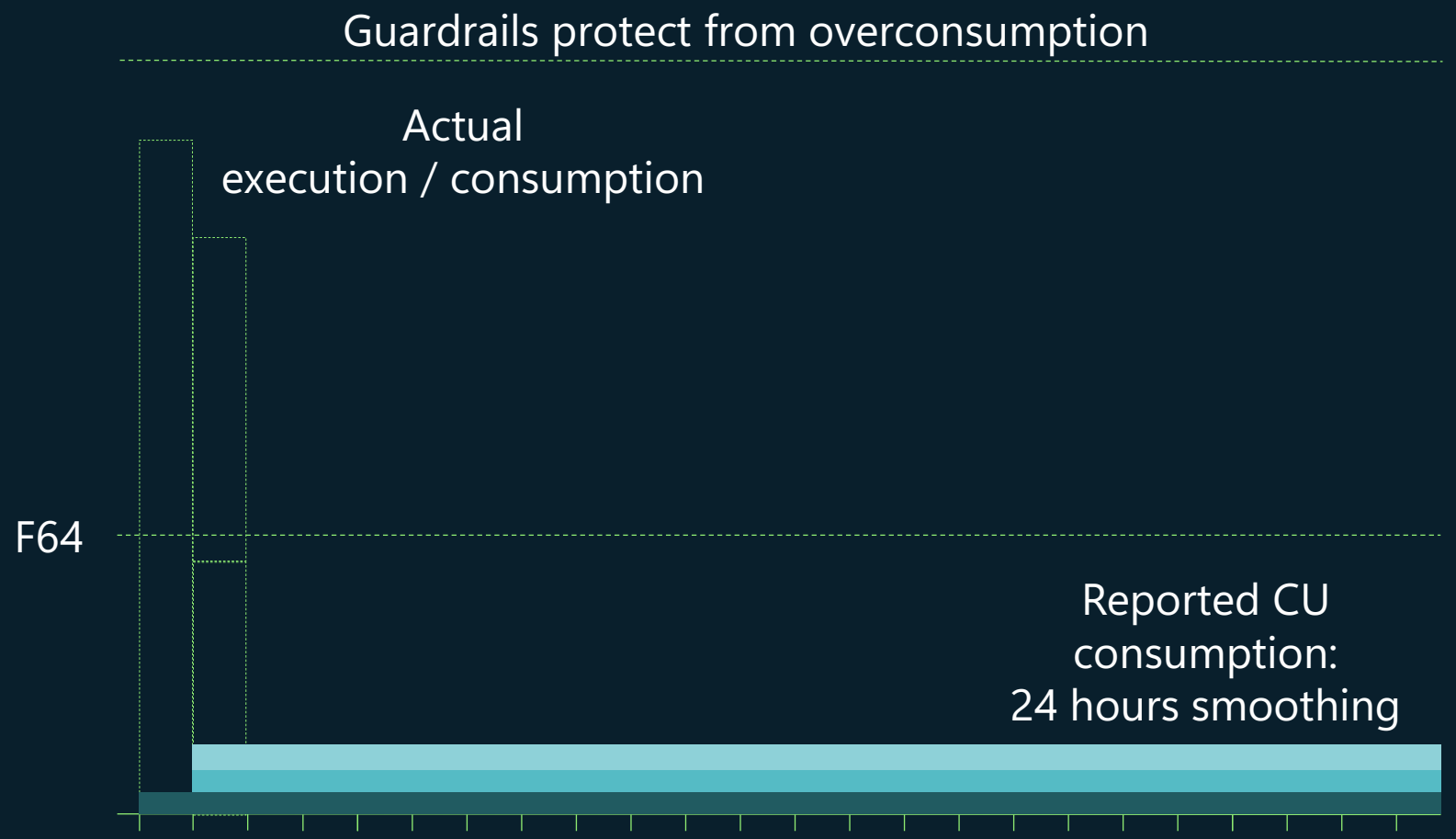


No more sizing for peaks



No more sizing for peaks

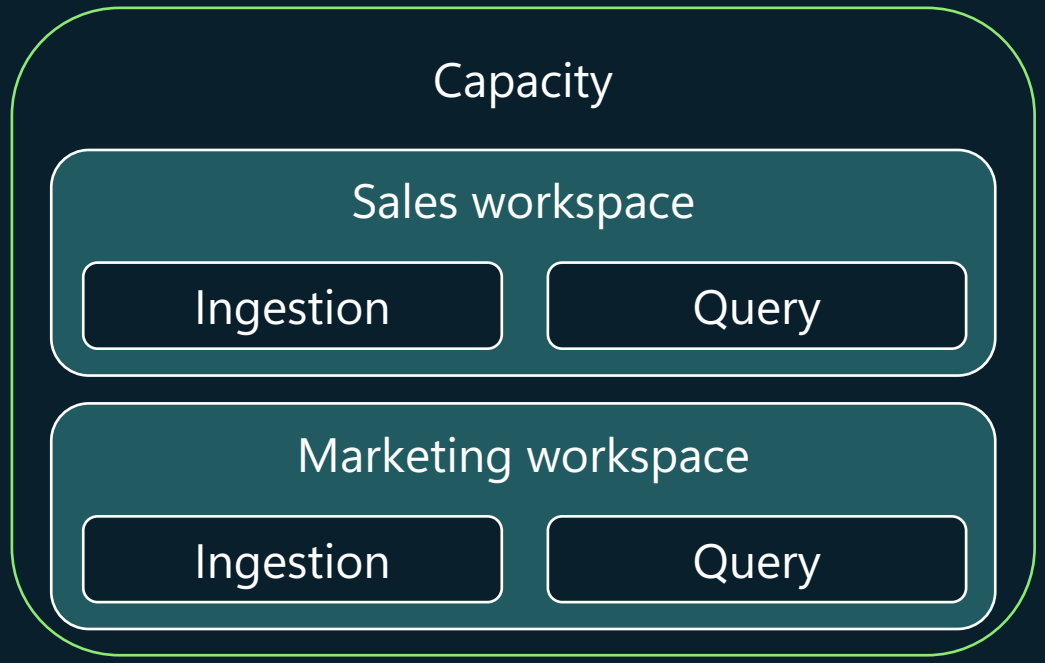
- Smoothing to future time windows to avoid throttling
- Temporary peaks are handled without DBA intervention or increasing capacity size
- Use Capacity metrics app to track consumption
- F64 (trial) capacity can run TPCH 10TB power run thanks to bursting and smoothing!



Predictable workload performance

With automatic workload management and workspace as natural isolation boundary

- Predictable ETL performance based on compute isolation
- Scales independently



Demo

Connecting to SQL in Fabric

Virtual warehouses and cross database querying

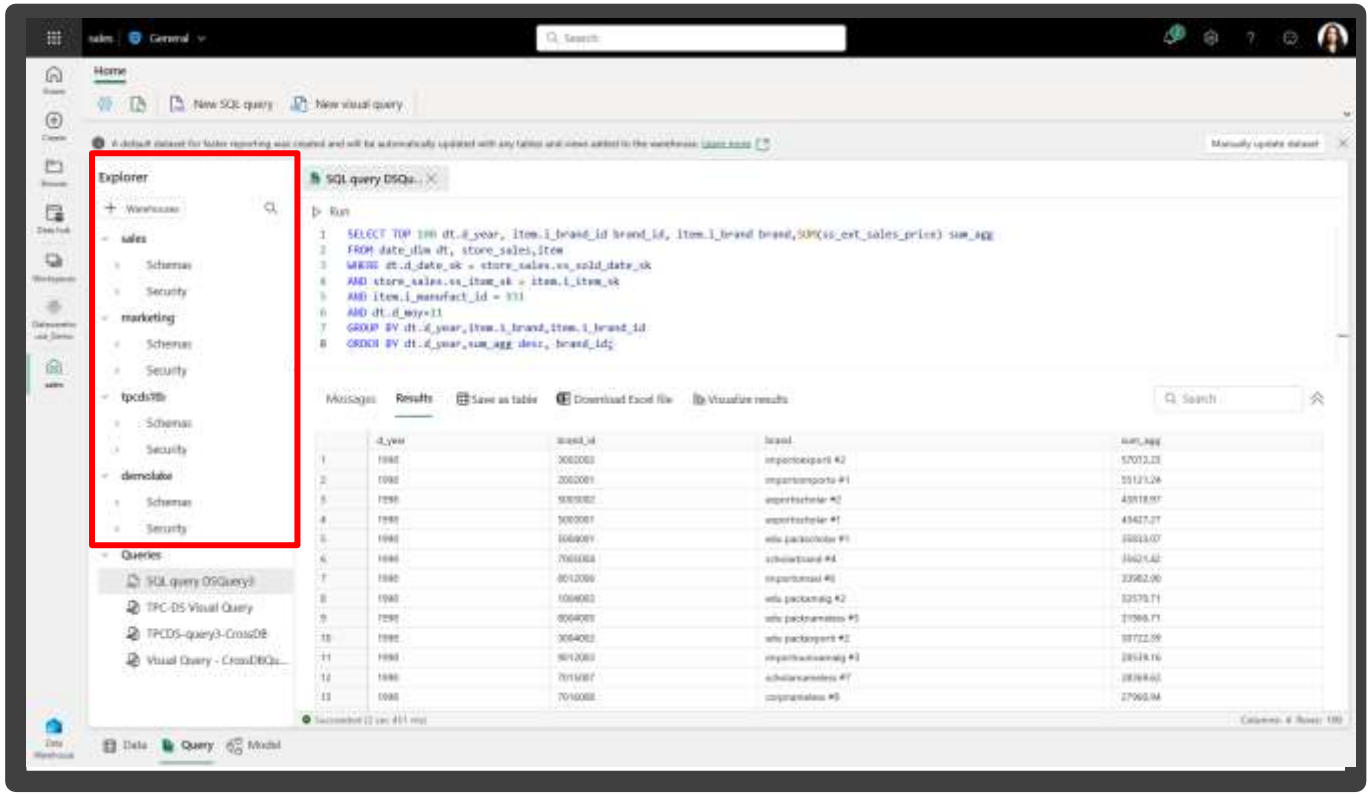
Simple and intuitive warehouse experience

Eliminate the need to duplicate data from any Lakehouse to any Warehouse

Customers can query data in the lake without the need for data movement, reducing costs.

Key Capabilities:

- Build virtual warehouses by creating Lakehouses with shortcuts to data in the Lake
- Query across Warehouse and Lakehouse SQL Endpoint with zero data movement
- Write T-SQL query with 3-part-naming for cross-joining
- Build queries visually using the Visual Query Editor



Easy to get data into the Warehouse



Zero ETL

Use Shortcuts or Mirroring to accelerate data access

COPY INTO
CTAS
INSERT … SELECT
CTE
Stored Procedures

SQL for data integration

Write T-SQL code for transforms



Pro Code data integration

Build Pipelines for ETL and Automation



No Code data integration

Data prep using Dataflows Gen2

Data loading and transforms at scale



High scale data loading from any source.

Data managed by Warehouse compute engine provides full multi-table transaction guarantees

Key Capabilities:

- Integration with Data Factory Pipelines
- Data published to OneLake as Delta tables
- Support for multi-statement transactions
- Supports snapshot isolation
- T-SQL support for:
  - COPY command
  - CTAS
  - INSERT … SELECT
  - Common Table Expressions
- Locking:
  - Do not block INSERT and SELECT statements
  - UPDATE and DELETE do not lock tables

```
COPY INTO table_name.[(Column_list)]
FROM '<external_location>' [,...n]
WITH
 (
 [FILE_TYPE = {'CSV' | 'PARQUET' '} ]
 [,CREDENTIAL = (AZURE CREDENTIAL) ]
 [,ERRORFILE = '[http(s)://storageaccount/container]/errorfile_directory[/]]'
 [,ERRORFILE_CREDENTIAL = (AZURE CREDENTIAL) ]
 [,MAXERRORS = max_errors ]
 [,COMPRESSION = { 'Gzip' | 'Snappy'}]
 [,FIELDQUOTE = 'string_delimiter']
 [,FIELDTERMINATOR = 'field_terminator']
 [,ROWTERMINATOR = 'row_terminator']
 [,FIRSTROW = first_row]
 [,ENCODING = {'UTF8'|'UTF16'}]
 );

CREATE TABLE [warehouseA].[dbo].[FactInternetSales_new]
AS
SELECT * FROM [warehouseB].[dbo].[FactInternetSales];

INSERT INTO dbo.EmployeeSales
SELECT sp.BusinessEntityID, c.LastName, sp.SalesYTD
FROM warehouseA.Sales.SalesPerson AS sp
INNER JOIN warehouseB.Person.Person AS c
    ON sp.BusinessEntityID = c.BusinessEntityID
WHERE sp.BusinessEntityID LIKE '2%'
ORDER BY sp.BusinessEntityID, c.LastName;
```

Alternative to MERGE Statement

Screenshot of partial code

```
 9
10  UPDATE
11      dim
12  SET
13      HackneyLicenseBkey = src.HackneyLicenseBKey,
14      HackneyLicenseCode = src.HackneyLicenseCode
15  FROM
16      dbo.HackneyLicense dim
17          INNER JOIN
18      temp.HackneyLicenseNew src
19          on dim.HackneyLicenseID = src.HackneyLicenseID;
20  GO
```





Time to build together!

