Microsoft

Implement a governed Data Platform using Microsoft Fabric and Microsoft Purview



Piethein Strengholt
CDO Microsoft Netherlands



Sarath Sasidharan
Senior PM Fabric CAT

Data Pancakes
https://www.youtube.com/@DataPancakes

Latest trends and developments



Data marketplace envisioning

Develop a future in where data is easily accessible and quickly available to anyone who needs it.



Self-serve data analytics

Empowering customers with self-service and personalized user experience, so they can drive the value creation themselves..



Embracing data product thinking

Data product thinking by treating data as a valuable asset that can be leveraged to create innovative products and services.



Architectures for specific business units

Scalability by domain-orientation to support the needs of specific business domains.



Infusion of AI-based assistant models

Enhanced report creation, data analysis, data engineering, SQL and Python development using Copilots.



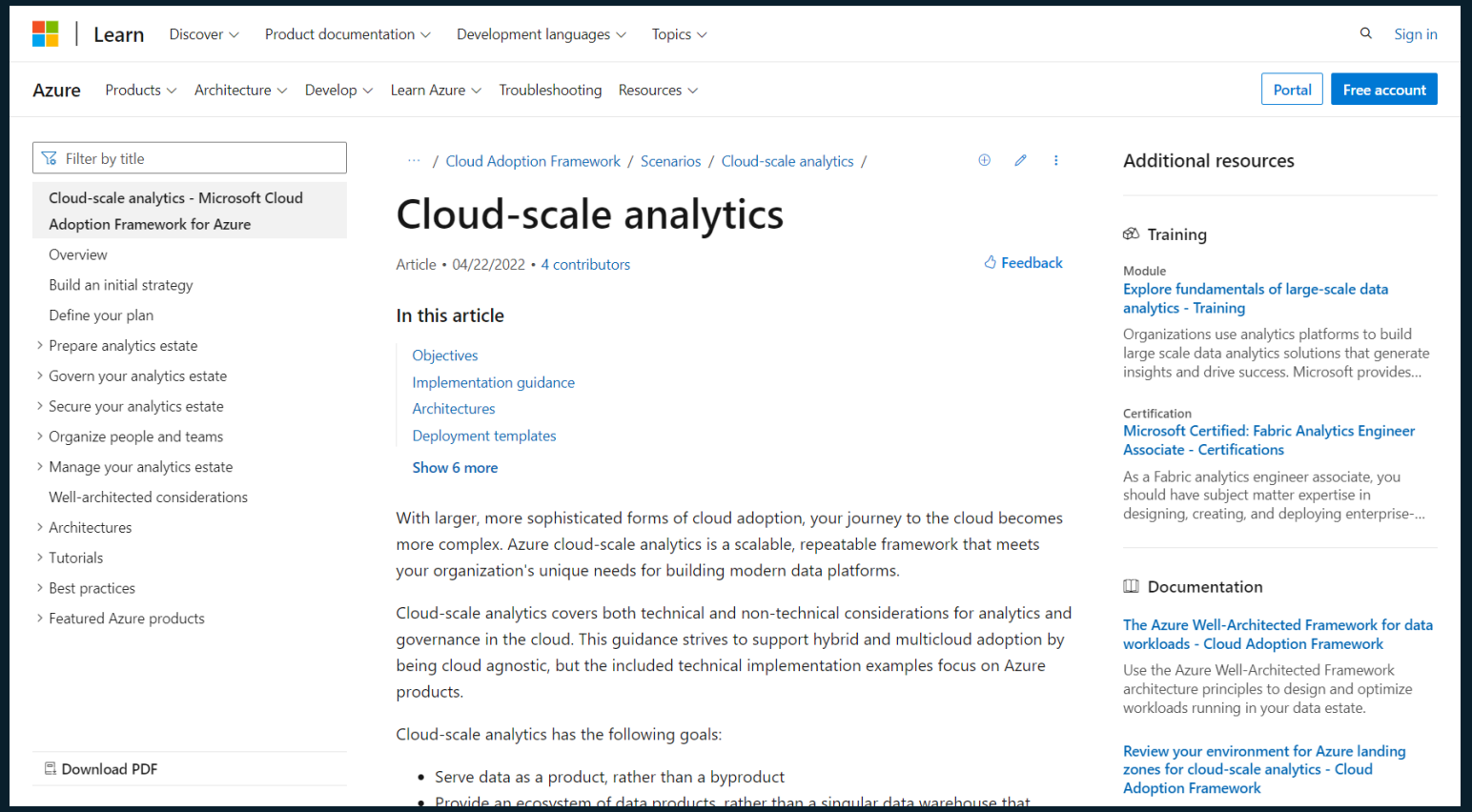
Computational and community-based data governance

Strong governance to improve data quality, ownership, and accountability by establishing standardized governance policies and procedures.

What are today's best practices?

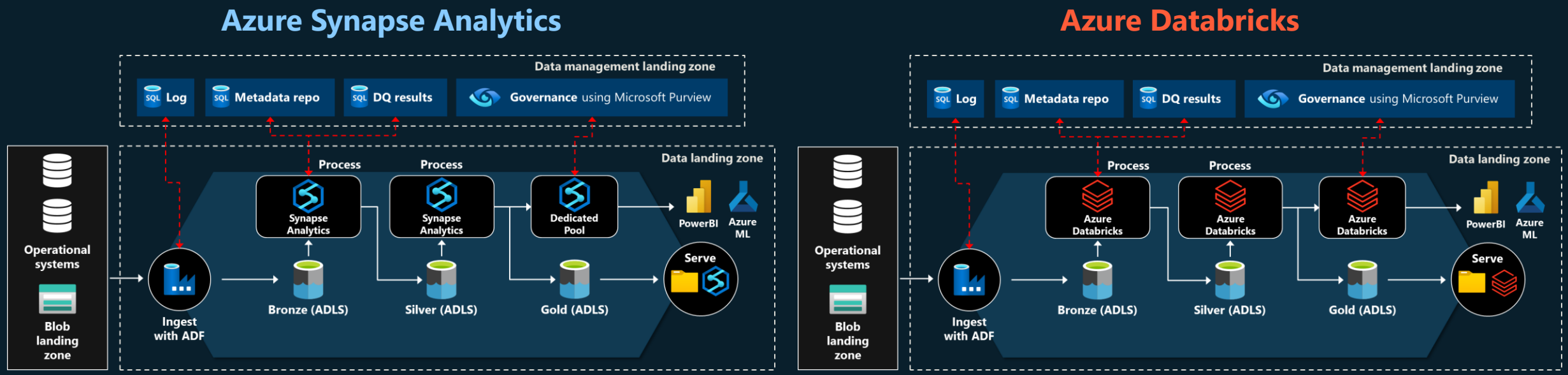
- Cloud Adoption Framework provides a structured approach to accelerate cloud option
- Subscription democratization for scale
- Start by implementing Azure data landing zones
- Use infrastructure as code for consistency
- Provide an ecosystem of data products, rather than a singular data warehouse
- Horizontal teams taking ownership of applications, ingestion, data product creation, as well consumption
- Use a data catalogue (Azure Purview) for providing transparency
- Use a Medallion architecture for layering data. Storage accounts for logically separating data.
- Use either Azure Synapse Analytics and/or Azure Databricks for data processing
- Compliment the architecture with Azure Data Factory, Azure ML and PowerBI

