› Home › › Azure Data › › Azure Synapse Analytics Blog
› › Data mesh: A perspective on using Azure Synapse Analytics to build data products

Back to Blog | ‹ Newer Article | Older Article ›

Data mesh: A perspective on using Azure Synapse Analytics to build data products

By Amanjeet Singh

Published Nov 03 2022 08:38 AM 6,045 Views

CSE Customer Success Engineering
Azure Synapse Analytics

Author(s): Amanjeet Singh is a Program Manager in Azure Synapse Customer Success Engineering (CSE) team.

Introduction: Data mesh and challenges it addresses

As organisations grow, the volume of data and the number of systems generating data also grows. The size and number of disparate systems introduces procedural, operational and technology challenges which impact (in a negative way) agility, scale, and governance associated with analytics. This is the problem space which data mesh is targeted at. It’s an approach to data architecture which aims at addressing these challenges through decentralization of technology platforms and processes; establishing domain-aligned teams; and governance at scale. It brings product thinking to analytics and calls for a fundamental shift in the assumptions, architecture, technical solutions, and structure of organisations. [1]

Data mesh has four principles [2]:

- Data as a product
- Domain ownership
- Self-serve data platform

Skip to primary navigation
Skip to main content

- Federated computational governance

Before we go any further, it's worth noting that:

1. In a real-world scenario, one would arrive at the technical implementation of data product(s) **after** requirements; design; structure of cross-functional teams; domains; data products; and other key decisions have been finalized. Building a data product shouldn't be the first step for any organization embarking on a data mesh journey.
2. There is no default standard or reference implementation of data mesh [3] and its components. Most data mesh implementations are different and vary depending on requirements.
3. Discussing details of all aspects of data mesh architecture and data management is out of scope for this blog. We recommend books authored by Zhamak Dehghani (Data Mesh: Delivering Data-Driven Value at Scale) and Piethein Strengholt (Data management at scale) if you want to learn more about these topics.

About Azure Synapse Analytics

Azure Synapse is an analytics service that brings together enterprise data warehousing and Big Data analytics. Azure Synapse brings together the best of SQL technologies used in enterprise data warehousing, Spark technologies used for big data, Data Explorer for log and time series analytics, Pipelines for data integration and ETL/ELT, and deep integration with other Azure services such as Power BI, Cosmos DB, Azure ML etc. which are part of Microsoft Intelligent Data Platform. A conceptual view of Azure Synapse ecosystem is shown in figure 1.

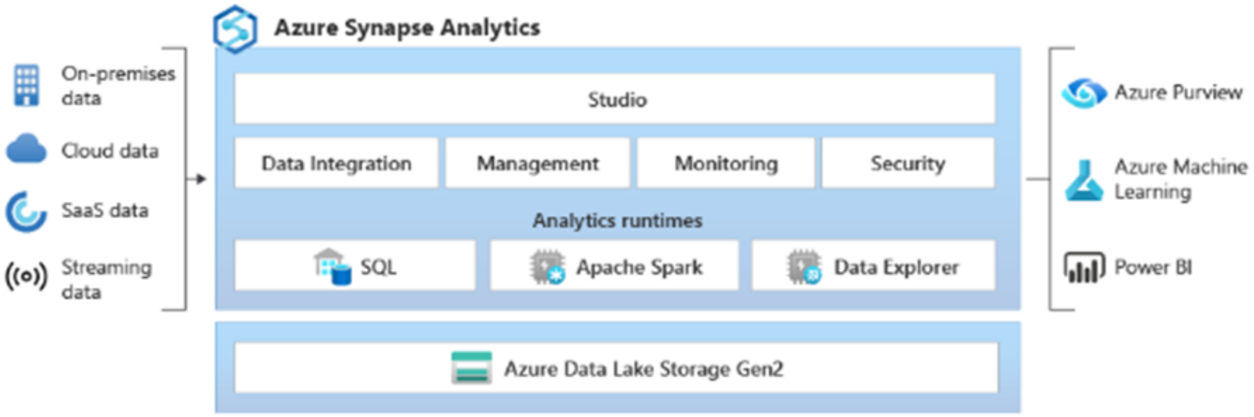


Figure 1 – Overview of Azure Synapse Analytics

What is a data product and where does Azure Synapse Analytics fit-in?

A data product is one of the core building blocks of data mesh architecture. It encompasses the domain-specific data; pipelines to load and transform the data; and interfaces to share the data with other domains. In simple terms, a data product consists of data and the underlying software and infrastructure required to ingest, store, and share data with other domains.

There are three main functions [4] of a data product:

- Consume data
- Transform data
- Serve data

Figure 2 shows alignment of Synapse Analytics features to the functions of a data product.