Lab 2 and 3

Create WH and load ingest the data

Easy to get data into the Warehouse



Zero ETL

Use Shortcuts or Mirroring to accelerate data access

COPY INTO
CTAS
INSERT … SELECT
CTE
Stored Procedures

SQL for data integration

Write T-SQL code for transforms



Pro Code data integration

Build Pipelines for ETL and Automation



No Code data integration

Data prep using Dataflows Gen2

Mirroring

Fabric Mirroring

Seamlessly connect your databases and data warehouses to OneLake



Azure
Cosmos DB



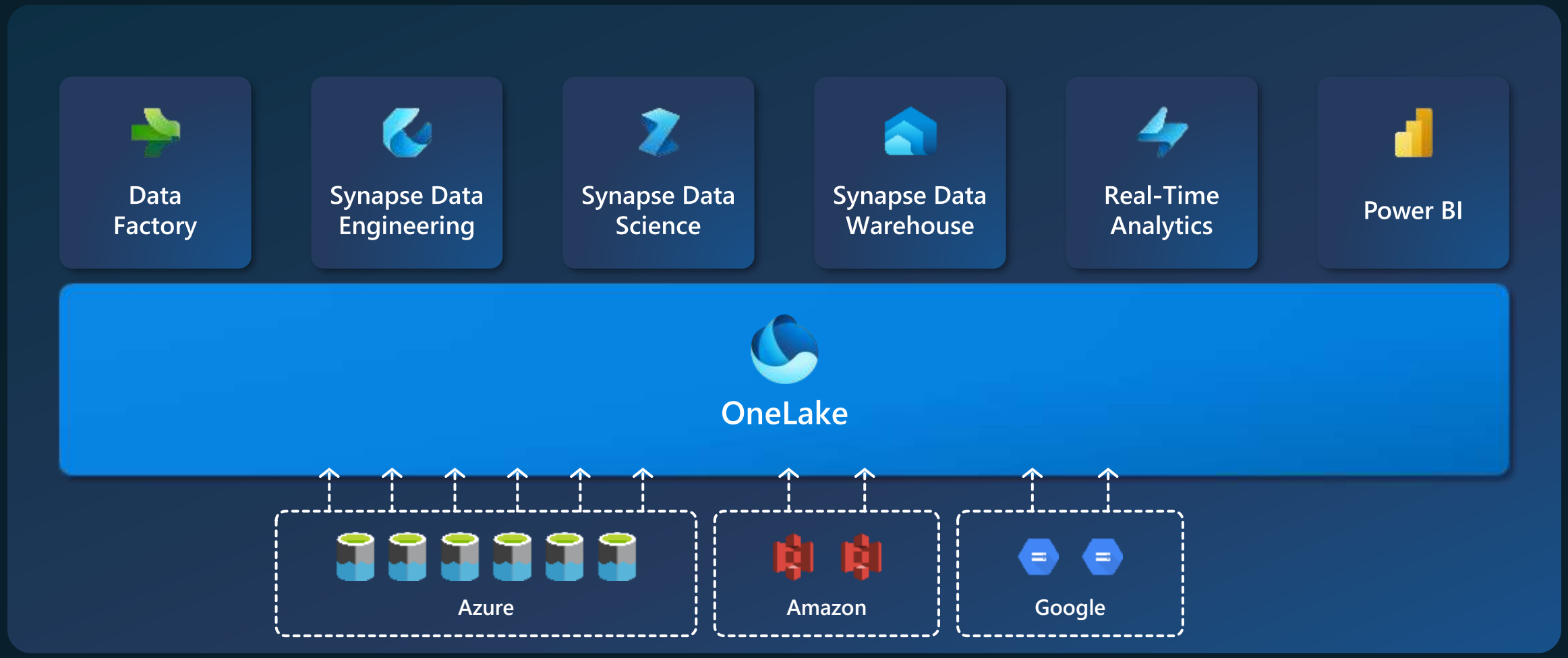
Azure
SQL DB



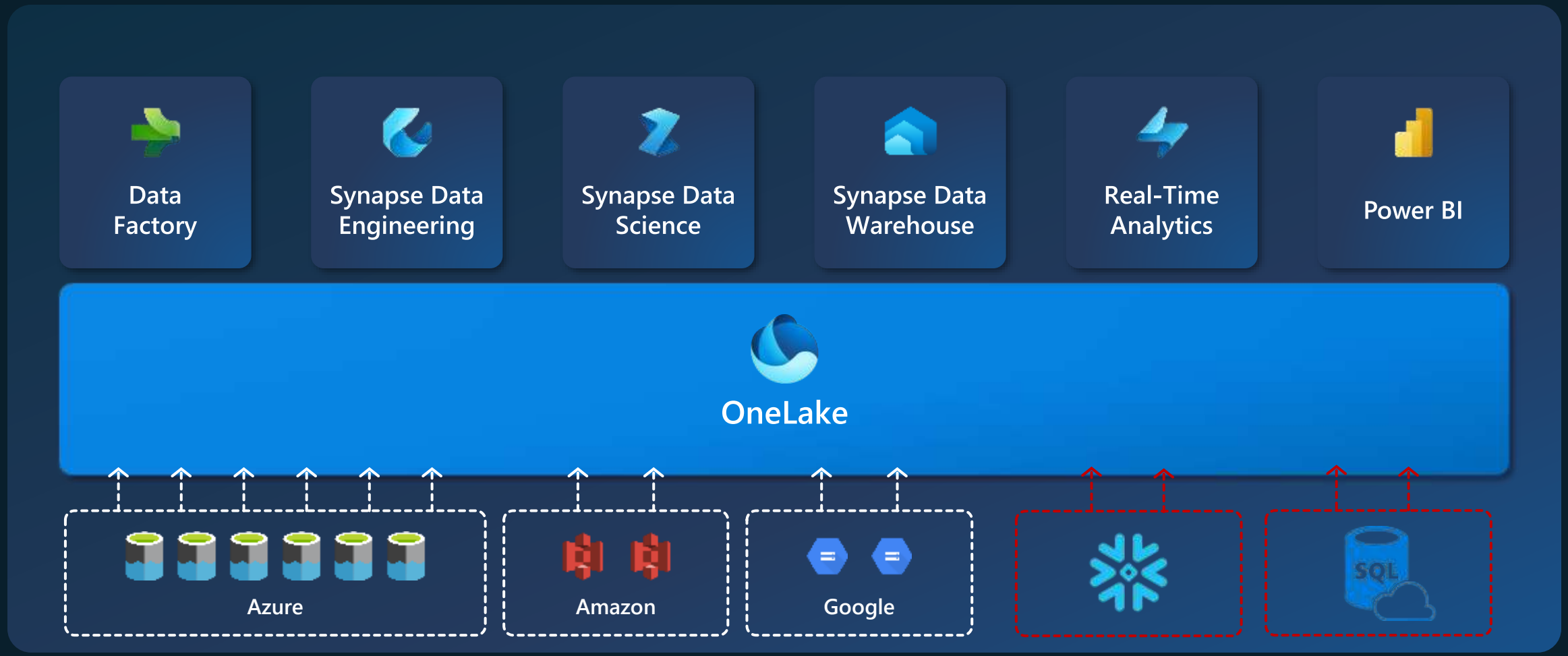
Snowflake

And More...