Cloud-scale analytics: considerations for analytics and governance in the cloud

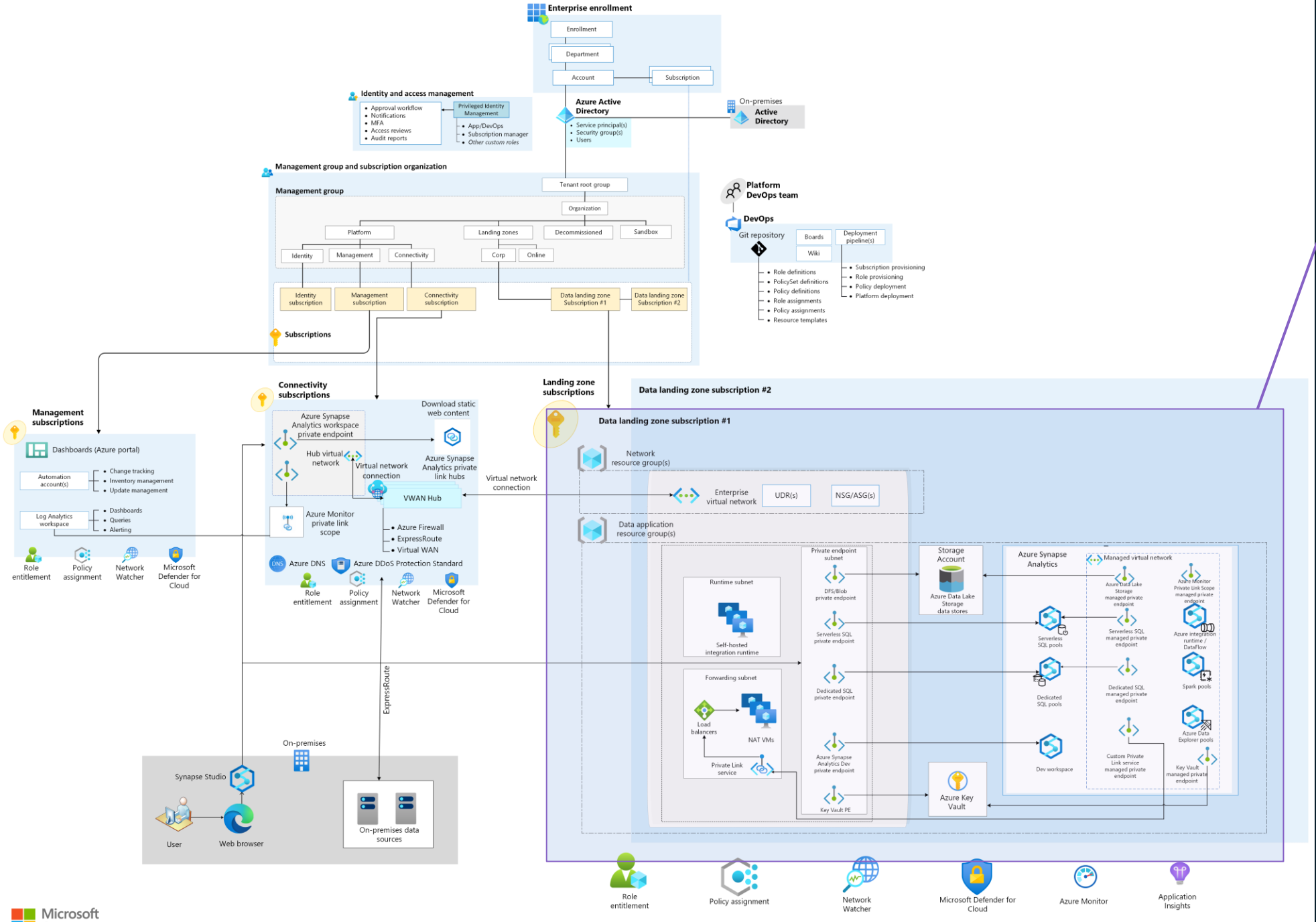


aka.ms/adopt/cloudscaleanalytics

Medallion architecture looks similar for Synapse and Databricks



- Both environments need to be integrated with ADLS storage accounts
- Both environments offer serverless compute.
- Both environments require integration with ADF for ingestion/orchestration and PowerBI for reporting.
- Both environments use workspaces for logically grouping concerns.
- Both environments are Platform as a Service, hence you provision and manage them via infrastructure templates

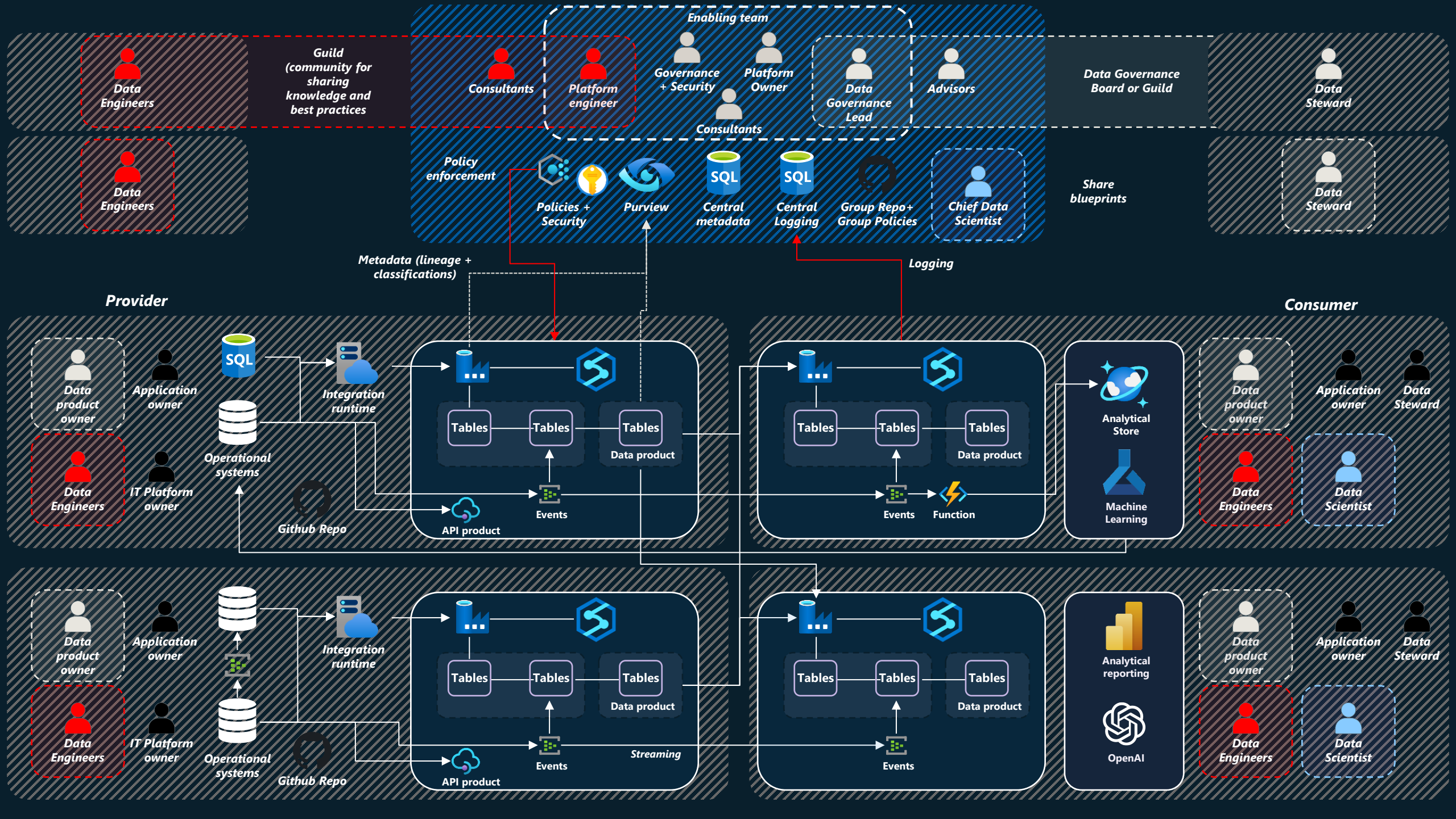


Each region or functional area gets its own data landing zone

Within each landing zone there are sets of resource groups and workspaces for all the different teams.