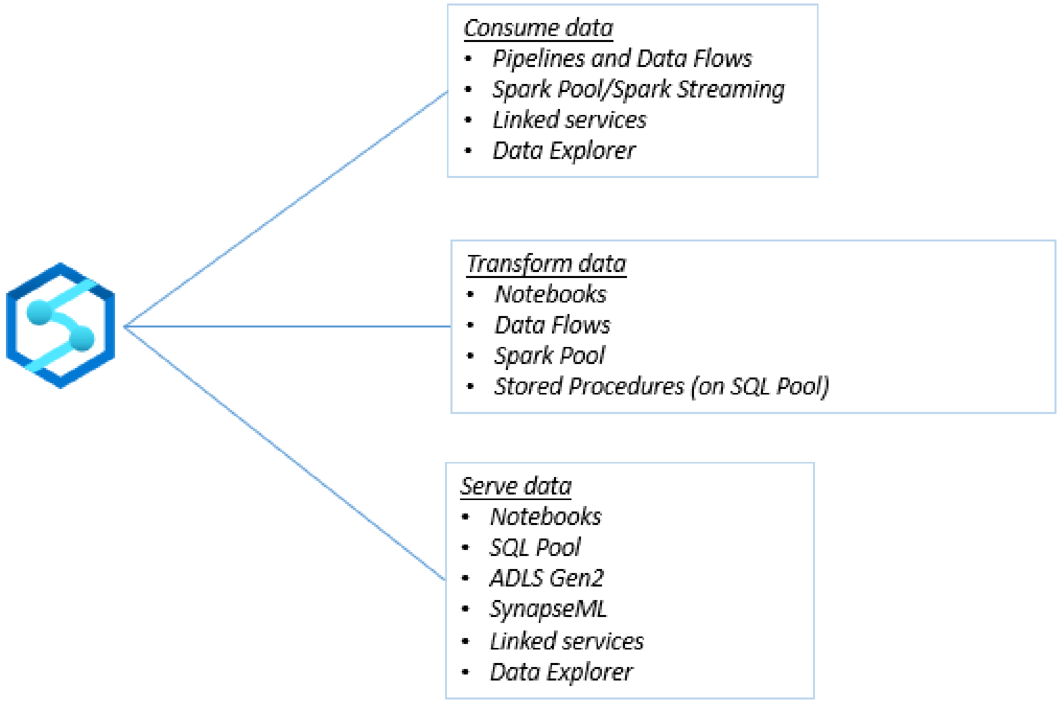


Figure 2 – Data product functions and Azure Synapse Analytics feature alignment

Now that we understand the composition of a data product, we shall now walk through our perspective on how Azure Synapse Analytics features align to its functions.

Function	Properties [5]	Synapse Ana
Consume data	A data product has one or more input data ports or a mechanism to connect and source data from various sources including other data products.	Azure Synaps which enable various first-p such as SAP, ship with Pip Additionally, extensible an libraries to co do not ship w Through inte Microsoft Int options for ir Explorer poo

SkSkip to primary navigation main content

Transform data	All data products perform data transformation and within data mesh architecture, transformation is an internal implementation of data product.	On Azure Syr Pipelines, Da procedures e complex tran products.
Serve data	A data product may present its data in multimodal form such as columnar files, relational tables, graph, or events.	Through a cc capabilities a polyglot pers can be serve mechanisms **File access –** served from Gen2. **SQL access -** relations whi such as Powe use SQL end data to dowr **Events** – Azu Capture featu to ADLS Gen Events writte processed ar domains dov Additionally, Organization place reads (physically shi domains.

	Immutability	With Pipeline developers c ingestion pat data unchang Immutability file formats v Parquet. Parc Serverless, Sy To summarise implementec immutable o are append c
	Bitemporal data	Bitemporal d so that every timestamps " With Azure S features, user to the data s add these att written out to

Let's now look at primary capabilities of a data product and where Azure Synapse Analytics features fit-in. A data product must have the following capabilities:

- Storage
- Data movement
- Data serving
- Transformation
- Governance

The following table outlines data product capabilities and Azure Synapse Analytics feature alignment.

Data Product capability	**Azure Synapse Analytics Features**
Data storage	Synapse Dedicated SQL pools Azure Data Lake Storage Gen2 Synapse Data Explorer
Data movement	Synapse Pipelines and Dataflow Spark Synapse Link for Cosmos DB, SQL, Da

Skip to primary content Skip to main navigation

Data serving	Synapse Dedicated SQL Pool Serverless SQL Pool Data Explorer Power BI Notebooks
Transformations	Pipelines and Data Flows Synapse Spark Transact-SQL or CLR stored procedur
Governance	Platform - Azure Policy Data - Microsoft Purview

To summarize, by using a combination of one or more Azure Synapse Analytics features and its native integrations with the wider Microsoft Intelligent Platform, individual domains can build a gamut of rich data products.

Strategies for building data products using Azure Synapse

It's important to understand Synapse Analytics hierarchy model as it has an implication on scale and domain access control within data mesh architecture. Figure 3 below shows key components of a workspace and their relationship with each other.