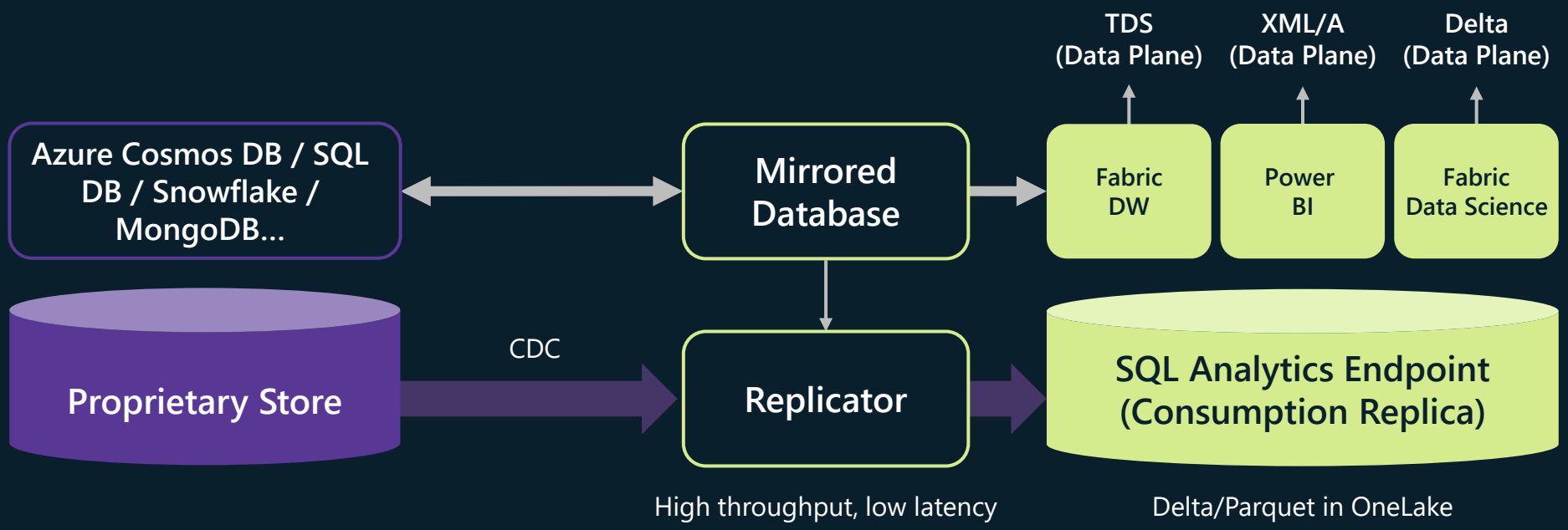
OneLake is a multi-cloud data lake



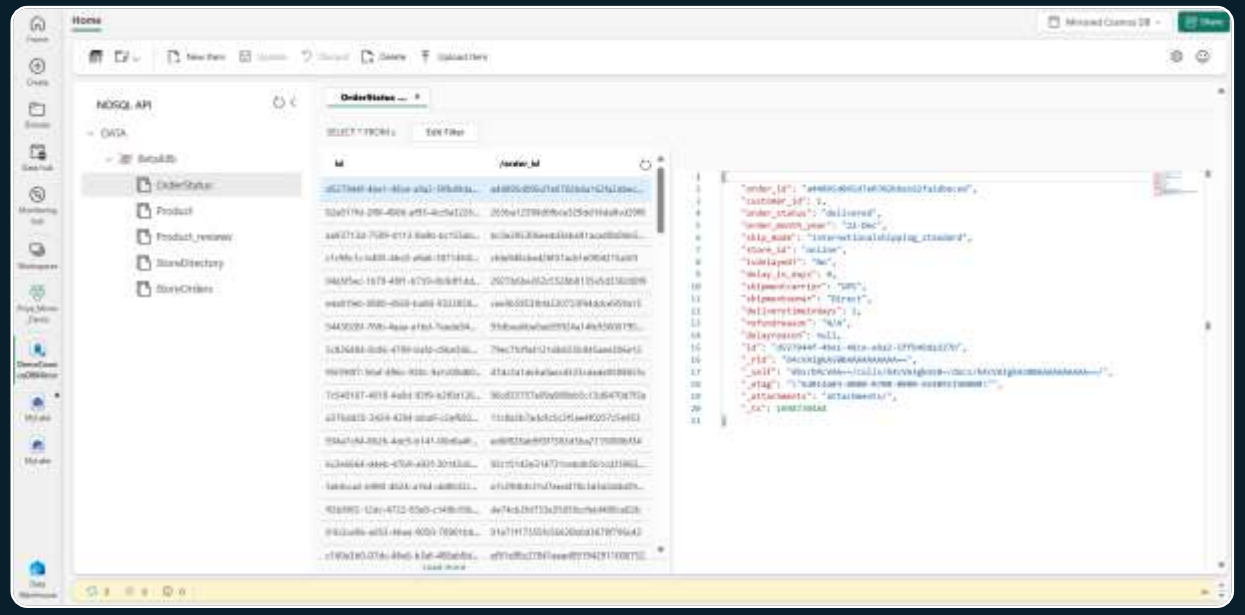
OneLake is a multi-cloud data lake



Mirroring architecture



Management of the data source

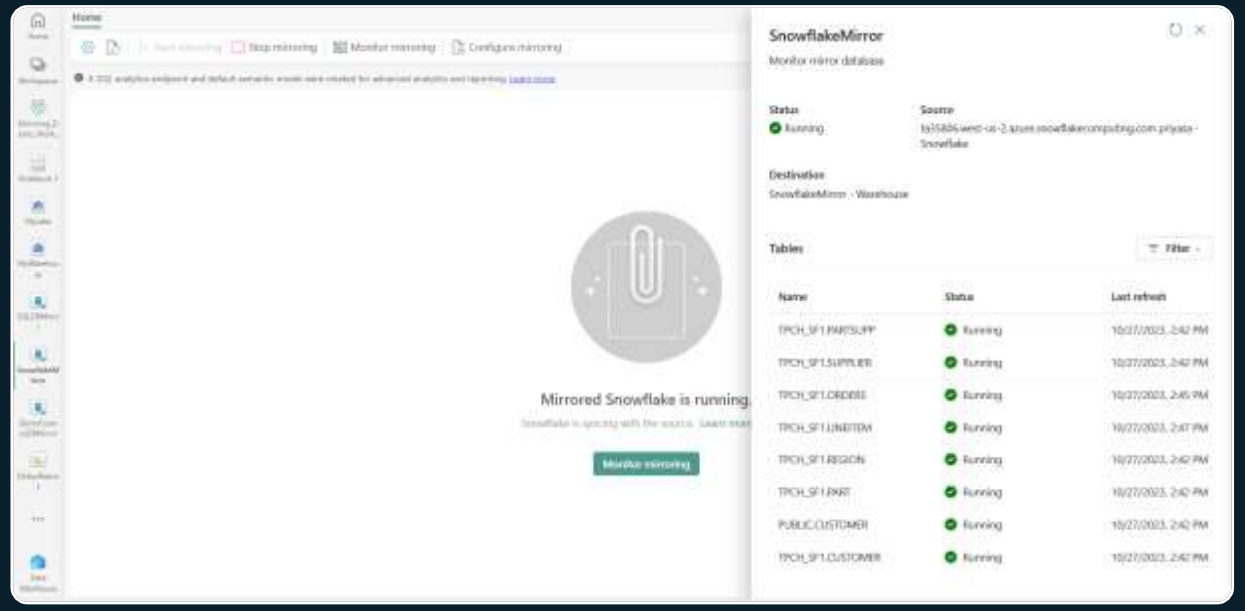


Database is available in Fabric as a Mirrored database.

Source database can be accessed and managed from right within Fabric.

Familiar database editors provide the ability to manage the database with changes going to the source even across clouds.

Near real-time replication

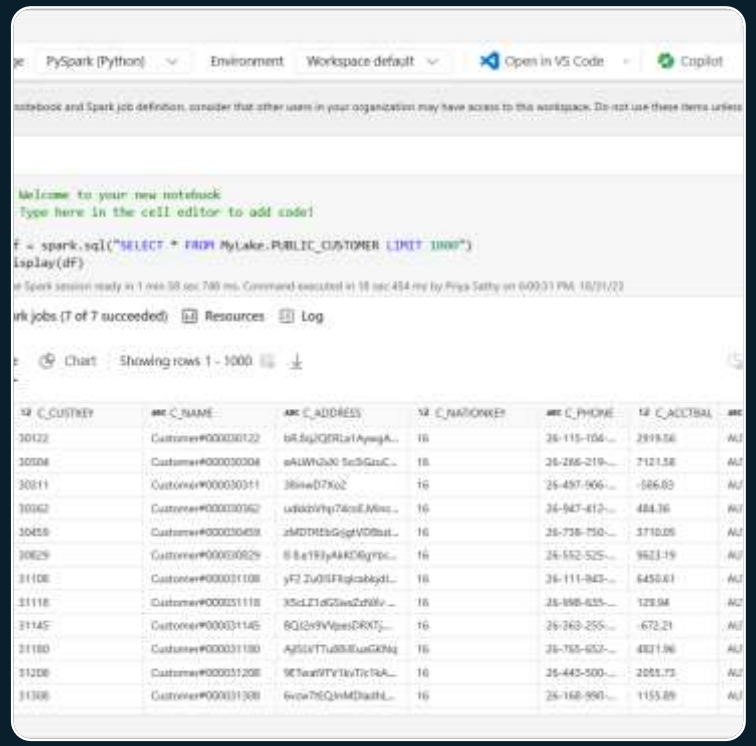


- Granular controls for configuring mirroring.
- An initial snapshot is created after which data is kept in sync with every transaction.
- Replication uses the database CDC.
- Transforms to Delta tables are automatic.
- Intelligent logic determines when the source has changed, ensuring compute isn't used unnecessarily.
- Detailed monitoring is available to view mirroring operations.

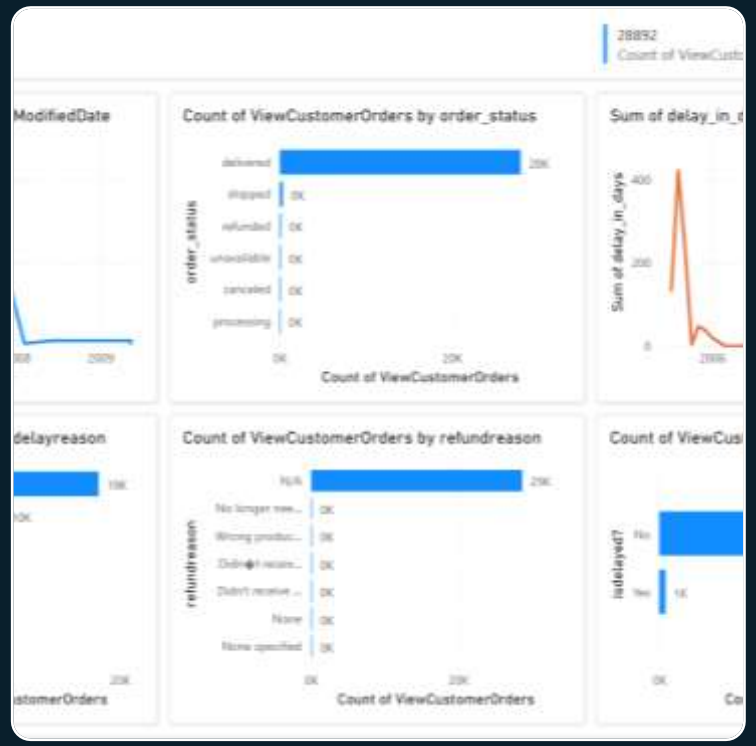
Fabric experiences unlocked on real-time data



Cross-database querying

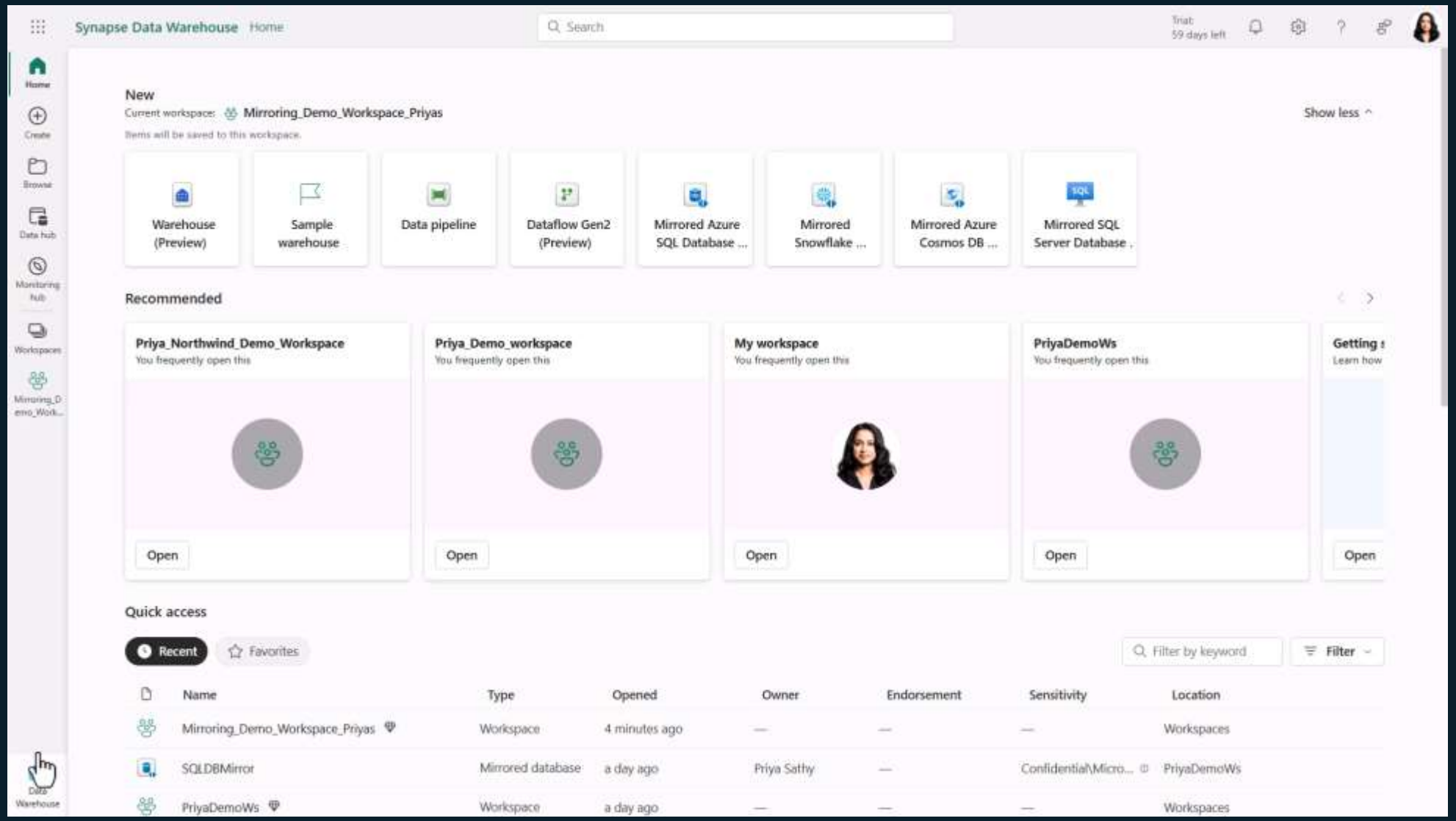


Data Science experiences



Power BI Direct Lake mode

Demo – Mirroring (3min)



Roadmap

Available today



Azure Cosmos DB



Snowflake



Azure SQL DB

CY24 and beyond

















And More…



Time to build together!

Lab 4

Orchestration

Time travel

Query data warehouse tables as it existed in the past up to 7 calendar days

- Historical rend analysis
- Troubleshooting
- Stable reporting





Statement Level

- In a SELECT statement, state the timestamp only once for the entire query.