Additionally, there is a data management landing zone for overseeing the whole technical environment

Microsoft Azure

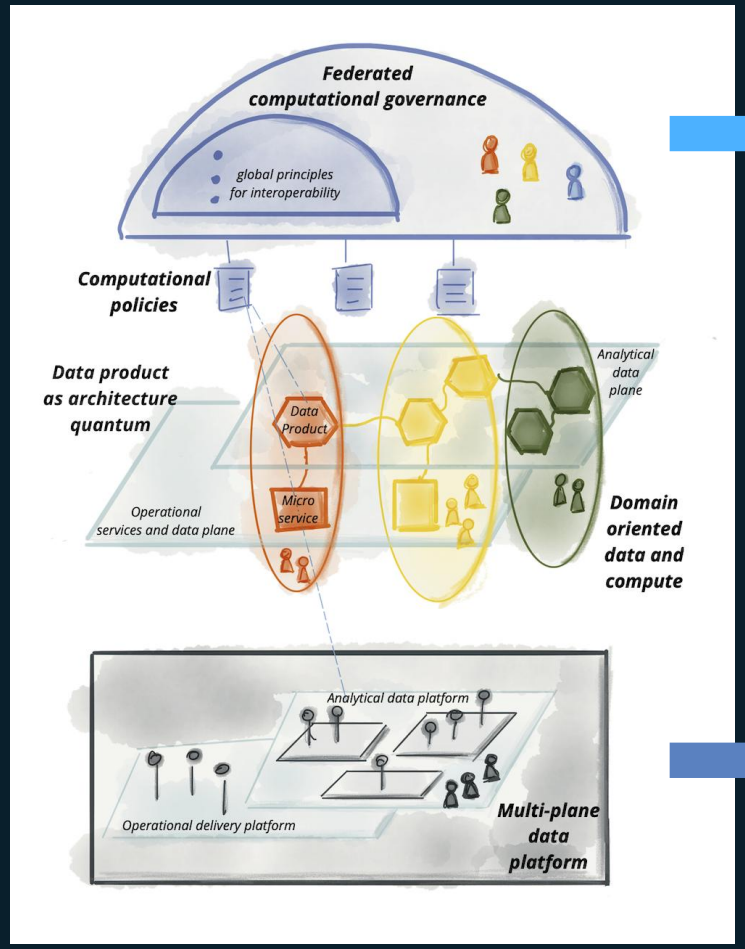

Enabling team
Data Engineers
Guild (community for sharing knowledge and best practices)
Consultants
Platform engineer
Governance + Security
Platform Owner
Consultants
Data Governance Lead
Advisors
Data Governance Board or Guild
Data Steward
Data Engineers
Policy enforcement
Policies + Security
Purview
Central metadata
Central Logging
Group Repo+ Group Policies
Chief Data Scientist
Share blueprints
Data Steward
Metadata (lineage + classifications)
Logging
Provider
Consumer
Data product owner
Application owner
Data Engineers
IT Platform owner
Operational systems
Integration runtime
Github Repo
Tables
Data product
API product
Events
Function
Analytical Store
Machine Learning
Data Scientist
Analytical reporting
OpenAI
Streaming

What pain points do we mainly see?

The initial focus is mainly on technical integration, as it poses significant challenges.

- Blueprints require tactical choices regarding the number of storage accounts, different permission and role models, and cost management strategies, among other factors.
- Every service comes with its own set of limitations, such as the number of private endpoints or storage accounts.
- Challenges often arise when sharing and securing data, with data duplication frequently being the best practice.
- Implementing self-service can be difficult due to the need to balance complex automation with simple human interaction.
- Everyone receives an instance of the same blueprint, regardless of their specific requirements, which can lead to overprovisioning of services.
- Additional components are necessary for parameterized approaches, security workflows, and other needs.

Strategic direction: moving to a higher level of abstraction



Microsoft Purview

+

Microsoft Fabric

Data Mesh Principles by Zhamak Dehghani

Zoom out and look at what your organization does: most companies organize themselves by business capabilities, workflows or journeys

Terminal and landside management

- Asset management
- Real estate management
- Facilitates management
- Customer and information management
- Airport and lounges management
- Safety and incident management
- Advertising and marketing
- Ground and operations management

Airflight and baggage management

- Terminal management
- Baggage handling management
- Flight optimization planning
- Safety and Planning management
- Airflight information management
- Trunk transportation management
- Commercial management
- Operations management

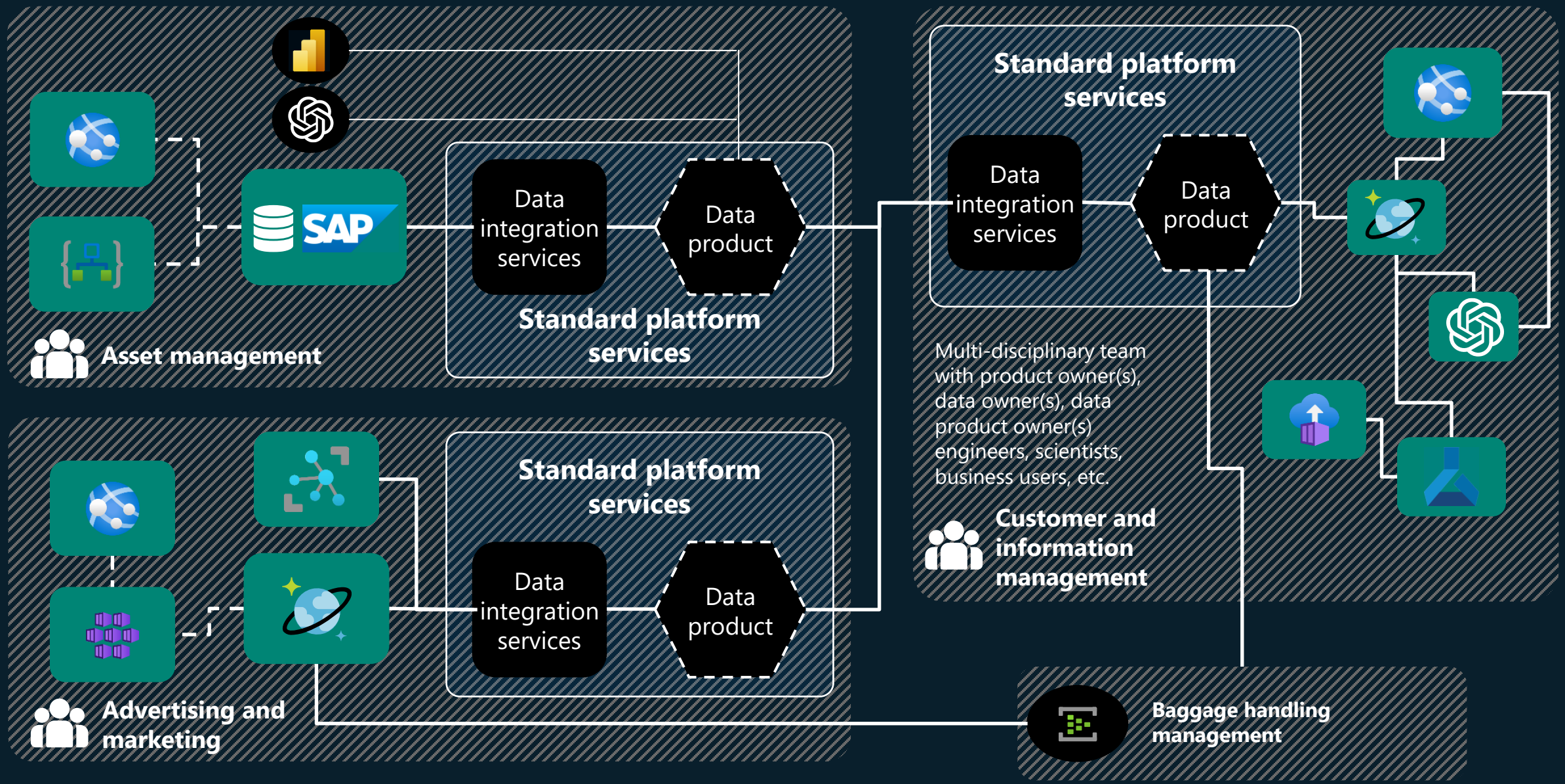
Staff and personnel management

- Recruitment & employee management
- Personnel support management
- Groundcrew personal management
- Terminal personal management
- Hall personal management
- Landside personal management
- Productivity management
- Security personal management