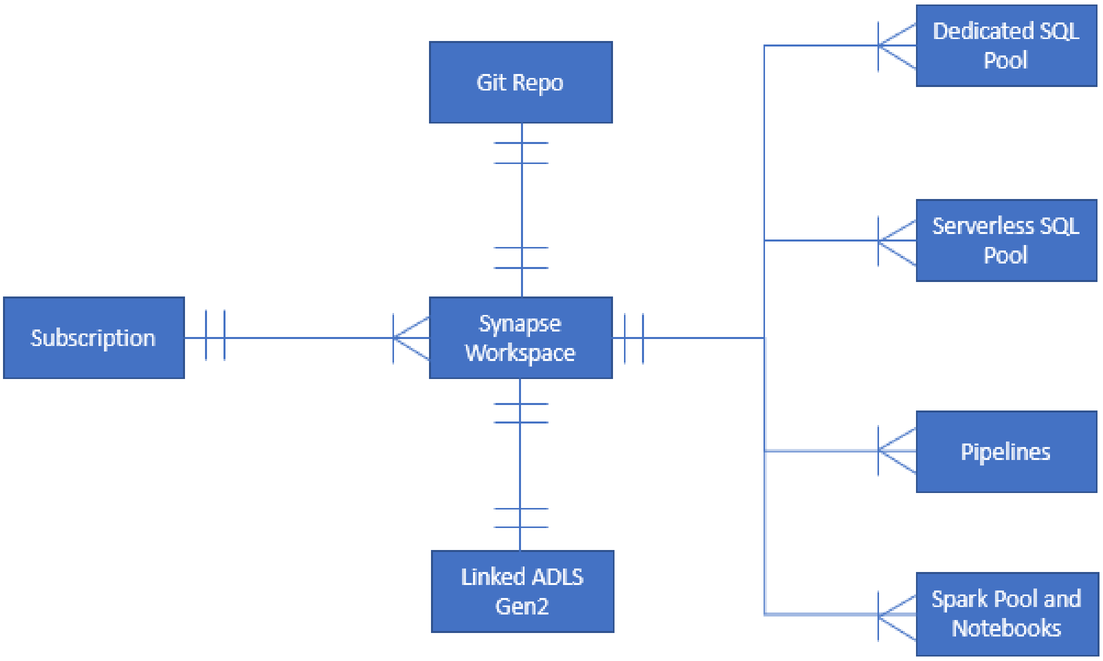


Figure 3 – Azure Synapse Analytics hierarchy

Skip to primary navigation Skip to main content

1. An Azure Subscription can have one or more Azure Synapse Analytics workspaces. A single Subscription has limits and subscription-scoped management characteristics. A Synapse Workspace has limits and workspace-scoped management characteristics. Hence, number of subscriptions and workspaces has an impact on scale and how management boundary is setup between domains which we will cover in the next section.
2. A Synapse Workspace can only be linked to a single version control repo. Any code associated with a workspace must be stored in this repo. If teams share a workspace, by extension they share the version control repo. Thus, a single Workspace is a unit of collaboration.
3. A Synapse Workspace is linked to a primary ADLS Gen2 account which is used for storing job output and logs. This is not to be confused with number of ADLS Gen2 storage endpoints a Workspace can read data from or write data to. Linked storage's primary role is to store logs and temporary data. The linked ADLS Gen2 account can also be used to store user data.
4. A workspace may have zero or more of artifacts such as SQL Pool, Spark Pool, Pipelines etc. The figure above shows a few artifacts, however there is a one-to-many relationship between a single workspace and underlying artifacts. Limits associated with a workspace are published here. This has an implication on scale and management boundary.

The relationship between a Subscription, Synapse workspace and underlying Synapse artifacts influences scale and access control a domain has within each deployment pattern discussed below. **Scale boundary** refers to how large a single instance of a Synapse artifact or Synapse Workspace can scale. **Domain autonomy** refers to access control which members of a domain have within a Subscription, a Synapse workspace, and a single instance to build data products. This will influence agility of a domain and on Azure, this is controlled by Azure RBAC and resource-level permissions (on control and data planes).

The table below summarizes various patterns for deploying Azure subscription(s) and Synapse workspace(s) as a data product. A Synapse artifact refers to individual features of Synapse such as SQL Pool, Spark Pool, Data Explorer, Pipelines etc.

Pattern	**Deployment Pattern**			**Scale boundary**
	Azure Subscription	**Azure Synapse Analytics Workspace**	**Synapse Artifact (SQL Pool, Spark Pool etc.)**	

Pattern# 1 ***Refer Figure 4***, domains share a single instance of a Synapse artifact (SQL Pool, Spark Pool etc.) within a single workspace**.**	Single subscription	Single workspace	Single instance of a Synapse artifact shared by all domains, i.e., a single instance of SQL Pool services all relational use-cases for all the domains.	Scale limits for a sir Azure Subscription apply here. Scale limit for a sing Synapse workspace inherited by artifact such as pools etc. a apply here. Scale limit for a sing instance of an artifa such as SQL Pool, Spark Pool etc. app to this pattern.
Pattern# 2 ***Refer Figure 5***, each domain has a dedicated Synapse artifact within a workspace.	Single subscription	Single workspace	Domain aligned artifacts such as SQL Pool, Spark Pool, Pipelines etc.	Since each domain a dedicated instanc of Synapse artifact, scale limit is dictate by a single instance a Synapse artifact.

Skip to primary content Skip to navigation

Pattern# 3 ***Refer figures 6A and 6B***, single subscription with multiple workspaces. Domains consolidated across multiple workspaces.	Single subscription	Multiple workspaces	Multiple instances of Synapse artifacts deployed since there are multiple workspaces.	Scale limits for a sir Azure Subscription apply here. Individual Synapse artifacts can scale within bounds of a workspace.
Pattern# 4 ***Refer figure 7***, Single subscription with a dedicated Synapse workspace for each domain.	Single subscription	Separate workspaces for each domain.	Each domain gets a dedicated workspace and associated Synapse artifacts.	Scale limits for a single Azure Subscription apply here. Scale enabled throu multiple workspace and artifacts.
Pattern# 5 ***Refer figure 8***, Separate subscription with a separate Synapse workspace for each domain.	Separate subscription for each domain	Separate workspaces for each domain	Each domain gets a dedicated workspace and artifacts.	Scale enabled throu separate subscriptic multiple workspace and Synapse artifac

Let’s discuss considerations for each of the deployment patterns discussed above.

Skip to primary navigation Skip to main content

Pattern 1 - A single Azure Subscription with a single Synapse Analytics workspace

In this deployment model, a single Synapse workspace is shared across domains. In this domain multi-tenancy model, analytics pools and other artifacts belonging to a workspace are shared across domains (as shown in the figure below). Essentially, each domain gets a slice of resources and privileges to ship data products.