- Results of the query will reflect the state of the data at the timestamp specified in the query.

```
SELECT TOP (10)
   FS.[CustomerKey],
   DC.[Customer],
   SUM(FS.TotalIncludingTax) AS TotalSalesAmount
FROM
   [dbo].[dimension_customer] AS DC
INNER JOIN
   [dbo].[fact_sale] AS FS ON DC.[CustomerKey] =
FS.[CustomerKey]
GROUP BY
   FS.[CustomerKey],
   DC.[Customer]
ORDER BY
   TotalSalesAmount DESC
   OPTION (FOR TIMESTAMP AS OF'2024-05-04T02:05:06.280');
```

OPTION clause for time travel can be used only with **SELECT** statements



Query a single table



Complex queries with joins and stored procedures



Complex queries with group by and order by clause



Pre- created views

- CTAS and INSERT INTO SELECT are not supported for time travel with OPTION clause.



Limitations

- Data will not be returned if the table is ALTERED between now and the point in time submitted for time travel.
- Data will not be returned if the table is queried beyond the retention period of seven calendar days.
- Data will not be returned if the table did not exist at the point in time submitted for time travel.

Zero-Copy Clone

A size of metadata operation



Test

Recover

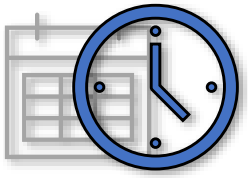
Collaborate



Copy near-instantaneously

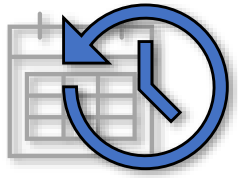


Save on storage costs



Current Point in Time

Creates a clone as of the present moment



Previous Point in Time

Creates a Clone as of a past point in time

Granularity



Table

Once created, clones exist as separate entities from their source

Scenario	Source Table	Cloned Table
New attribute added to source table	👍	👎
New records added to source table	👍	👎
New attribute added to cloned table	👎	👍
New records added to cloned table	👎	👍

Dropping Table Clones:

- Table clones can be dropped at any point in time.
- The source table can be dropped while the clone table exists.



Table clones of the Lake warehouse tables cannot be created.

Column-level security(CLS) is automatically transferred from the source to the cloned table

Automate your interactions with Fabric Warehouse



Simple and intuitive warehouse experience

REST APIs let you automate Fabric procedures and processes

Public APIs enable:

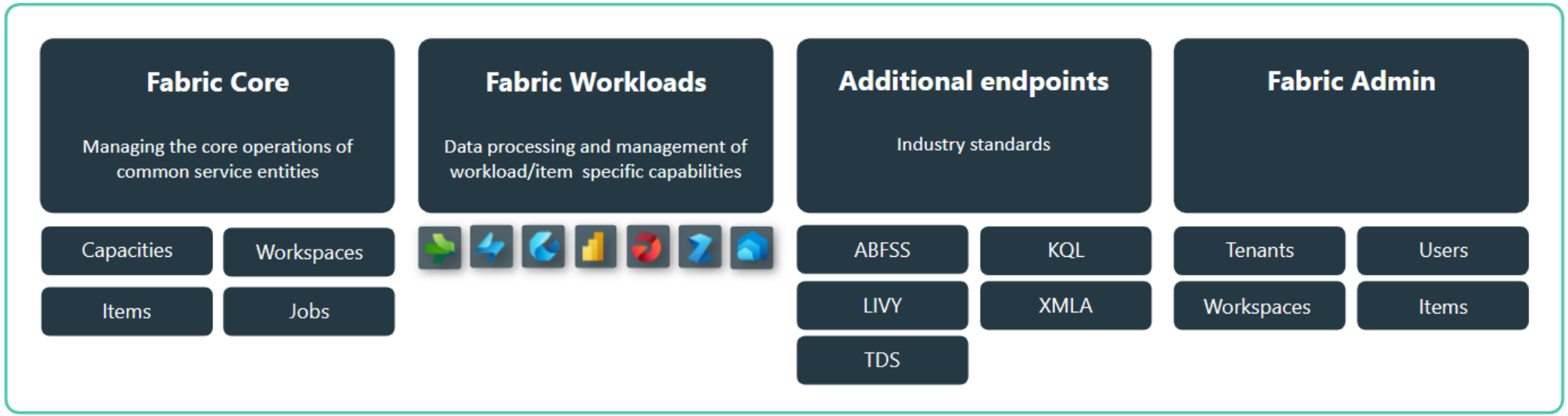
- End-to-end fully automated interactions with Microsoft Fabric
- Integration of Microsoft Fabric into external web applications
- Scaling solutions in a quick and easy fashion

JSON

```
{
    "id": "<root_activity_Id>",
    "type": "Warehouse",
    "displayName": "<warehouse_name>",
    "description": "<description>",
    "workspaceId": "<workspace_Id>",
    "properties": {
        "connectionInfo": "<sql_connection_string>",
        "createdDate": "2023-09-28T22:52:13.78",
        "lastUpdatedTime": "2023-10-04T22:56:33.283",
    }
}
```

REST APIs

Automate your Fabric procedures and processes to complete your tasks more efficiently



My colleague left the company and he is Warehouse owner!

- Leverage Takeover API to change ownership

Data warehouse recovery
Restore in-place



Process and techniques used to protect and recover data in the event of data loss or corruption

Data warehouse recovery is a 2-step process
1. Creation of a Restore-point
2. Recover using the restore point



Restore point in synapse data warehouse can be leveraged to restore the data warehouse to its previous state by replacing or overwriting on the existing data warehouse . It is a “size of metadata" operation.



System generated restore point

Automatically creates restore points every **8hrs**

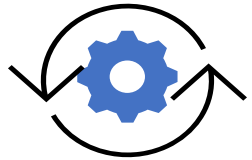


User-defined restore point

Allows for manual creation of restore points



Retention of restore points refers to the period until which the restore points will be retained.



Default retention of 7-days

Restore point is deleted when it hits the 7-day retention period

Monitoring through Dynamic Management Views



Get insights into SQL query lifecycle

Customers can view query performance and easily terminate any noisy or costly queries

Key Capabilities:

- Use Dynamic Management Views (DMVs) to:
  - Find open connections through sys.dm_exec_connections
  - Find users through sys.dm_exec_sessions
  - Find queries executed by user through sys.dm_exec_requests
  - Find all completed request through queryinsights.exec_requests_history (Transact-SQL)
  - Find query execution times through queryinsights.long_running_queries (Transact-SQL)
  - Find frequently run queries through queryinsights.frequently_run_queries (Transact-SQL)
  - Use KILL command to terminate session

```
-- Identifies long-running queries in order of which query has taken the longest since it has arrived, and which user executed the request.

SELECT r.request_id, r.session_id,
r.start_time, r.total_elapsed_time,
s.login_name
FROM sys.dm_exec_requests AS r
JOIN sys.dm_exec_sessions AS s
ON s.session_id = r.session_id
WHERE r.status = 'running'
ORDER BY r.total_elapsed_time DESC


-- Use the KILL command on the session with the long-running query.

KILL '[SESSION_ID WITH LONG-RUNNING
QUERY]'
```

Monitoring through Query Activity





Time to build together!

Lab 5 and 6

Time travel, clone and monitoring

Joining system catalogs with user tables



- Not possible yet, but there is a workaround
- Craft dynamic SQL that will mimic
  INSERT INTO user_tbl SELECT FROM system_catalog

Zero Maintenance



Simple and intuitive warehouse experience

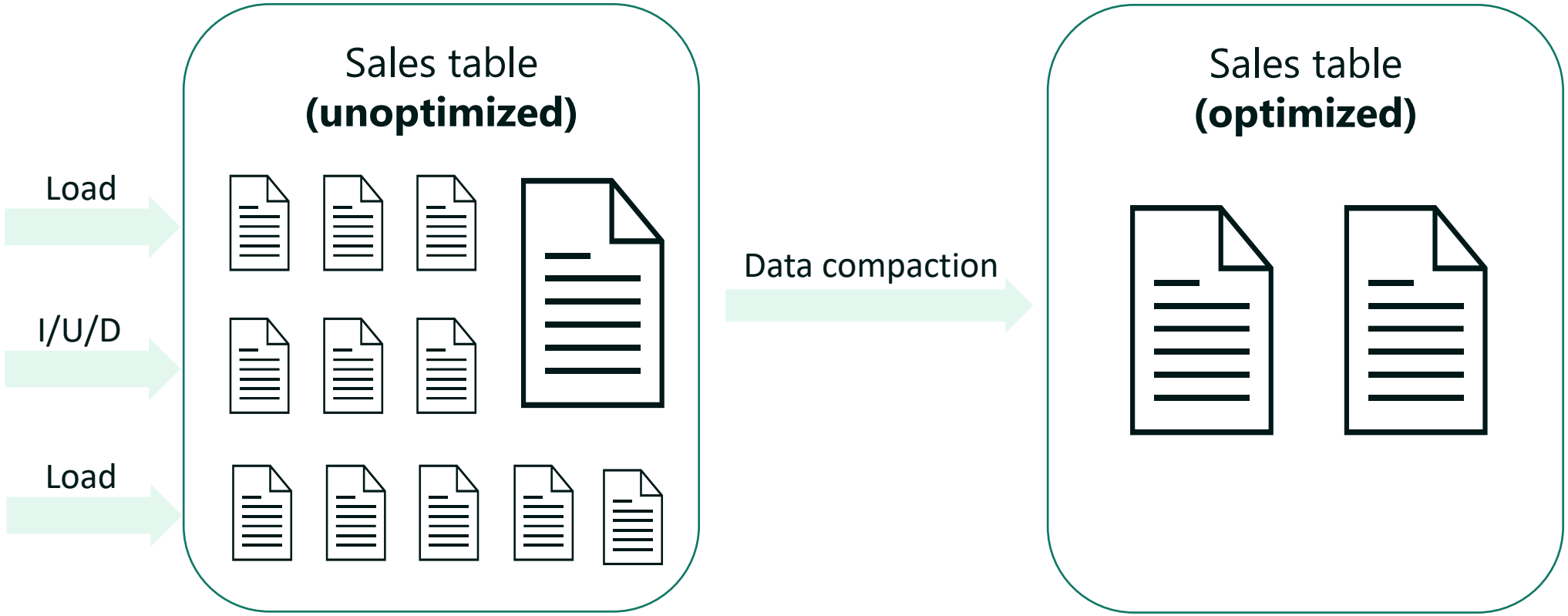
Automatic maintenance for you

Challenge: table layout deteriorates over time with DMLs.

Table layout is automatically maintained for optimal performance:

- Automatic compaction
- Automatic log checkpointing

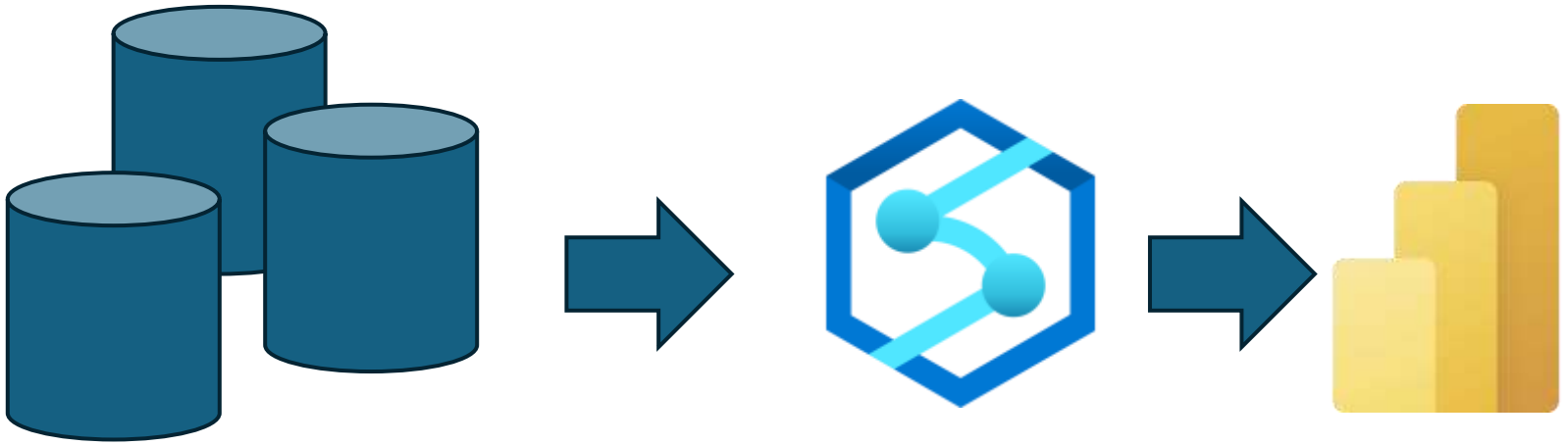
Zero maintenance with automatic data compaction