Supporting services management

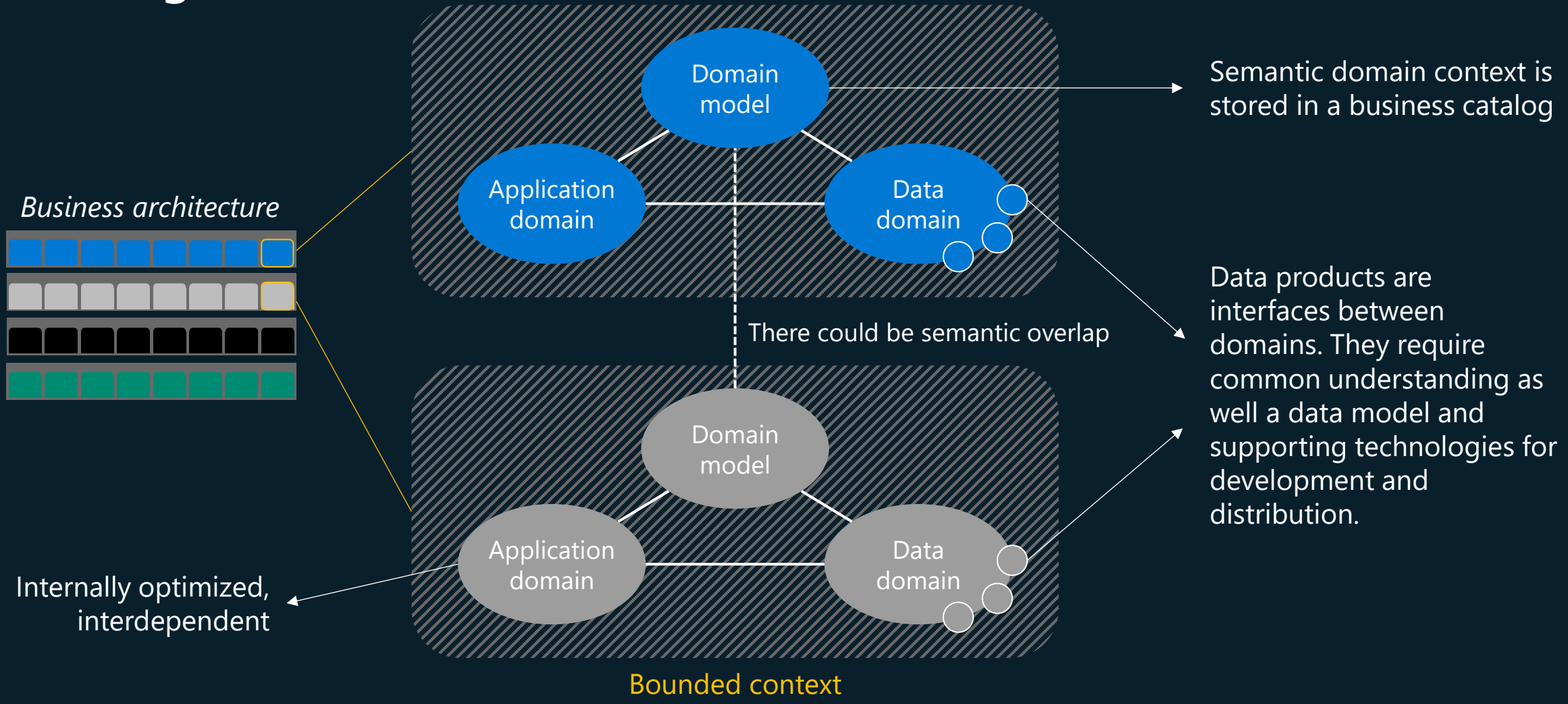
- Assets and financing
- Cost management
- IT services management
- Car leasing and pick up services
- Income and taxes management
- Partnership and communication
- Emission and fare trading management
- Regulatory procedures management

Many business domains generate value via digital feedback loops

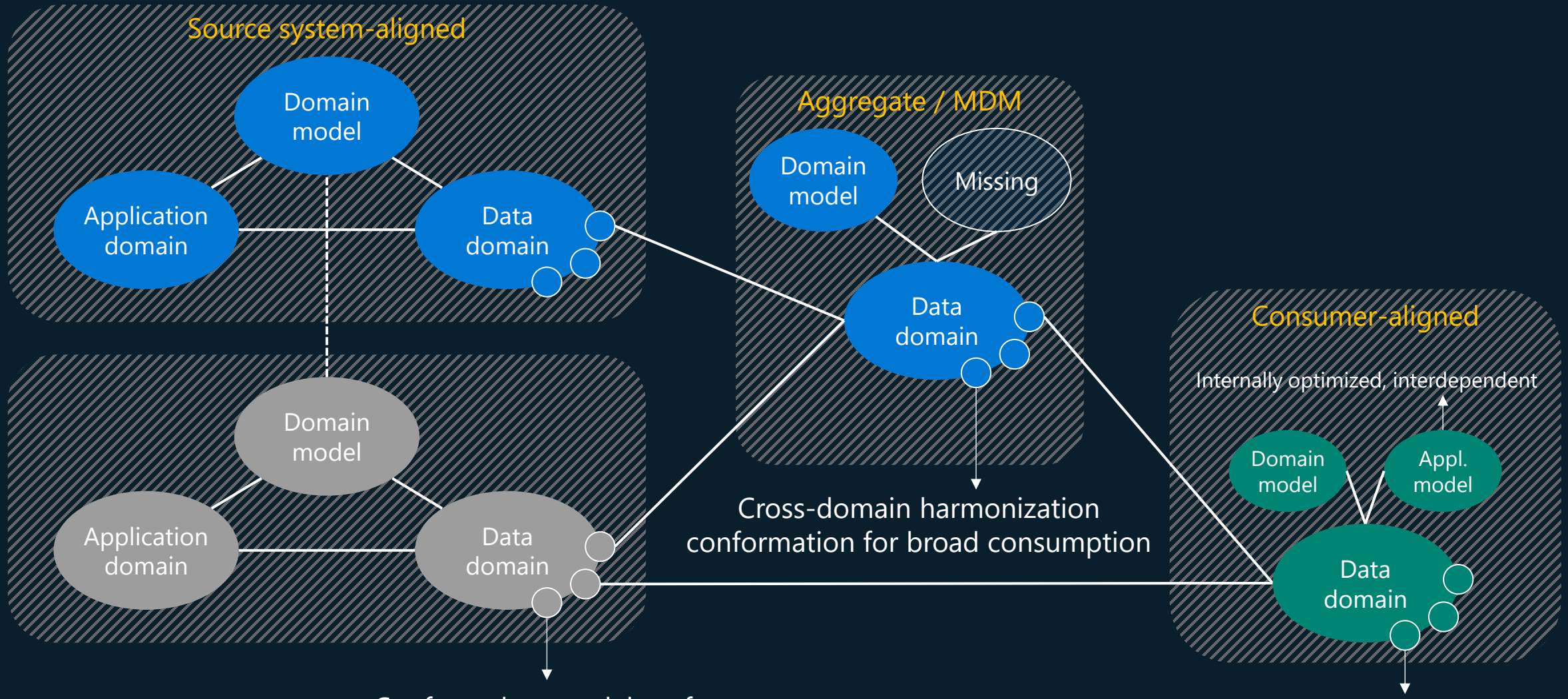


Demo Microsoft Fabric

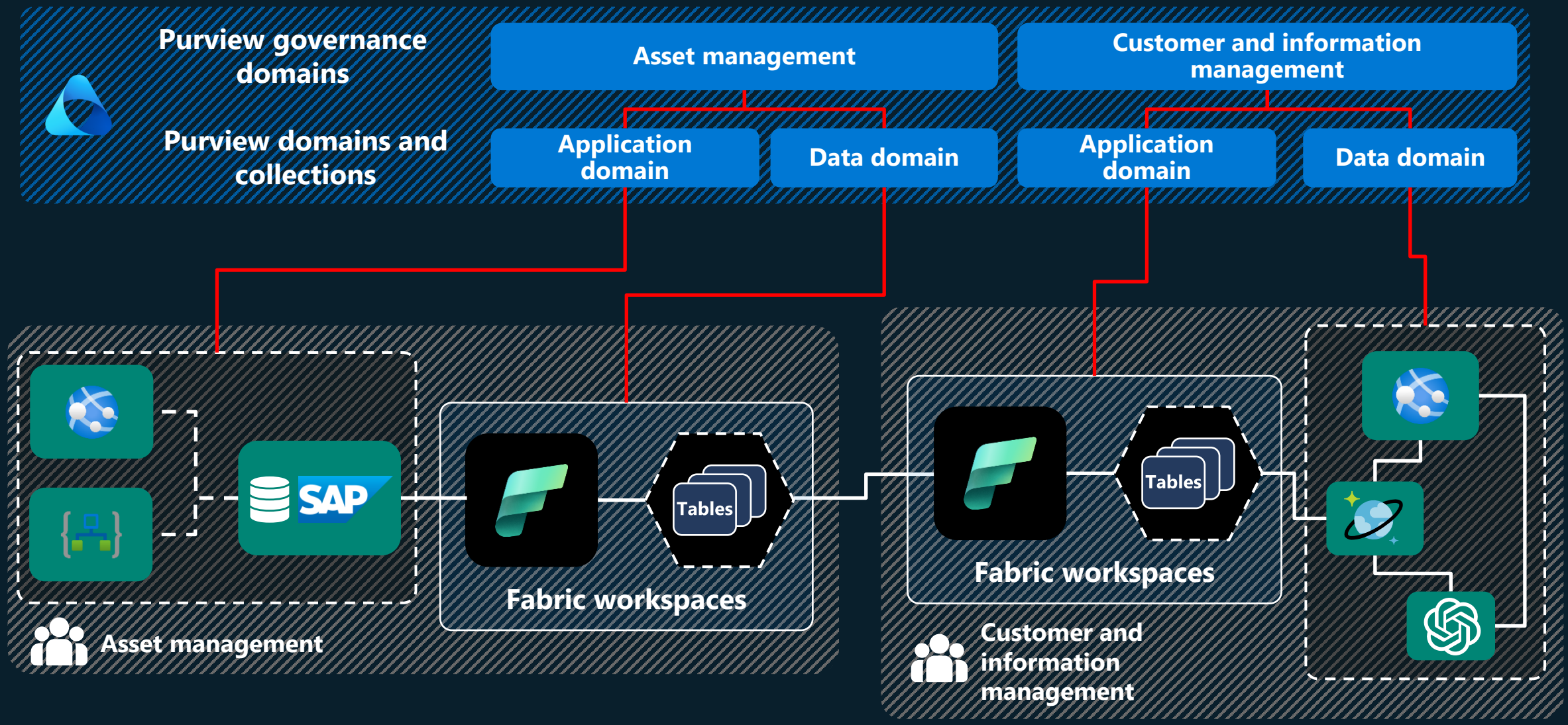
Within a boundary, all concepts have specific and unambiguous meanings