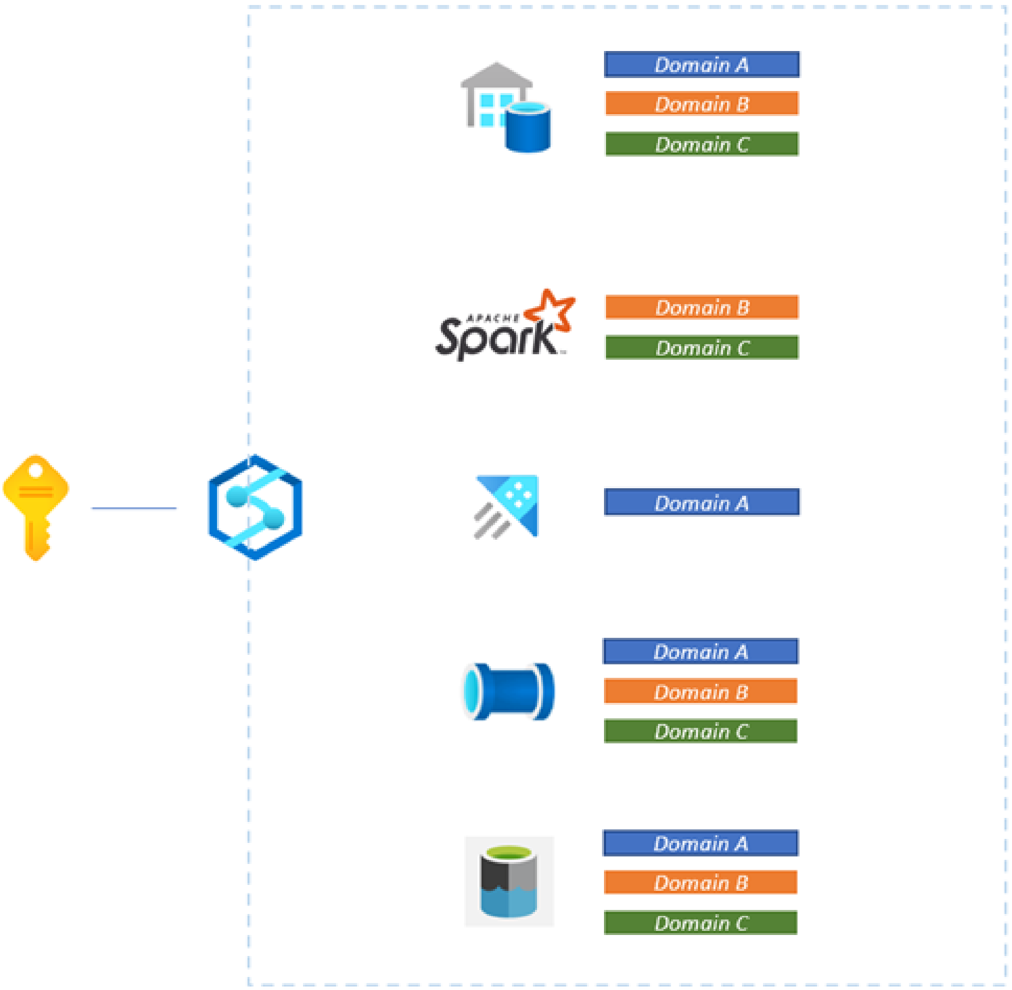


Figure 4 – Subscription and Synapse Workspace multi-tenancy model. A single workspace hosts all domains, and each domain gets a slice of a Synapse artifact.

From systems perspective, since there's a single instance of a resource, all domains sharing that single artifact are bound in terms of scale limits, performance targets, recovery targets etc. Essentially, we are treating all domains the same from perspective of scale, performance, recovery targets, maintenance etc.

Following considerations apply to this deployment pattern:

- Subscription-scoped limits, policies and Management Group membership applies.
- Azure Synapse Analytics workspace-scoped limits and RBAC applies. Since, there's a single workspace and a single instance of each of Synapse artifact, a single team assumes ownership/responsibilities of administering the workspace and supporting underlying analytics platform.
- Each domain has a slice of resources belonging to a workspace (pools, pipelines etc.).

Skip to primary navigation Skip to main content

- Scale limits associated with a single instance of a Synapse artifact applies. Example – if a single SQL Pool is shared across multiple domains, at any given point, no more than 128 queries can be processed concurrently as that's the upper limit. If you plan to run more than 128 queries at any given point in time, we recommend exploring one of the subsequent deployment patterns.
- Single instance of workspace means that all Synapse artifacts belong to a single region.
- Since a workspace can only be linked to a single source control repo it means that domain teams share a single repo. Whilst the repo is shared, what makes into production or gets merged to master branch will be controlled by access control configured at various levels (pipelines, managed identities, source connections etc.). A common strategy is where teams can create a feature branch from collaboration branch when they are working on new feature or release of the data products. When the changes are ready, they can create pull request to merge their changes from feature branch to collaboration branch.
- Since all domains share a workspace, there is no ability to restrict access for development artifacts like Synapse Pipelines, notebooks, etc. between different domain users.
- Concerns associated with a single repo must be considered such as branching strategy, repo limits etc.
- Workload management plays a critical role within this deployment and generally, requires fine tuning to ensure that each domain has adequate amount of resources (SQL Pool) to accomplish tasks. Generally, fine tuning WLM is required to guarantee QoS to domain teams.
- Encryption with customer managed key (CMK) is configured at workspace level. If there's a requirement for domains to use different CMK, then this model wouldn't work.
- Collations are applicable to an instance of SQL Pools. If domains require different collation, then separate SQL Pools must be provisioned as its defined per database.
- Chargeback model is hard to implement with a single shared workspace and shared artifacts such as pools, pipelines etc. The reason is that billing is based on per Synapse artifact basis.
- A multi-tenanted Synapse Spark pool is composed of a single type of compute size and node family. As such, if domains have different footprint and workload sizes, then it might not be efficient to adopt "one size fits all" approach to all domains. This might also have a cost implication on the cost of a single Spark pool.

When may organisations opt for this deployment pattern?

- Peak period resource utilization does not overlap between domains and scale limit of a single instance of an artifact satisfies performance requirements.
- When there is a significant overlap in terms of datasets used across domains, hence an organisation may choose to group services together to reduce duplication of data and services.
- Upstream systems and applications are common to domains and hence, organizations may choose to extend that model to data products belonging to those domains.
- The teams responsible for supporting and management of the underlying platform and services is the same or perhaps there are not enough headcount to manage platform; hence, from ops and management perspective, it may make sense to use a single Synapse workspace where a single platform team provides support and management.