Direct Lake

Traditional Microsoft BI Architecture



Source Data

Relational Serving Layer (eg Synapse, Azure SQL DB, Snowflake, BigQuery etc)

Power BI

Direct Lake Fundamentals

- Only one data source can be used with Direct Lake
- Direct Lake semantic model starts life with no data in memory
- Data is *paged* into semantic model on demand triggered by query
- Tables can have resident and non-resident columns
- Column data can get evicted for multiple reasons
- Direct Lake fallback to SQL for *suitable* sub-queries
- “Framing” for data consistency in Power BI reports

Direct Lake limitations (for now)

- No calculated columns or calculated tables
- No composite models
  - Although calculation groups and field parameters are now allowed
- Can only be used with tables, not views
- Can only be used with security defined in the semantic model
- Web authoring experience (or 3rd party tool)
- Not all data types supported
  - No structured data types, binary or GUID columns
  - DateTime relationships not supported

Direct Lake

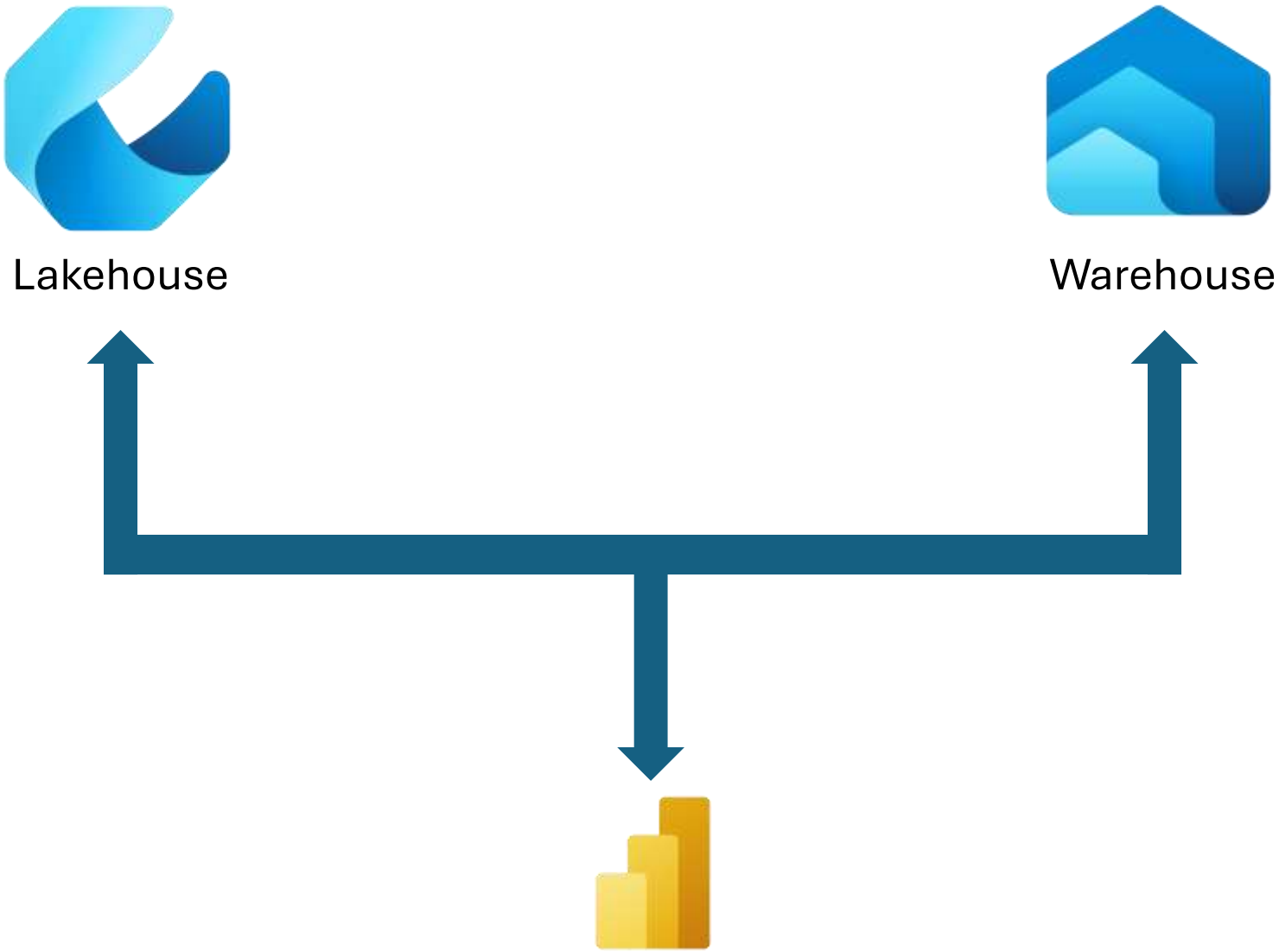


Lakehouse

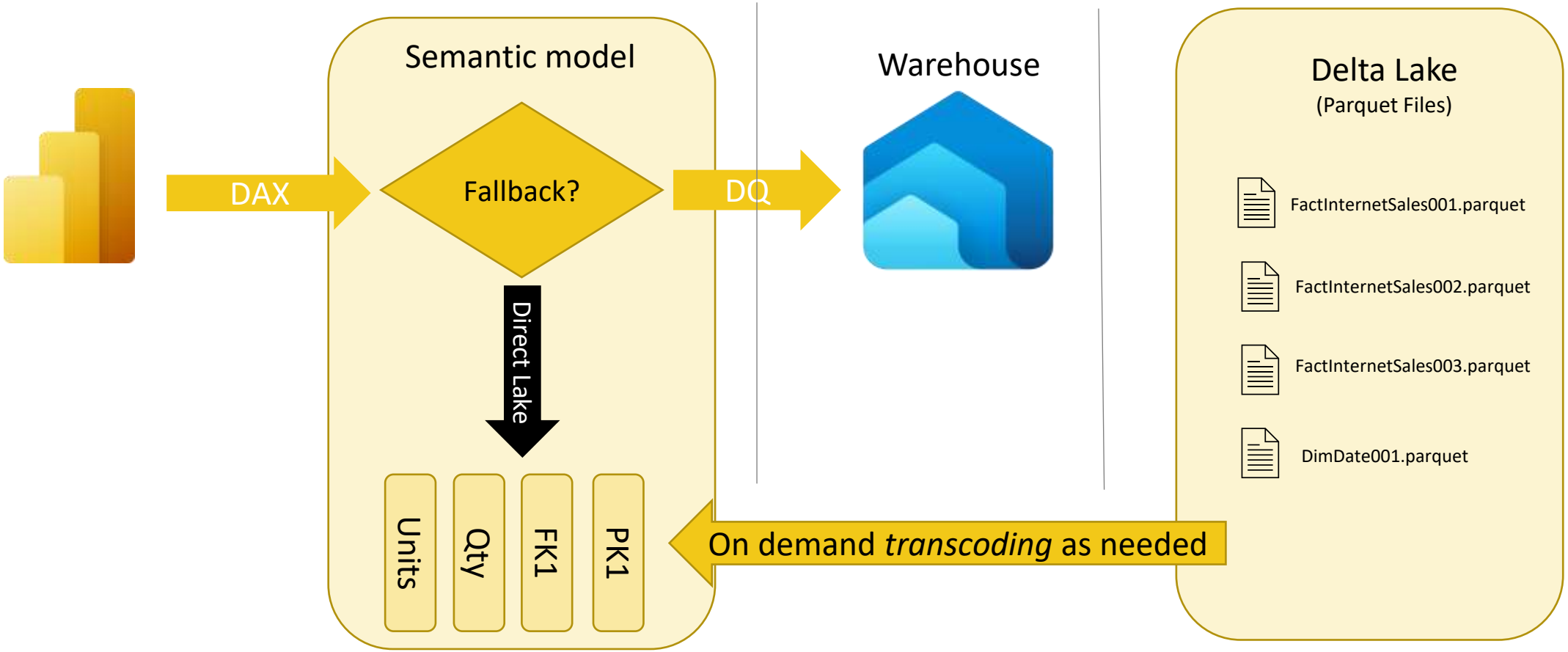




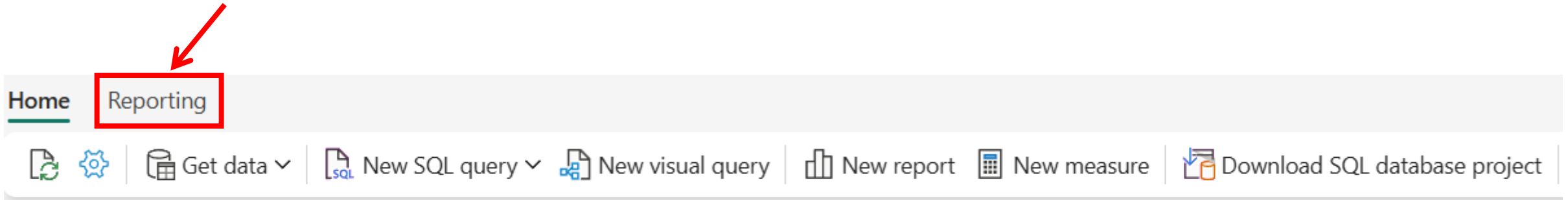
Direct Lake

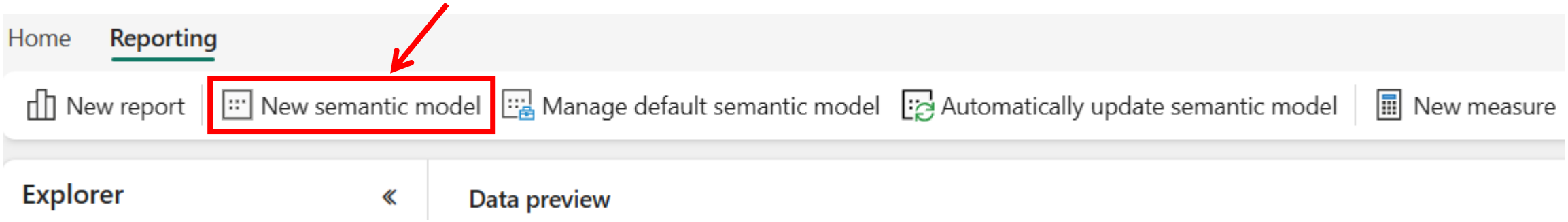


Fallback to DirectQuery



Creating a Direct Lake semantic model







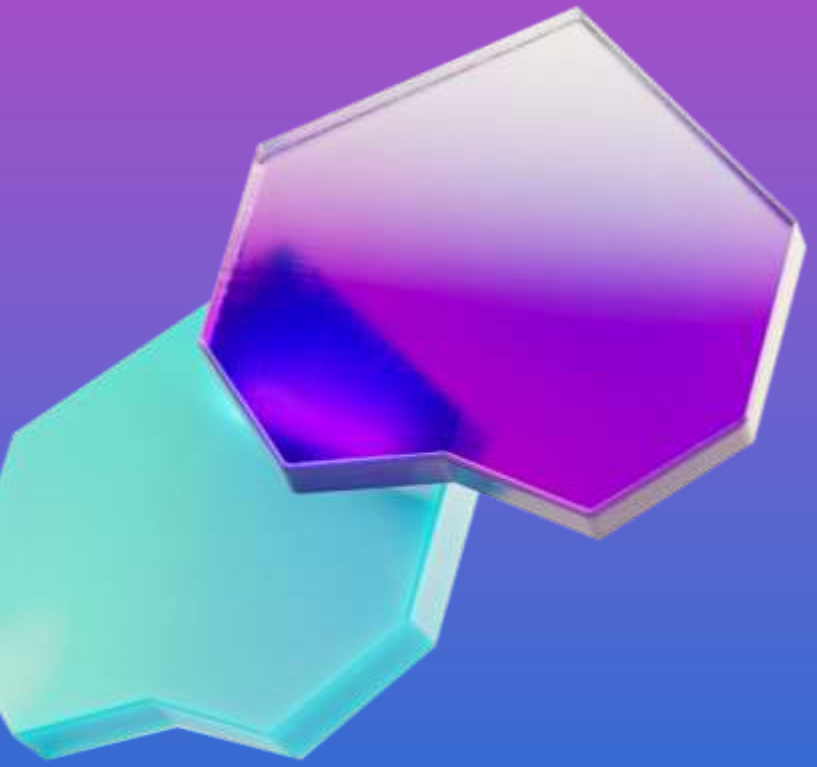
Time to build together!