Different types of conformation and harmonization



Glueing business domains, application domains, data domains and collections together



Demo Microsoft Purview

What are good recommendations

- Define a standard for data product, including guidance for modelling
- Work out and align personas with roles
- Setup your data governance process
- Think about skilling and education
- Establish a community and a program for driving changing
- Define a target operating model and decide for balancing central vs decentral alignment of activities
- Security model needs to be defined and worked out

Typical personas you might have within a business domain



Airflight operations management



Business owner: *The person who plays a strategic role and manages the day-to-day business operations.*



Application owner: *The person responsible for the application's availability, performance, change and capacity planning.*



Data owner: *Typically, a senior business executive who has overall responsibility for the data within an organization. Responsible for data quality, data privacy, data control, data lifecycle management, and data strategy.*



Data product owner: *This role involves leading the development team that builds and maintains data-oriented products, working with stakeholders to understand their needs, prioritize tasks, and ensure data products deliver value.*



DevOps engineer: *person responsible for the development and maintenance of applications and systems infrastructure.*



Data engineer: *This role involves preparing data, which includes design, modelling, development, performance testing, etc.*



Analyst: *A data analyst collects, processes, and performs statistical analysis of data.*



Report builder: *This role involves collecting, interpreting, and visualizing data for the purpose of creating comprehensive reports.*



MLOps / LLMs engineer: *responsible for integrating and scaling machine learning or large language models into production software and systems. They work closely with data scientists to understand and implement algorithms.*



Data scientist: *A data scientist's role is to convert data into meaningful insights using algorithms, machine learning and LLMs.*



Data steward: *A data steward ensures the company's data quality, usability, and security. Responsible for data governance, data management standards, and data quality assurance.*

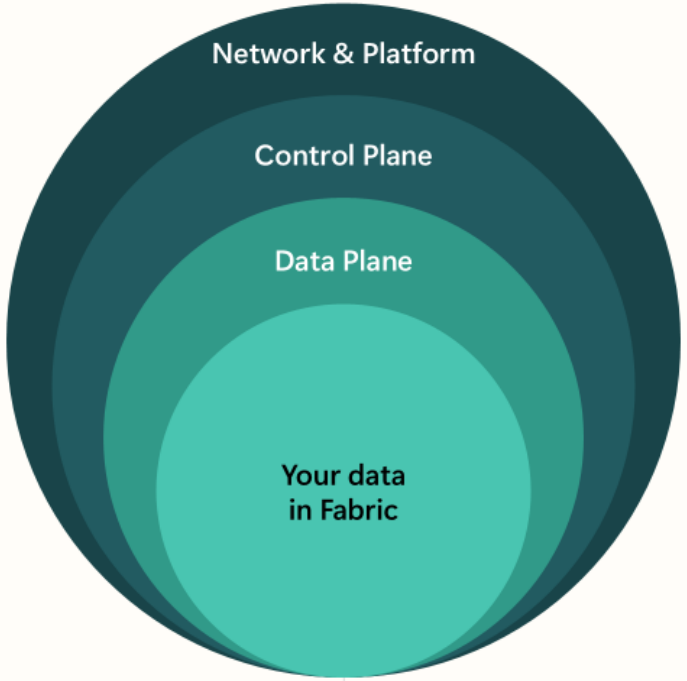


Business user: *Anyone who uses the data for decision-making processes, analyses, and business operations.*

		Microsoft Fabric			Microsoft Purview	
Persona	**Environment**	**Domain**	**Workspace**	**Items**	**Data Map Roles**	**Data Catalog Roles**
Data Owner	DEV					
	TEST					
	PROD		Viewer			Data product owner
Data Engineer	DEV		Contributor		Data Source Admin	Data quality steward
	TEST		Contributor		Data Source Admin	Data quality steward
	PROD				Data Source Admin	Data quality reader
DevOps Engineer	DEV	Domain contributor	Admin			
	TEST		Admin			
	PROD					
Analyst	DEV					
	TEST				Data Estate Insights Reader	Data health reader
	PROD		Viewer		Data Estate Insights Reader	Data health reader
Business user	DEV					
	TEST					
	PROD			Access to specific report	Data Estate Insights Reader	Data Catalog Reader
Business owner	DEV	Domain admin				
	TEST	Domain admin			Data Estate Insights Admin	Business domain owner
	PROD	Domain admin			Data Estate Insights Admin	Business domain owner

Security Slide

Security layers in Microsoft Fabric



Example model for data literacy curriculum

Organizations should start to build a curriculum to develop data literacy across their organizations. Data literacy is a fundamental skill that helps individuals to make informed decisions, solve complex problems, and communicate effectively with others.

Level 3: Essential for all pros

- **Intro on architecture**
- **Deep-dive on architecture**
- **Leadership employee attributes**
- **Deep-dive on ML & AI**
- **Deep-dive Governance**
- **Deep-dive Data Engineering**
- **Self-Service**

Level 2: Essentials for all data co-workers

- **Intro to governance**
  Course on data marketplace, processes, roles and responsibilities
- **Data products**
  Workshop on creation and distribution of data products
- **Data Ethics & AI**
  Course on principles for responsible AI
- **Data & AI**
  Course for executives