Pattern 2 - A single Azure Subscription with a single Synapse workspace with dedicated Synapse artifacts for each domain

This model of deployment is like pattern 1, however organizations deploy separate instances of Synapse artifacts aligned to various domains. Example – dedicated SQL Pool for domain A; dedicated Spark Pool for domain B and so on and so forth within a single shared Synapse Analytic workspace.

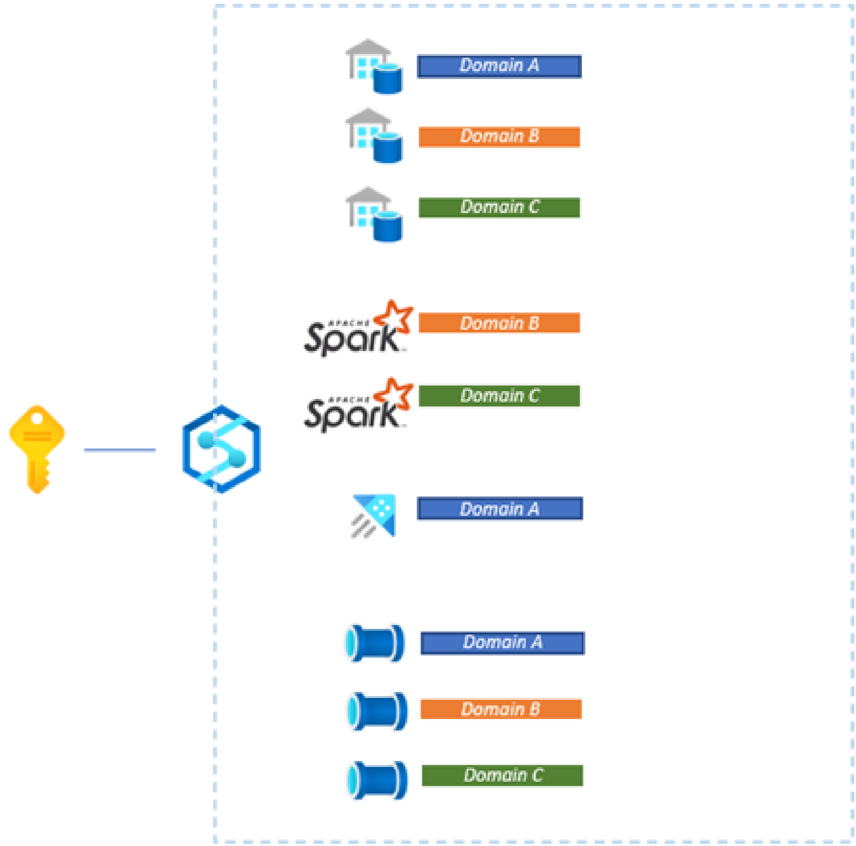


Figure 5 – Each domain has a separate instance of Synapse artifact but within bounds of a single workspace.

In this model, following considerations apply:

- Domains can scale within bounds of a single instance of an artifact. Generally, this model offers better scale to domains compared to pattern 1.
- Compared to pattern 1, this model offers better flexibility to organisations in terms of choosing appropriate sizes for Synapse artifacts depending on performance requirements, number of users etc. for a domain.
- Compared to pattern 1, workload management is easier to configure and manage as each domain has access to a dedicated set of resources within a workspace. Risk of noisy neighbour is significantly reduced in this deployment pattern.
- Domains can use different collation for each SQL Pool instance.
- Each domain, depending on performance requirements, can choose to deploy an appropriately sized SQL pool database.
- Flexibility to use different node types for Spark pools for each domain depending on scale and performance requirements.
- CMK remains the same across all domains as they all share the same workspace.
- Region of deployment is determined by the region where a workspace is deployed, and this is inherited by all the resources belonging to a workspace.
- Domains share common version control repo across all domains as they are all sharing workspace.
- Since all domains share a workspace, there is no ability to restrict access for development artifacts like Synapse Pipelines, notebooks, etc. between different domain users.
- Simpler billing model as each domain has a dedicated Synapse artifact. Charging each domain for their share of the usage is simpler.

Skip to primary navigation Skip to main content

- Depending on domain team composition, a domain may control and manage its instances of Synapse artifacts. For example, a domain may have a dedicated SQL pool and they can manage objects (schemas, tables etc.) deployed in the context of this instance.
- Generally, compared to pattern 1, this deployment model requires more data movement between domains as each domain has a dedicated instance of a Synapse dedicated SQL pool. However, this can be reduced significantly through use of Synapse SQL Serverless and/or using strategies where a common data lake is shared across multiple Synapse artifacts. One such a pattern is discussed here in this blog.

When may organisations choose this model?
The pattern has similarities to pattern 1 where single subscription and single workspace scale and management limits apply; however, this model offers larger scale boundary for a domain to operate within. Common reasons to adopt this model include:

- Each domain has different workload requirements and using a single instance of a Synapse artifact may not be large enough to support these performance requirements. Scale could be in terms of consumer base, or significant difference in usage patterns.
- In terms of autonomy, domains have freedom within an artifact to build data products, i.e., a domain which owns and manages a Synapse Dedicated SQL Pool artifact and can create/drop objects within that artifact. It's more agile compared to pattern 1.

Pattern 3 – An Azure Subscription with multiple Synapse workspaces

In this model, an Azure subscription houses one or more Synapse workspaces. The difference here is that workspaces could be used to consolidate a function such as data ingestion, or perhaps consolidate a set of domains based on a criterion such as region of deployment.