Lab 7 – create a semantic model



Security

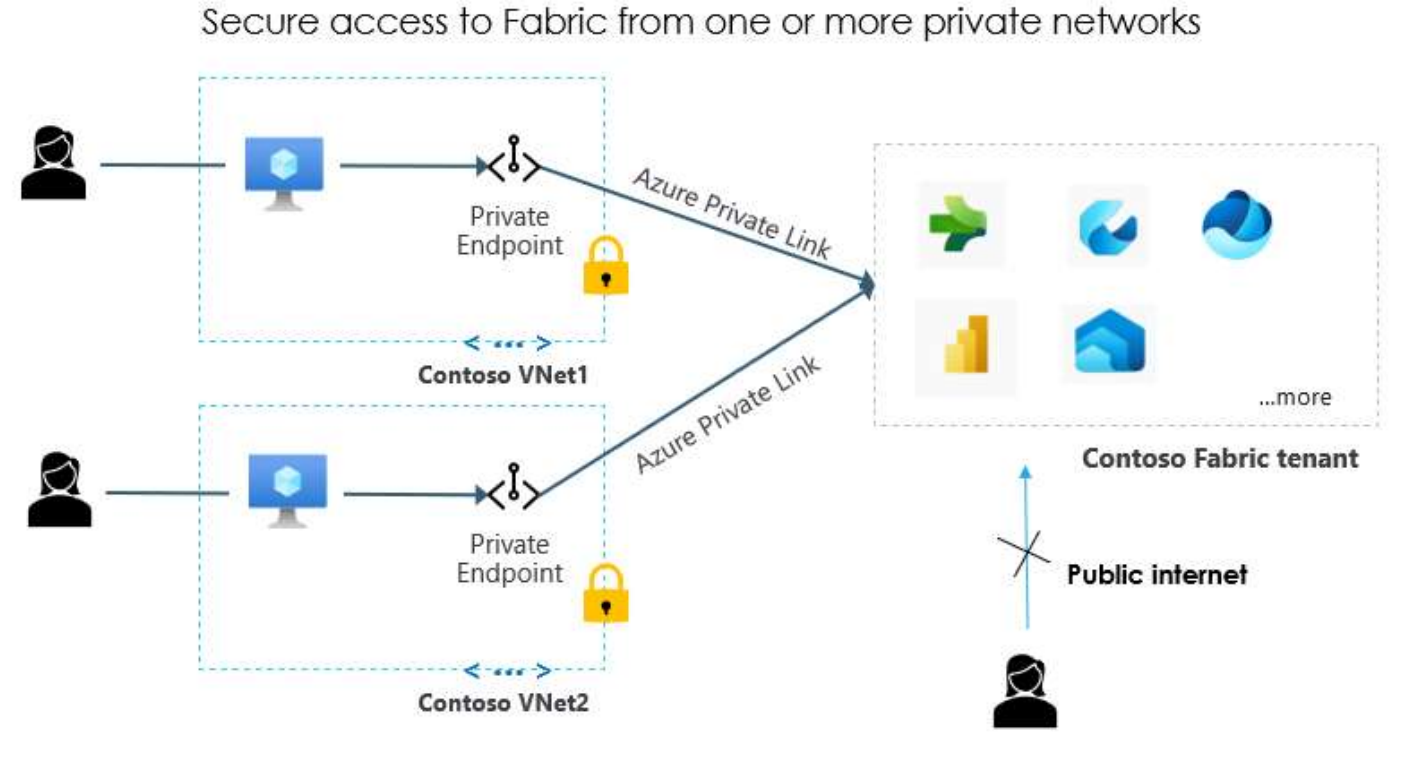
Business Requirements



1. Our company has strict compliance requirements on security and mandates private access
2. We want our developers to be able to have full access rights in our workspace
3. We want our data product owners to have Full Read access rights
4. Business should only have access to trusted data products
5. Our data citizens need access to the semantic models for data exploration
6. Sensitive information in our semantic models is not shared with everyone
7. The managers are only interested in their departments when consuming reports

1 Our company has strict compliance requirements on security and mandates private access

→ Private Link tenant level



2 3 4 5 6 7

1 Our company has strict compliance requirements on security and mandates private access

2

3

4

5

6

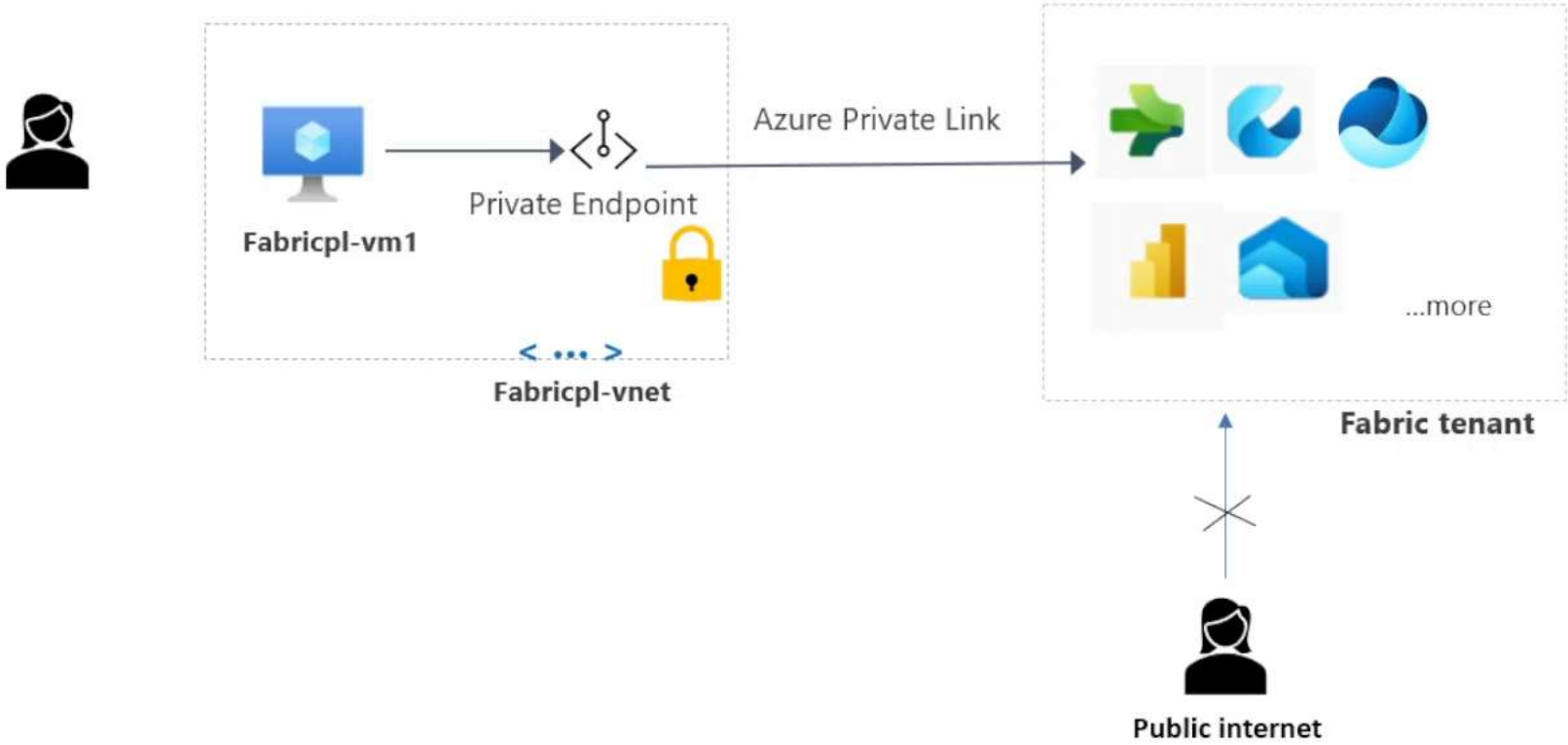
7

DEMO

→ Private Link tenant level

Tenant level Private Link for accessing Fabric Demo

(Warehouse via SSMS and Onelake access via Storage explorer as example)



2 We want our developers to be able to have full access rights in our workspace

1

3 We want our data product owners to have viewer rights

→ Workspace Access



Admin
Control Permission
KILL, MONITOR, SESSIONS, Reshare



Member
Control Permission, Reshare warehouse



Contributor
Control Permission



Viewer
Connect & db_datareader

4

5

6

7

4 Business should only have access to trusted data products

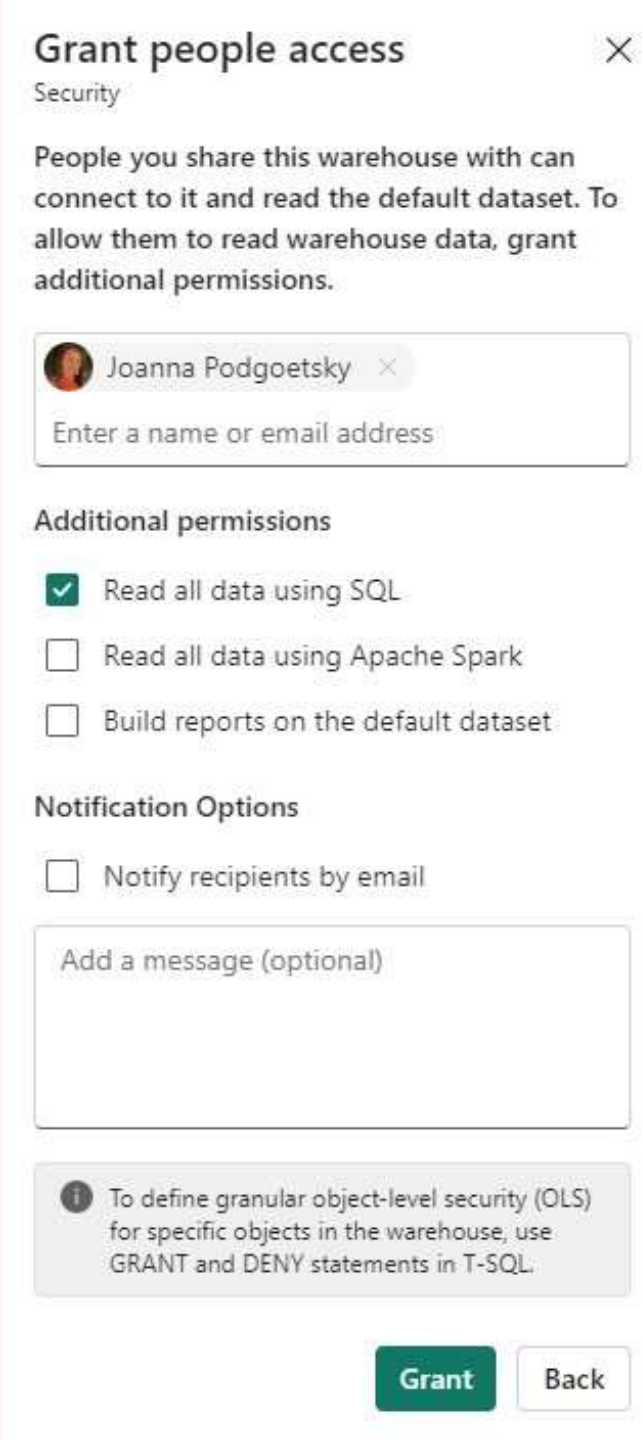
→ Item permission

Read: Connect permission

ReadData: db_datareader

ReadAll: Access to Onelake files

Write: Control permission (Only through Workspace AMC)



1 2 3 5 6 7

4 Business should only have access to trusted data products

1

2

3

DEMO

→ Item permission

5

6

7

5 Our data citizens need access to the semantic models for data exploration

1

2

3

DEMO

→ Item permission

4

6

7

6 Sensitive information in our semantic models is not shared with everyone

→ **Dynamic data masking**

1
2
3
4
5
7

7 The managers are only interested in their departments when consuming reports

→ Row-level security

Filter rows based on user's rights

Security Policies & Table Valued Functions

No Application changes

Even for admins!