Level 1: Introduction course

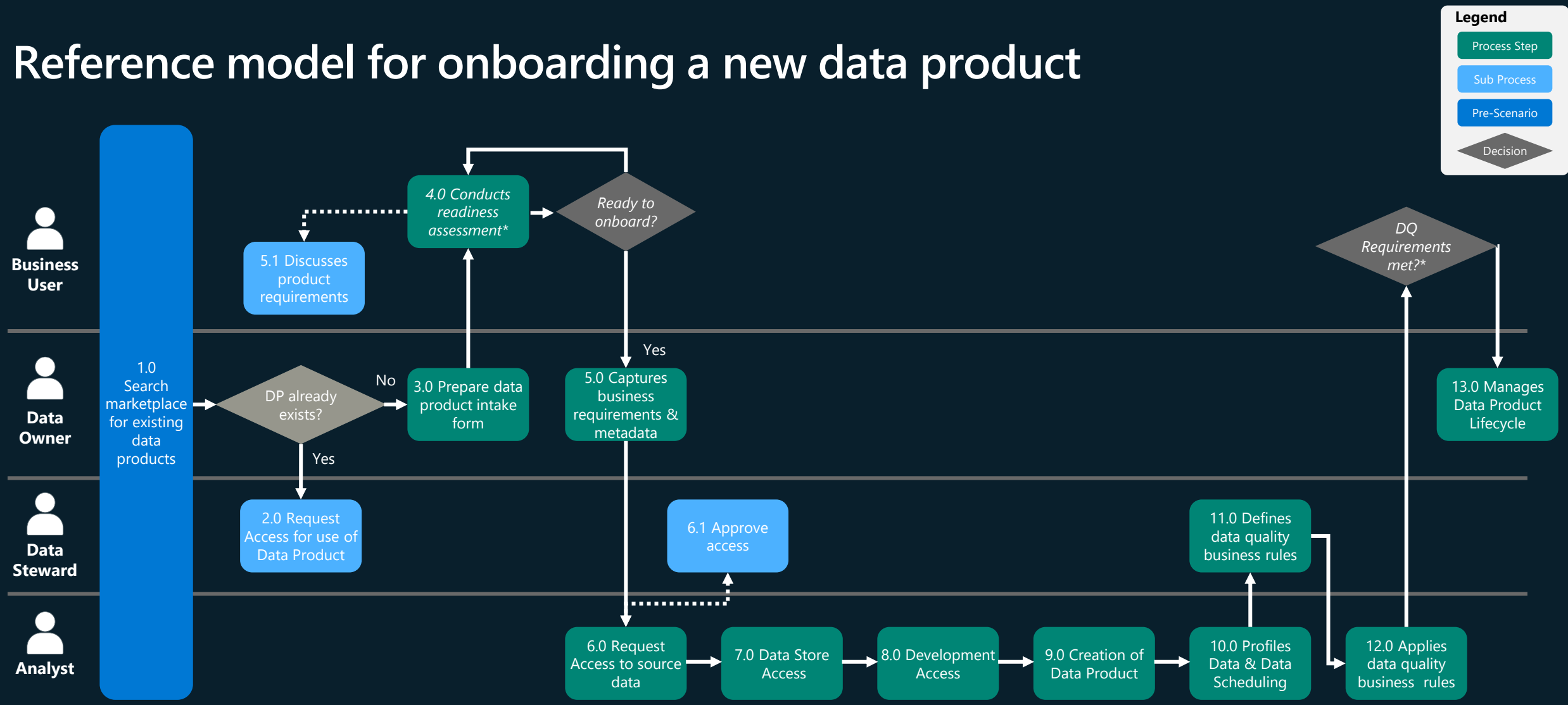
- **Data literacy for everyone**
  Key course with basics of data management, importance of quality, key concepts and the way data management is done within the organization
- **Citizen Development**

Foundational for everyone

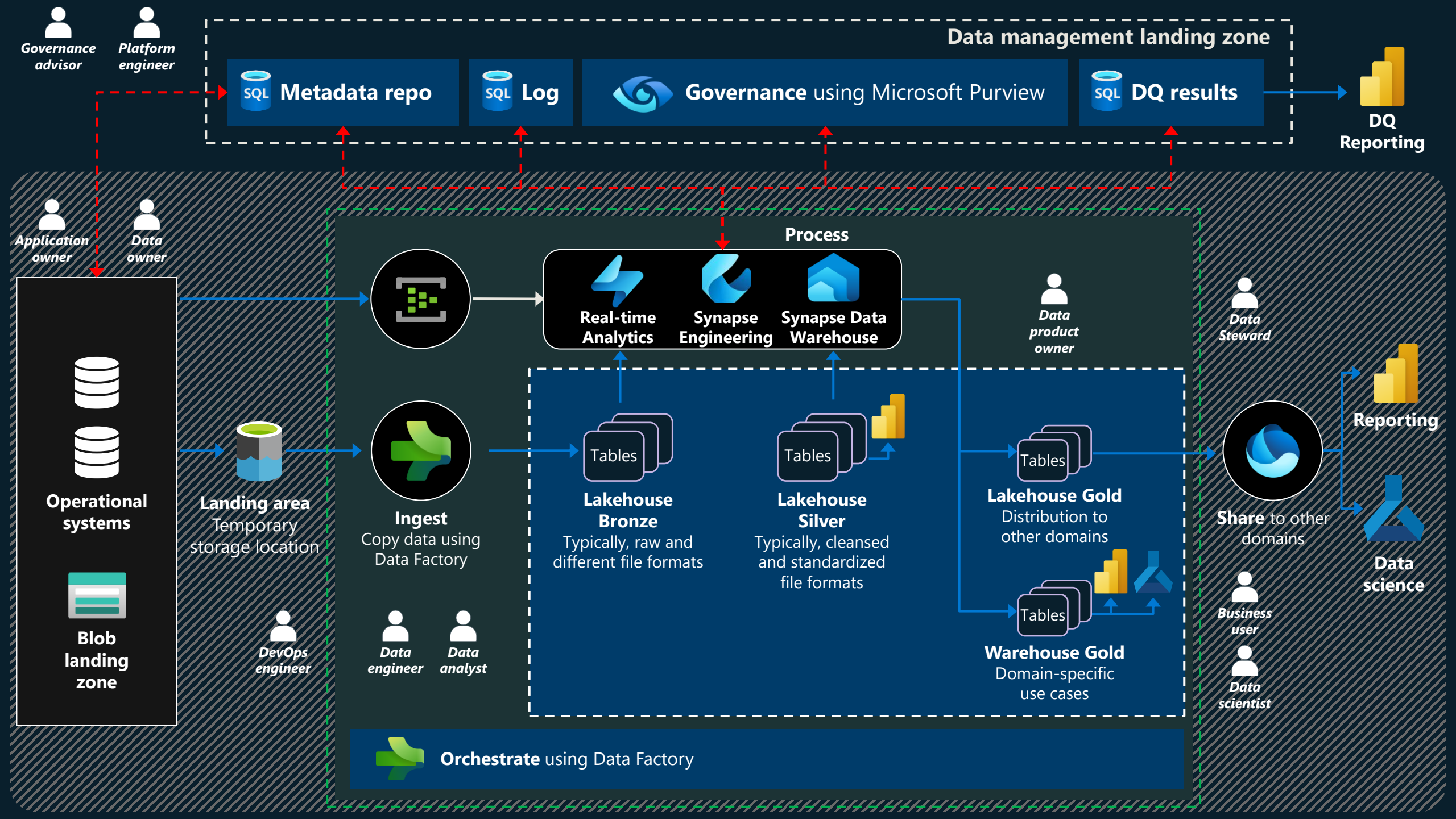
- **Fundamentals of corporate data strategy, benefits of data management and analytics**
  Suggested as optional for employees that don't directly work with information

Optional **Essential** **Advanced** **Workshop**

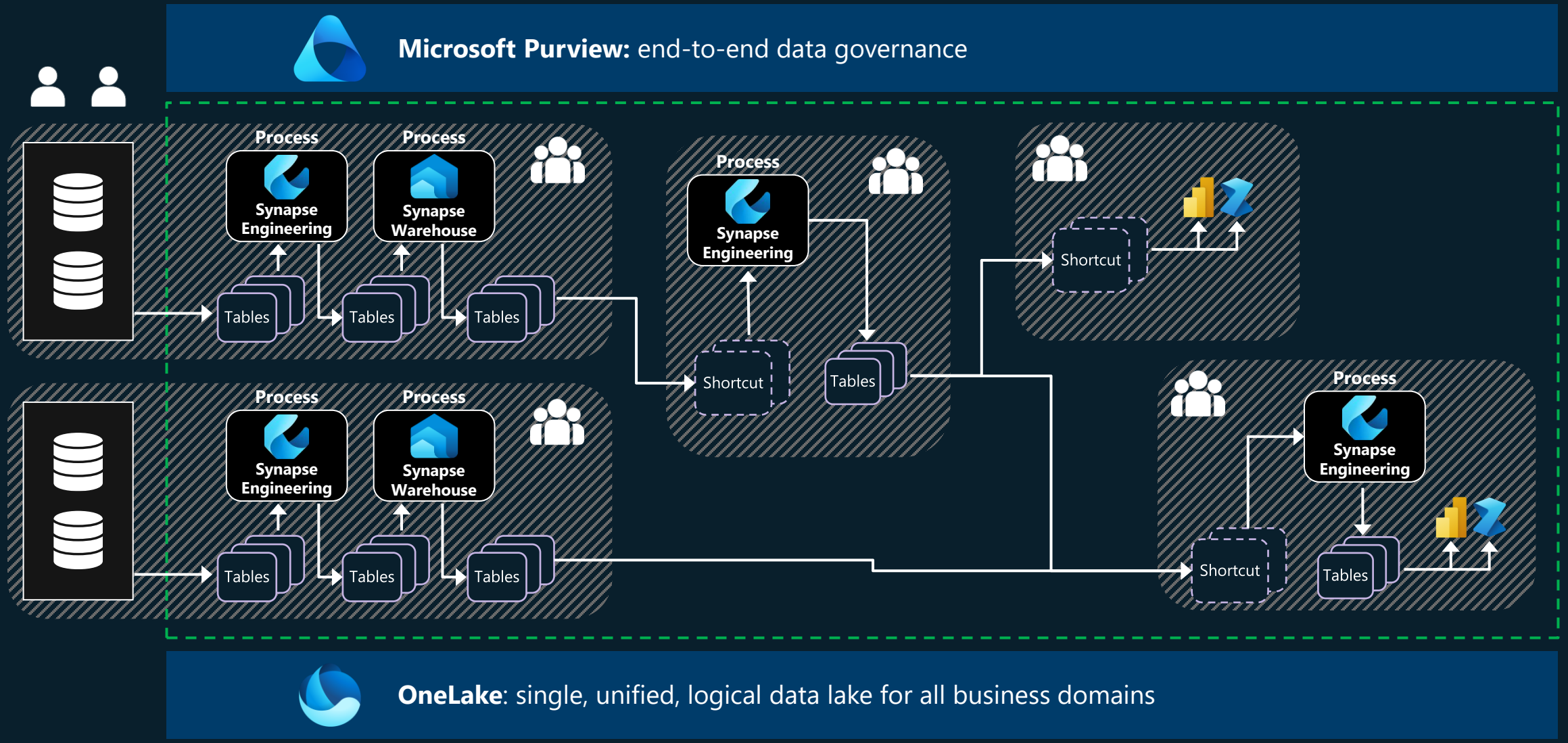
Reference model for onboarding a new data product