Skip to primary navigation Skip to main content

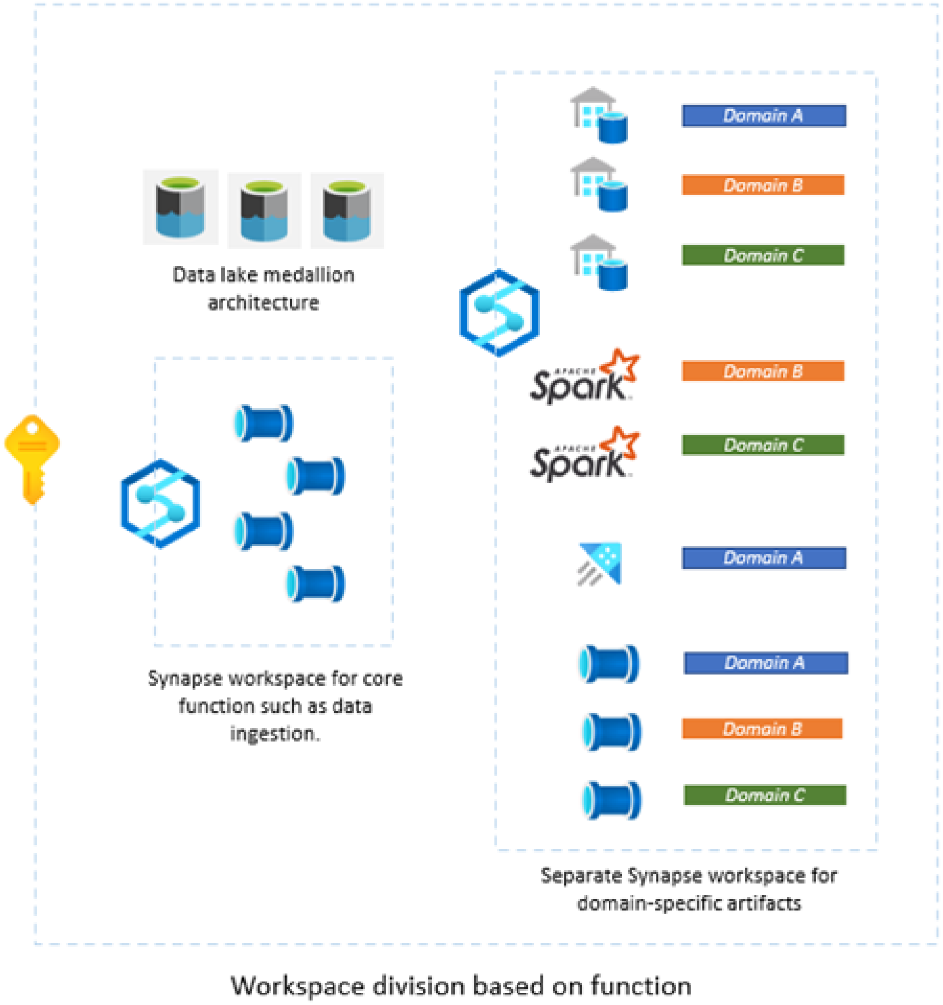
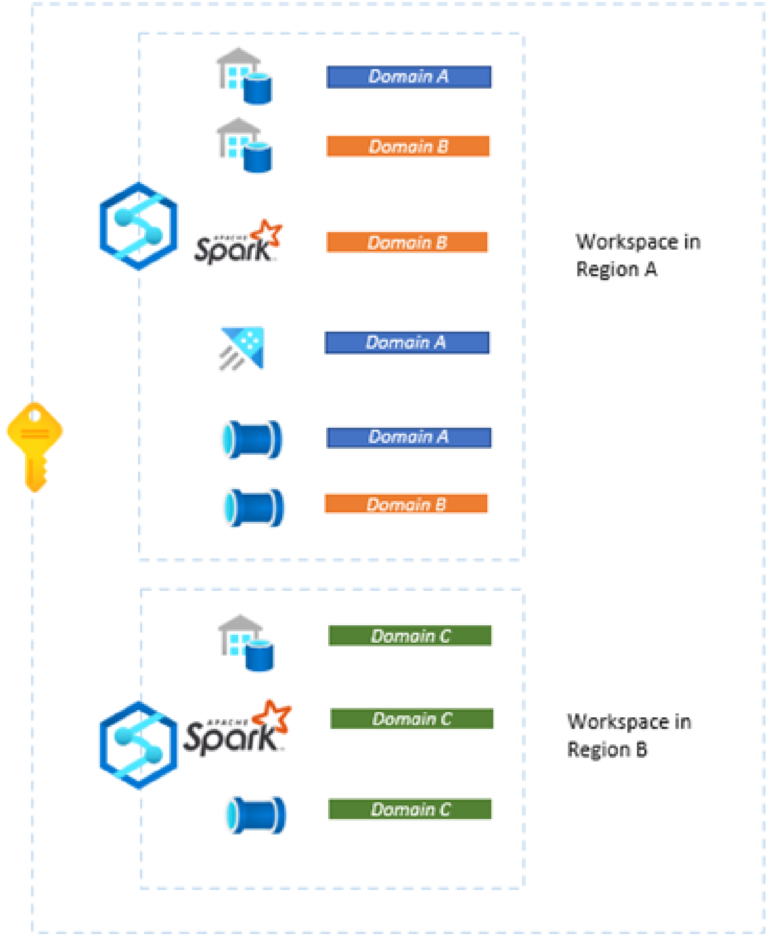

Domain A
Domain B
Domain C
Data lake medallion architecture
Domain B
Domain C
Domain A
Domain A
Domain B
Domain C
Synapse workspace for core function such as data ingestion.
Separate Synapse workspace for domain-specific artifacts
Workspace division based on function

Figure 6A – An example of grouping workspaces based on a function such as data ingestion within a single subscription.