1

2

3

4

5

6

7 The managers are only interested in their departments when consuming reports

→ **Row-level security**

→ **Column-level security**

Select permissions on column level

Blocking star queries

No Application changes

Embedded in GRANT no longer a need for views!

1 2 3 4 5 6

1. Mandates private access – **Private link**
2. Full access rights in our workspace – **Workspace access**
3. Viewer rights – **Workspace access**
4. Access to trusted data products – **Item access**
5. Access to the semantic models – **Item access**
6. Sensitive information is not shared – **Dynamic data masking**
7. Selective rights – **Column-level & Row-level security**

Comparing Fabric Data Warehouse and Lakehouse

	Data Warehouse	Lakehouse
Data volume	Unlimited	Unlimited
Type of data	Structured	Unstructured, semi-structured, structured
Primary developer persona	Datawarehouse developer, SQL engineer	Data engineer, Data scientist
Primary developer skill set	SQL	Spark (Scala, PySpark, Spark SQL, R)
Data organized by	Databases, schemas and tables	Folders and files, databases and tables
Read operations	Spark, T-SQL	Spark, T-SQL
Write operations	T-SQL	Spark (Scala, PySpark, Spark SQL, R)
Multi-table transactions	Yes	No
Primary development interface	SQL scripts	Spark notebooks, Spark job definitions
Security	Object, column, row, DDL/DML	Row, table (when using T-SQL), none for Spark
Access data via shortcuts	Yes (indirectly through the lakehouse)	Yes
Can be a source for shortcuts	Yes (tables)	Yes (files and tables)
Query across items	Yes, query across lakehouse and warehouse tables	Yes, query across lakehouse and warehouse tables; query across lakehouses (including shortcuts using Spark)

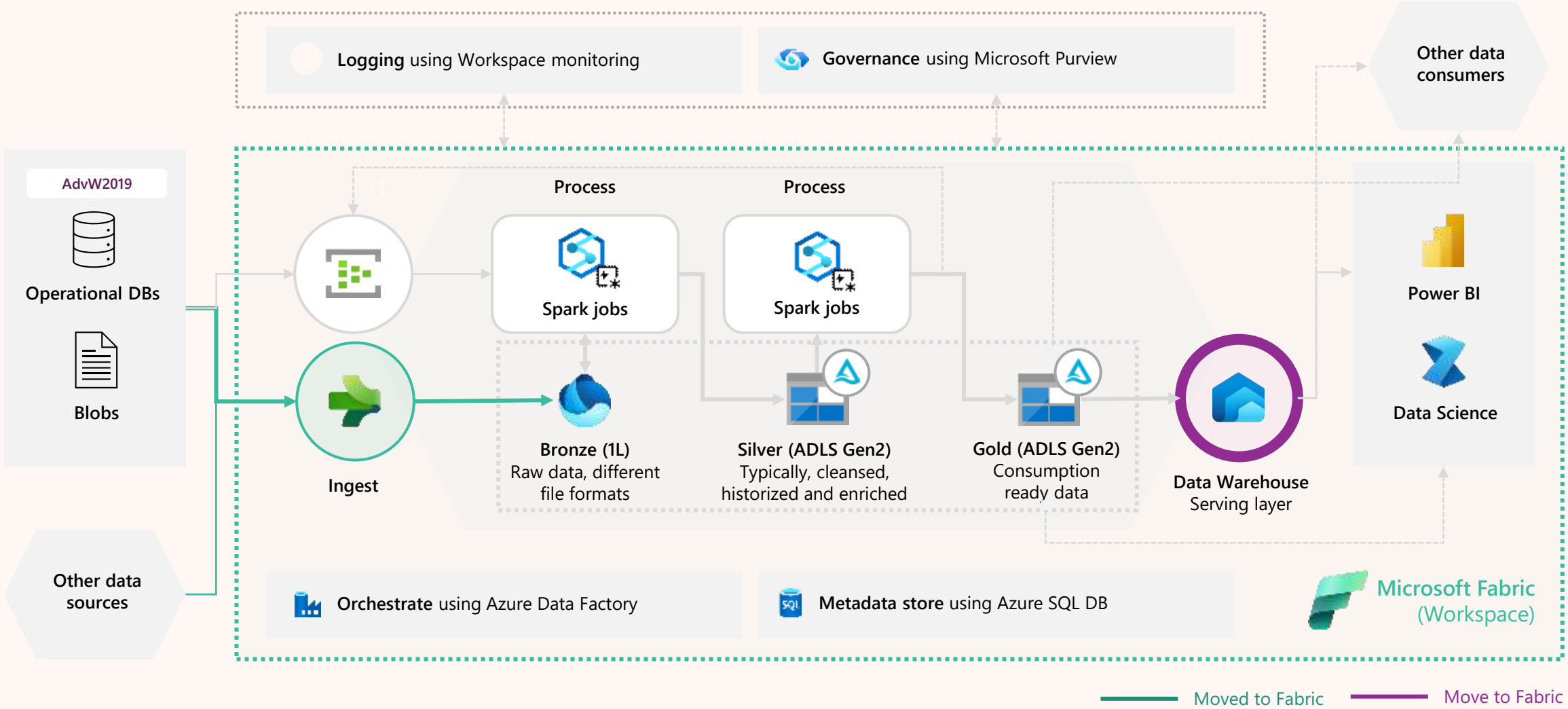
T-SQL surface area

- https://learn.microsoft.com/en-us/fabric/data-warehouse/tsql-surface-area

DW Migration



Customer scenario – moving DW resources



Customer scenario – Preparing the migration



Migration Process Recommendations

Start small and simple

Automate wherever possible

Leverage Azure built-in tools and features to reduce migration effort

Migrate metadata for tables and views

Migrate historical data to be maintained

Migrate or refactor code and business processes

Migrate or refactor ETL/ELT incremental load processes

Migration Approaches

Lift and Shift

- Minimizes risk and migration time
- Good fit for certain scenarios
  - Small number of data marts to migrate
  - Data already in well-designed star or snowflake schema
  - Under time and cost pressure to move to Fabric Warehouse
- Works well for optimized workloads
  - Doesn't require major changes in Fabric

Phased approach for modernization

- Re-engineering to maintain performance levels
- Redesign architecture to utilize new engines and features
- Fabric Workspace offers new opportunities
  - Fabric decision guide - choose a data store - Microsoft Fabric | Microsoft Learn

Migration Considerations

- Only raw data and metadata need to be migrated
- Synapse in Fabric Optimizations
  - Automatic Indexing
  - Automatic Statistics Creation
  - Automatic Statistics Updates – Coming Soon!
  - System Managed Distributions
- There are some limitations on T-SQL support
  - https://learn.microsoft.com/en-us/fabric/data-warehouse/tsql-surface-area

Migrating Synapse DW to Fabric DW

- Options
  - Data Factory
  - Stored Procedures with Accelerated Code
  - Schema Migration Using SQL Database Project
  - Create External Table As
  - Migration via dbt

- **Leverage Partners**

Demo

Migration Accelerator

https://github.com/microsoft/fabric-migration/

DDL Translation

Synapse DW

```
CREATE TABLE [dbo].[magictable]
(
    [col_bigint] [bigint] NOT NULL,
    [col_binary] [binary](1) NULL,
    [col_bit] [bit] NULL,
    [col_char] [char](21) NULL,
    [col_date] [date] NULL,
    [col_datetime] [datetime] NULL,
    [col_datetime2] [datetime2](7) NULL,
    [col_datetimeoffset] [datetime2](7) NULL,
    [col_decimal] [decimal](18, 7) NULL,
    [col_float] [float] NOT NULL,
    [col_int] [int] NOT NULL UNIQUE NONCLUSTERED NOT ENFORCED,
    [col_money] [decimal](19, 4) NULL,
    [col_nchar] [nchar](21) NOT NULL,
    [col_numeric] [decimal](11, 7) NULL,
    [col_nvarchar] [nvarchar](21) NOT NULL,
    [col_real] [real] NULL,
    [col_smalldatetime] [datetime2](4) NULL,
    [col_smallint] [smallint] NULL,
    [col_smallmoney] [decimal](10, 4) NULL,
    [col_time] [time] NULL,
    [col_tinyint] [smallint] NULL,
    [col_uniqueidentifier] [uniqueidentifier] NULL,
    [col_varbinary] [varbinary](1) NULL,
    [col_varchar] [varchar](21) NULL,
    [col_varcharmax] [nvarchar](max) NULL
)
WITH ( DISTRIBUTION = REPLICATE, CLUSTERED INDEX([col_bigint] ASC) )
```

Fabric DW

```
CREATE TABLE [dbo].[magictable]
(
    [col_bigint] BIGINT NOT NULL
    ,[col_binary] VARBINARY NULL
    ,[col_bit] BIT NULL
    ,[col_char] CHAR(21) NULL
    ,[col_date] DATE NULL
    ,[col_datetime] DATETIME2(6) NULL
    ,[col_datetime2] DATETIME2(6) NULL
    ,[col_datetimeoffset] DATETIME2(6) NULL
    ,[col_decimal] DECIMAL(18, 7) NULL
    ,[col_float] FLOAT NOT NULL
    ,[col_int] INT NOT NULL /*UNIQUE NOT ENFORCED*/
    ,[col_money] DECIMAL(19, 4) NULL
    ,[col_nchar] CHAR(21) NOT NULL
    ,[col_numeric] DECIMAL(11, 7) NULL
    ,[col_nvarchar] VARCHAR(21) NOT NULL
    ,[col_real] REAL NULL
    ,[col_smalldatetime] DATETIME2(4) NULL
    ,[col_smallint] SMALLINT NULL
    ,[col_smallmoney] DECIMAL(10, 4) NULL
    ,[col_time] TIME(6) NULL
    ,[col_tinyint] SMALLINT NULL
    ,[col_uniqueidentifier] UNIQUEIDENTIFIER NULL
    ,[col_varbinary] VARBINARY NULL
    ,[col_varchar] VARCHAR(21) NULL
    ,[col_varcharmax] VARCHAR(8000) NULL
)
```

Prepare the Data Export

Accelerator Scripts helps out with the process

```
--Create Maester Key and Scoped Credential
exec SynapseMig.create_master_key_and_scoped_credential

--Create External File Format
exec SynapseMig.create_external_file_format

--Create External Data Source
exec SynapseMig.sp_create_external_data_source
     @adls_base_location = 'abfss://demo@DemoStorage.dfs.core.windows.net/data/',
     @credential_name ='fabric_migration_credential',
     @external_data_source_name='fabric_data_migration_ext_data_source'
```

Data Export Scripts

- Accelerator Scripts helps out with the process

```
--Create CETAS Scripts
exec SynapseMig.sp_cetas_extract_script
 @adls_gen2_location = 'abfss://demo@DemoStorage.dfs.core.windows.net/data/',
 @storage_access_token = 'Demo_AccessToken'
```

SchName	objName	DropStatement	data_extract_statement
dbo	magictable	IF (object_id('db…	CREATE EXTERNAL TABLE [dbo].[migration_magictable] WITH (LOCATION = '/dbo/magicta…
gold	DimTitleAkas	IF (object_id('go…	CREATE EXTERNAL TABLE [gold].[migration_DimTitleAkas] WITH (LOCATION = '/gold/Dim…
gold	Dimtitle	IF (object_id('go…	CREATE EXTERNAL TABLE [gold].[migration_Dimtitle] WITH (LOCATION = '/gold/Dimtitl…
gold	DimName_basics	IF (object_id('go…	CREATE EXTERNAL TABLE [gold].[migration_DimName_basics] WITH (LOCATION = '/gold/D…
gold	FactTitleEpisode	IF (object_id('go…	CREATE EXTERNAL TABLE [gold].[migration_FactTitleEpisode] WITH (LOCATION = '/gold…
gold	FactTitlePrincipals	IF (object_id('go…	CREATE EXTERNAL TABLE [gold].[migration_FactTitlePrincipals] WITH (LOCATION = '/g…
gold	FactTitleRatings	IF (object_id('go…	CREATE EXTERNAL TABLE [gold].[migration_FactTitleRatings] WITH (LOCATION = '/gold…