Skip to primary navigation Skip to main content



Workspace division based on grouping of domains. In this example, it's based on regions.

Figure 6B – Deployment pattern where domains are grouped under separate workspaces are due to reasons such as region.

Following considerations apply for this deployment pattern:

- Policy, scale, management group membership scoped at subscription level will flow down to its child artifacts such as Synapse workspace and other services deployed within that subscription.
- Organizations have flexibility in terms of how to structure workspaces. Workspaces can include domains consolidated along dimensions such as compliance requirements; usage pattern; scale requirements; teams etc.

When may organisations choose this model?

- Grouping of a function (or functions) under a single Synapse workspace. Example - An organization may choose to deploy a dedicated Synapse workspace for data ingestion function where all pipelines are deployed to a dedicated workspace.
- Organisations may opt-in to implement a medallion lakehouse architecture and centralize lakehouse operations whilst enabling domains access to data via bronze, silver, and gold layers.
- Separation of code base as separate workspaces ship with dedicated version control repo.
- Some domains could be grouped based on location as region of deployment is determined by Synapse workspace. Example – Synapse workspace housing domains for region A and a separate workspace for domains in region B.

Skip to primary content Skip to navigation

- Other common reasons to deploy more than one workspace - scale, org structure, teams supporting underlying infrastructure.
- Separate workspaces enable organisations to use separate CMKs for domains or core functions sharing the same domain.
- Separate workspaces maybe required due to domains operating in different regions.

Pattern 4 – A single Azure Subscription with separate workspaces for each domain

The key difference between this pattern and other patterns (discussed above) is that each domain has a separate dedicated Synapse workspace. All the considerations for separate workspaces highlighted in patterns 2 and 3 apply here.

Figure 7 shows a logical view of layout of workspaces within a subscription within context of lakehouse architecture.

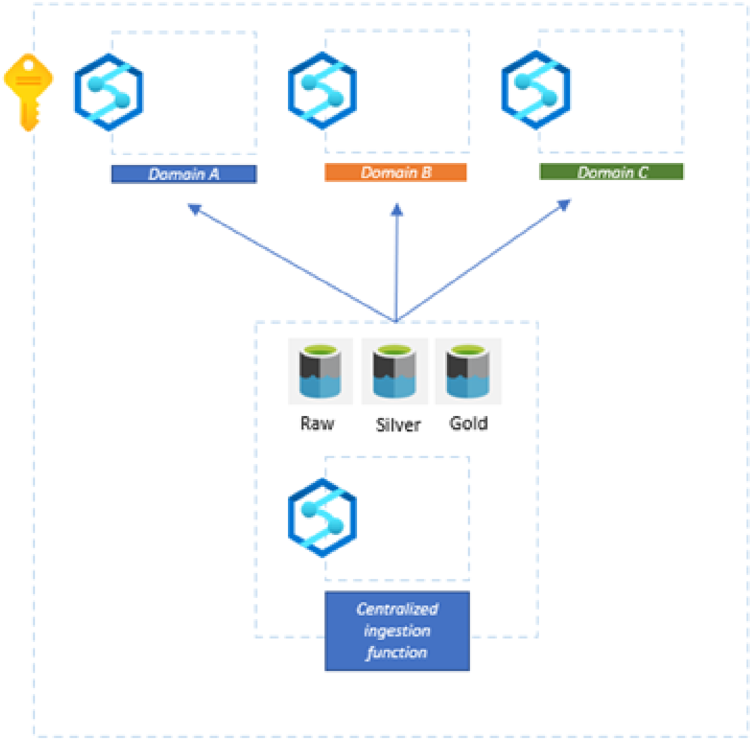


Figure 7 – Separate workspaces for each domain along with a dedicated workspace for a function such as lakehouse medallion architecture.

Pattern 5 – Separate Azure subscriptions with separate workspaces for each domain

This model offers largest scale and highest degree of autonomy to domains. Considerations discussed previously for separate subscriptions and workspaces apply here.

Common reasons for implementing this model include:

- Using subscription as scale, management, and security boundary between domain workspaces.
- Organisations may already have pre-existing Azure landing zones in place and they may choose to deploy Synapse workspaces in those subscriptions closer to application(s) generating data for a given domain. Figure 8 extends the traditional hub-spoke Azure Enterprise Scale Landing Zone model and shows Synapse Workspaces deployed within Azure landing zones.

Skip to primary navigation

Skip to main content

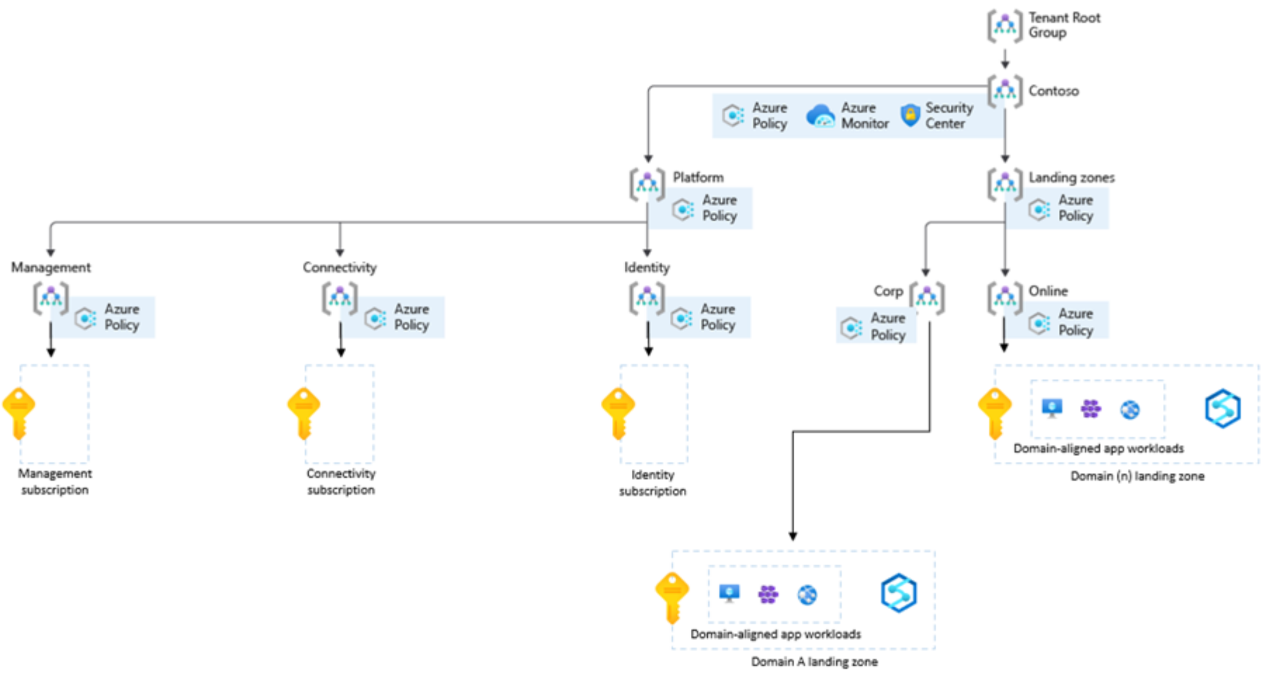


Figure 8 – Each domain with a dedicated subscription and workspace.

Conclusion

- Data mesh is a new architectural pattern for analytics at scale with security and governance baked-in to each deployment unit for a data product.
- As with most architectural patterns, there is no one size fits-all approach and technical implementation of data mesh varies.
- We have observed organisations who are designing and implementing data mesh adhere to its principles to varying degrees. This could be due to one or more of the following reasons:
  - Data volumes, scale and/or complexities (associated with systems implementation)
  - Org structure
  - Team size and skills
  - Internal ops processes
  - Existing investments in a set of technologies
  - Compliance and security requirements etc.
- Synapse Analytics capabilities can be further augmented by leveraging its native integration with the wider Microsoft Intelligent Data Platform ecosystem.
- When it comes to scale and management on Azure, Cloud Adoption Framework applies to technical implementation of data mesh.

In part two of this blog series, we will focus on topics such as ingestion patterns; networking; layout of subscriptions etc. in context of Azure Synapse Analytics and data mesh.

Our team publishes blog(s) regularly and you can find all these blogs here: https://aka.ms/synapsecseblog

For deeper level understanding of Synapse implementation best practices, please refer to Success by Design (SBD) site: https://aka.ms/Synapse-Success-By-Design

References

[1] Zhamak Dehghani (2022), *Data Mesh: Delivering Data-Driven Value at Scale* (O'Reilly Media, Inc, USA).

[2] See note 1 above.

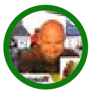
[3] Dehghani (2022), *Data Mesh: Delivering Data-Driven Value at Scale,* “Chapter 9. The Logical Architecture”.

[4] Dehghani (2022), *Data Mesh: Delivering Data-Driven Value at Scale,* “Chapter 12. Design consuming, transforming, and serving data”.

[5] See note 4 above.

9 Likes

4 Comments



Nov 04 2022 02:15

James van den Berg MVP

Thank you @amanjeet for Sharing this awesome blogpost with the community 🤩

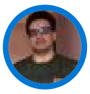
1 Like

Nov 06 2022 03:59

Raja Narayanaswamy Regular Visitor

Great article @amanjeet , I am sure this will be very useful to customers who plan to adopt Datamesh using Synapse. Well done!!

1 Like



Nov 08 2022 09:36

PradipVS Microsoft

Thanks @amanjeet for this wonderful blog post. have you implemented / part of the implementation for datamesh for any customers?

0 Likes

Dec 04 2022 05:08

RK Iyer Microsoft

Great article @amanjeet. This will be super useful to customers implementing Datamesh using Synapse. Thanks for Sharing with broader community. Looking forward for part 2!!!

0 Likes

You must be a registered user to add a comment. If you've already registered, sign in. Otherwise, register and sign in.

Skip to primary navigation Skip to main content

Comment

Co-Authors

amanjeet

Version history

Last update: Nov 03 2022 10:37 AM

Updated by: swoeng

Labels

Community	5
Synapse CSE	15
Synapse DevOps	17
Synapse Pipelines	32
Synapse Spark	54

‹ Previous Next ›

Share

What's new

Surface Pro 9

Surface Laptop 5

Skip to primary navigation Skip to main content

Surface Studio 2+

Surface Laptop Go 2

Surface Laptop Studio

Surface Duo 2

Microsoft 365

Windows 11 apps

Microsoft Store

Account profile

Download Center

Microsoft Store support

Returns

Order tracking

Virtual workshops and training

Microsoft Store Promise

Flexible Payments

Education

Microsoft in education

Devices for education

Microsoft Teams for Education

Microsoft 365 Education

Education consultation appointment

Educator training and development

Deals for students and parents

Azure for students

Business

Microsoft Cloud

Microsoft Security

Dynamics 365

Microsoft 365

Microsoft Power Platform

Microsoft Teams

Skip to primary navigation

Skip to main content

Microsoft Industry

Small Business

Developer & IT

Azure

Developer Center

Documentation

Microsoft Learn

Microsoft Tech Community

Azure Marketplace

AppSource

Visual Studio

Company

Careers

About Microsoft

Company news

Privacy at Microsoft

Investors

Diversity and inclusion

Accessibility

Sustainability

Sitemap Contact Microsoft Privacy Manage cookies Terms of use Trademarks Safety & eco
About our ads