Data Load Scripts for Fabric DW

- Accelerator Scripts helps out with the process

```
--Create Loading Scripts for Fabric
exec SynapseMig.generate_data_extract_and_data_load_statements
      @storage_access_token = 'Demo_AccessToken'
    , @external_data_source_base_location = 'Demo_DataSource'
```

SchName	objName	data_load_statement
dbo	magictable	COPY INTO [dbo].[magictable] FROM 'Demo_DataSourcedbo/magictable/' WITH (FILE_TYPE = 'PARQUET', CREDENTIAL=(ID…
gold	DimTitleAkas	COPY INTO [gold].[DimTitleAkas] FROM 'Demo_DataSourcegold/DimTitleAkas/' WITH (FILE_TYPE = 'PARQUET', CREDENTI…
gold	Dimtitle	COPY INTO [gold].[Dimtitle] FROM 'Demo_DataSourcegold/Dimtitle/' WITH (FILE_TYPE = 'PARQUET', CREDENTIAL=(IDEN…
gold	DimName_basics	COPY INTO [gold].[DimName_basics] FROM 'Demo_DataSourcegold/DimName_basics/' WITH (FILE_TYPE = 'PARQUET', CRED…
gold	FactTitleEpisode	COPY INTO [gold].[FactTitleEpisode] FROM 'Demo_DataSourcegold/FactTitleEpisode/' WITH (FILE_TYPE = 'PARQUET', …
gold	FactTitlePrincipals	COPY INTO [gold].[FactTitlePrincipals] FROM 'Demo_DataSourcegold/FactTitlePrincipals/' WITH (FILE_TYPE = 'PARQ…

DML Assessment

- Accelerator Scripts helps out with the process

```
--Check for unsuported code
exec SynapseMig.Check_UnsupportedDML
```

name	name	value	ExprName	ExprCode
dbo	SynapseMigration_ExtractAllDDL	↵-- ntext varchar(1..8000)	NText Data Type	NTEXT
dbo	SynapseMigration_ExtractAllDDL	↵-- text varchar(1..8000)	Text Data Type	TEXT
SynapseMig	Check_UnsupportedDML	↵ SELECT 'Check Sum Agg Function',' CHECKSUM_AGG 'UNIO…	Check Sum Agg Function	CHECKSUM_AGG
SynapseMig	Check_UnsupportedDML	↵ SELECT 'Choose Function',' CHOOSE 'UNION ALL	Choose Function	CHOOSE
SynapseMig	Check_UnsupportedDML	↵ SELECT 'Continue Statement',' CONTINUE'UNION ALL	Continue Statement	CONTINUE

Demo

Synapse Gen2 Schema and Data Migration to Fabric DW

aka.ms/FabricRoadmap

- What's new and planned for Synapse Data Warehouse in Microsoft Fabric - Microsoft Fabric | Microsoft Learn

Snowflake and Databricks integration – Forward looking



Do more with your data with open standards

Iceberg support in OneLake

Bi-directional data access

Seamless access from M365 and Copilot

Expanded Partnership



Microsoft Fabric

+



Snowflake

Better together

Open and governed



Azure Databricks



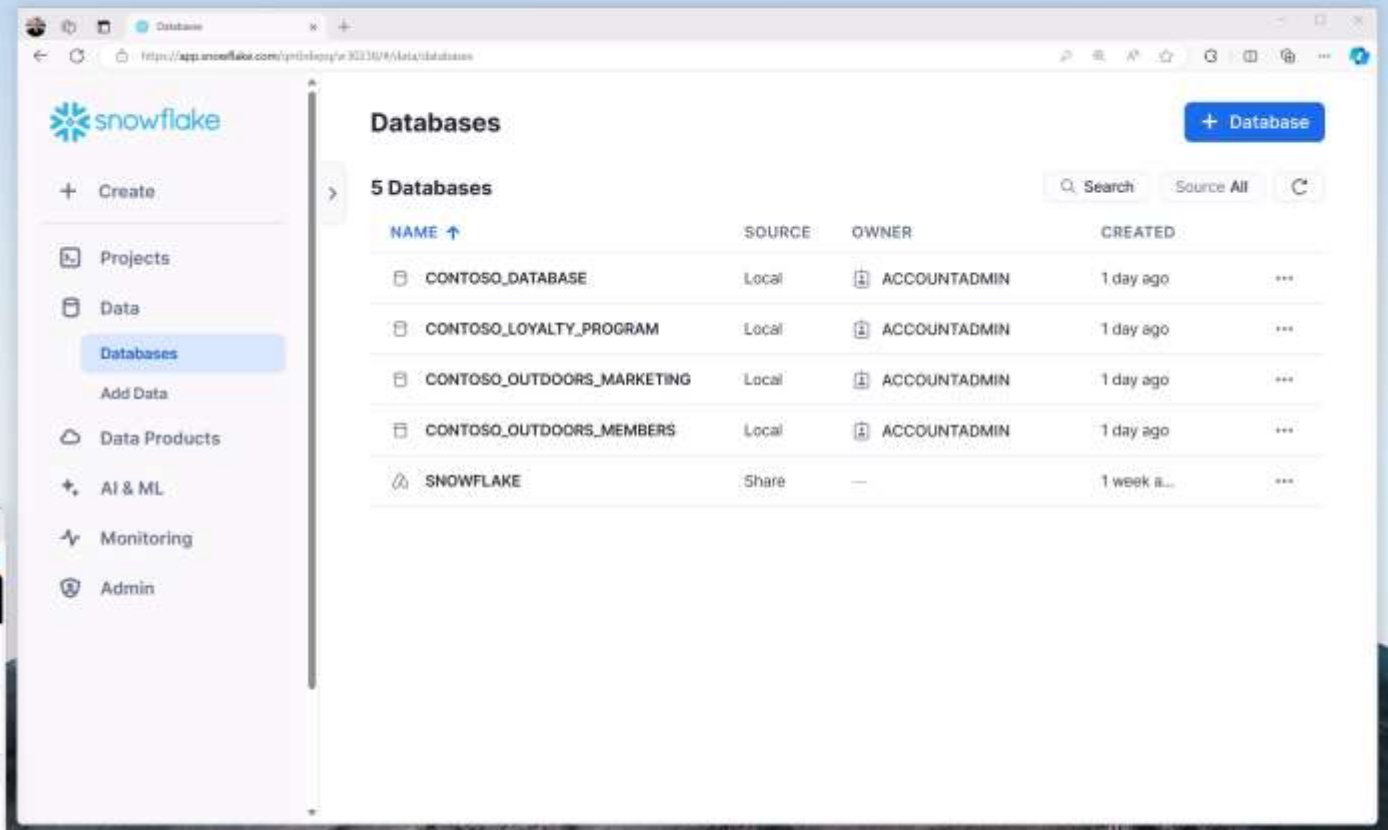
Microsoft Fabric

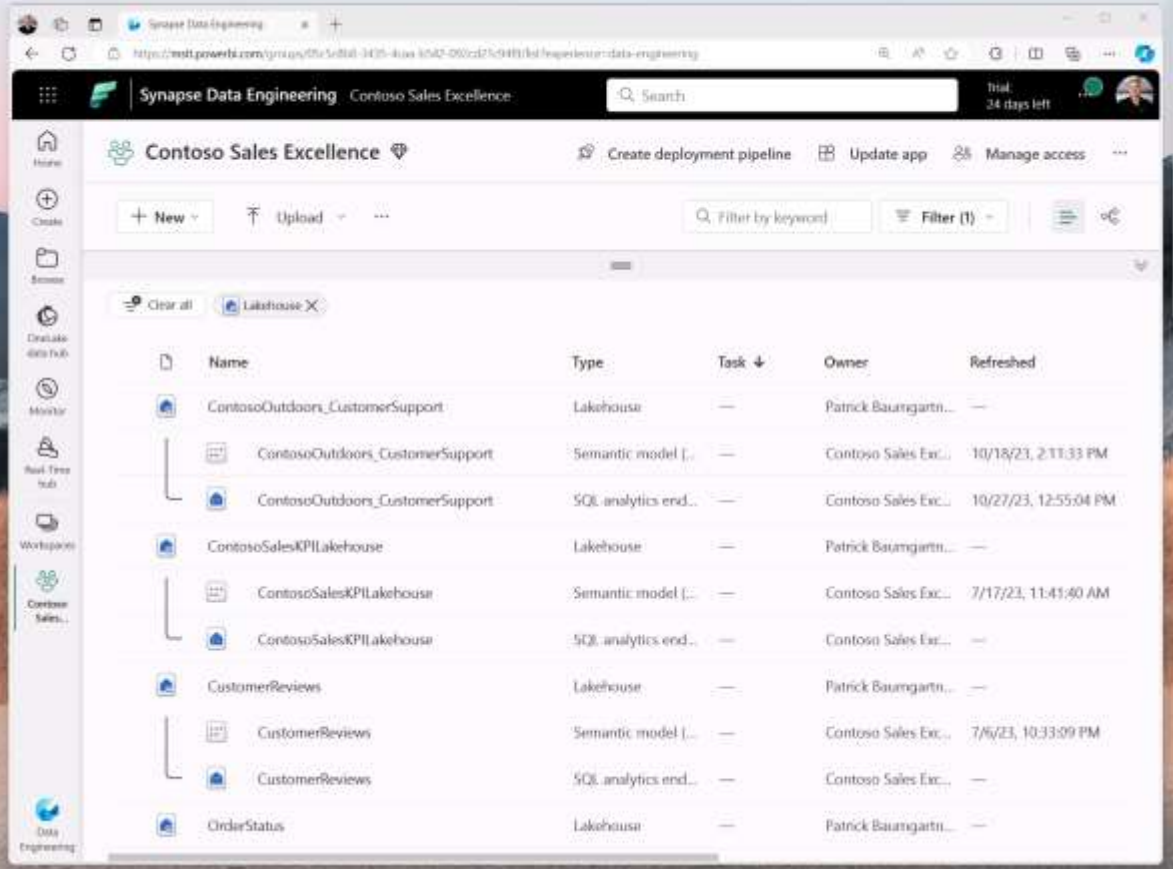


Snowflake



OneLake


snowflake
Create
Projects
Data
Databases
Add Data
Data Products
AI & ML
Monitoring
Admin
Databases
+ Database
5 Databases
Search
Source All
NAME
SOURCE
OWNER
CREATED
CONTOSO_DATABASE Local ACCOUNTADMIN 1 day ago
CONTOSO_LOYALTY_PROGRAM Local ACCOUNTADMIN 1 day ago
CONTOSO_OUTDOORS_MARKETING Local ACCOUNTADMIN 1 day ago
CONTOSO_OUTDOORS_MEMBERS Local ACCOUNTADMIN 1 day ago
SNOWFLAKE Share — 1 week a...


Synapse Data Engineering
Contoso Sales Excellence
Search
Trial: 24 days left
Contoso Sales Excellence
Create deployment pipeline
Update app
Manage access
New
Upload
Filter by keyword
Filter (1)
Clear all
Lakehouse
Name
Type
Task
Owner
Refreshed
ContosoOutdoors_CustomerSupport Lakehouse Patrick Baumgartn...
ContosoOutdoors_CustomerSupport Semantic model (... Contoso Sales Exc... 10/18/23, 2:11:33 PM
ContosoOutdoors_CustomerSupport SQL analytics end... Contoso Sales Exc... 10/27/23, 12:55:04 PM
ContosoSalesKPILakehouse Lakehouse Patrick Baumgartn...
ContosoSalesKPILakehouse Semantic model (... Contoso Sales Exc... 7/17/23, 11:41:40 AM
ContosoSalesKPILakehouse SQL analytics end... Contoso Sales Exc...
CustomerReviews Lakehouse Patrick Baumgartn...
CustomerReviews Semantic model (... Contoso Sales Exc... 7/6/23, 10:33:09 PM
CustomerReviews SQL analytics end... Contoso Sales Exc...
OrderStatus Lakehouse Patrick Baumgartn...

Evaluation