Process	1.0	2.0	3.0	4.0	5.0	6.0-7.0	8.0	9.0	10.0	11.0	12.0 / 13.0
Solution	Microsoft Purview	Microsoft Purview	Microsoft Purview / Microsoft Forms	Microsoft Purview	Microsoft Purview	Microsoft Fabric	Microsoft Fabric	Microsoft Fabric	Microsoft Purview	Microsoft Purview	Microsoft Purview
Technical steps	Query metadata from Purview data map	Trigger workflow + Configure ACLs	Update Azure DevOps	Update Azure DevOps	Data product template + business glossary	Linked services, ACLs, Update workspaces	Create new pipeline, add metadata	Develop notebooks and SQL scripts	Add new DQ connection	Add DQ rules and metrics	Monitor data health insights


Governance advisor
Platform engineer
Data management landing zone
Metadata repo
Log
Governance using Microsoft Purview
DQ results
DQ Reporting
Application owner
Data owner
Process
Real-time Analytics
Synapse Engineering
Synapse Data Warehouse
Data product owner
Data Steward
Operational systems
Blob landing zone
Landing area
Temporary storage location
Ingest
Copy data using Data Factory
Tables
Lakehouse Bronze
Typically, raw and different file formats
Tables
Lakehouse Silver
Typically, cleansed and standardized file formats
Tables
Lakehouse Gold
Distribution to other domains
Tables
Warehouse Gold
Domain-specific use cases
Share to other domains
Reporting
Data science
DevOps engineer
Data engineer
Data analyst
Business user
Data scientist
Orchestrate using Data Factory

Refreshed architecture with abstracted services



Best practices for implementing Microsoft Fabric

Domain-configuration:

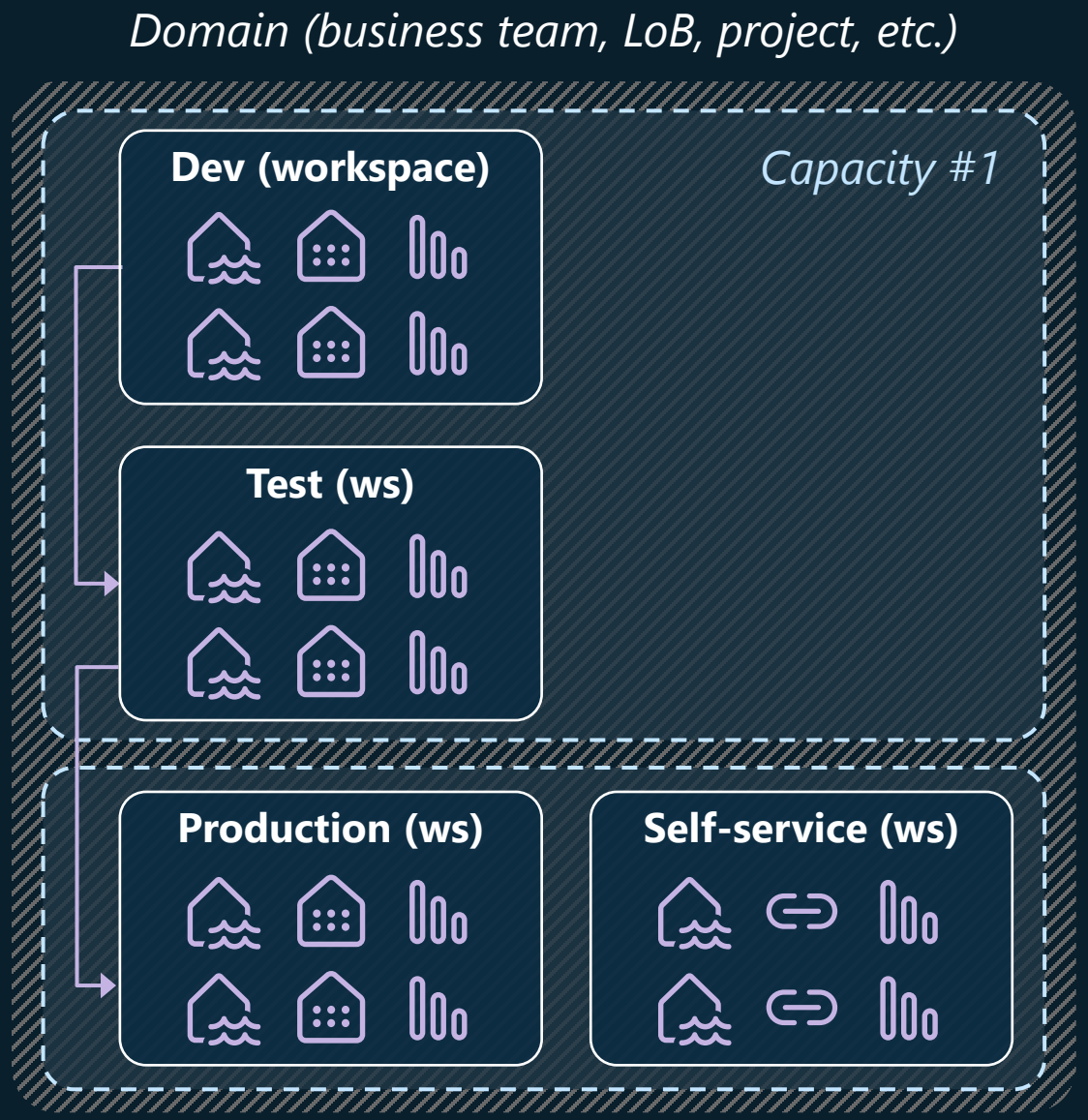
- Logically group workspaces per business domain, data team, project, line of business, etc.
- Larger domains may have subdomains.

Workspace-configuration:

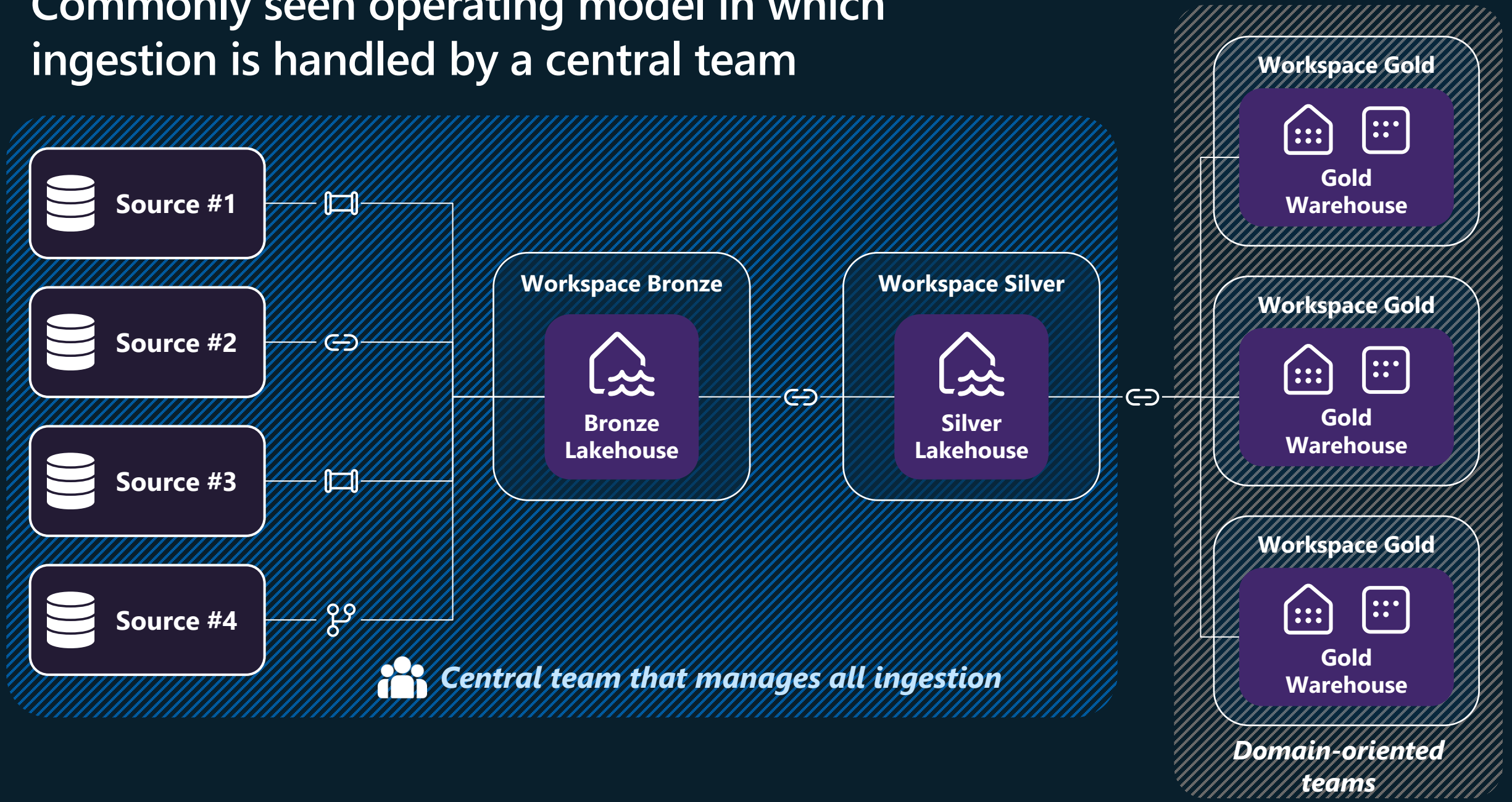
- Separate the concerns of Dev, Test and Production.
- Use as a boundary for access control / security.
- Implement additional workspaces for different Spark pools, workflows, workspace or network identities, self-service activities, etc.

Capacity-configuration:

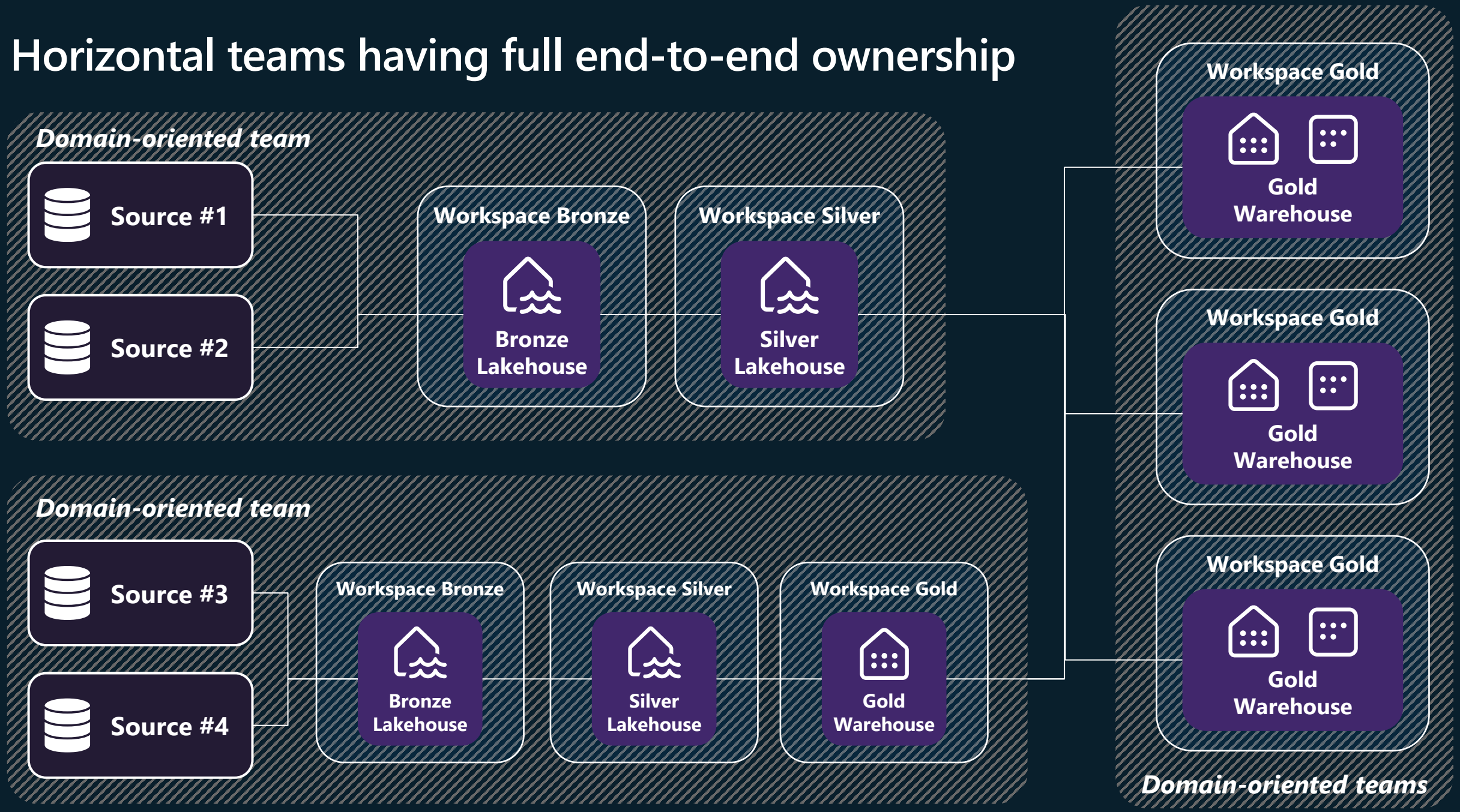
- Capacities align to regions in which data is stored.
- Use capacities for optimizing workloads, such as intensive data processing, ad-hoc reporting, etc.



Commonly seen operating model in which ingestion is handled by a central team



Horizontal teams having full end-to-end ownership



Data product design recommendations

- Intro: Rationale of the guidance, importance of data product design, data quality, etc.
- Guidance on roles, development and onboarding process, registration process
- Guidance on applications, unique sources and encapsulating other source systems
- Guidance on read-optimized data; data structure, grouping, reference data, data types, schema information, keys, classification, etc.
- Guidance on atomic data; linkage to business terms within catalog
- Guidance on concatenated data
- Guidance when splitting fields
- Guidance for internal and cross-system keys
- Guidance for business logic in data
- Guidance on handling missing values
- Guidance on defaulted data; system requires a mandatory field, but data misses. E.g. missing birth date.
- Guidance on local-to-local reference mappings
- Guidance on local-to-enterprise reference mappings
- Guidance on encapsulating or providing metadata for row-level filtering
- Guidance on reserved column names
- Guidance on historical data (metadata instructions for append-only, overwriting, merging and guidance on how to structure your data)
- Guidance on shared administrations with multiple owners
- Guidance on snapshots from legacy systems (data warehouses)
- Guidance on Master Data Management
- Guidance on re-deliveries and overwrites
  Guidance on building aggregates, consuming and distributing
- Interoperability patterns
- Process of escalation, discussion boards
- Backwards compatibility principles, such as versioning interfaces
- Metadata guidance (metadata-driven transformations, etc.)

Typical trade-off or design considerations for building a large-scale architecture

- Consider the definition and granularity of a domain, such as determining the number of horizontal teams or domains you foresee.
- Evaluate the choice between centralization and decentralization, for instance, whether a central team should own data products instead of individual domains.
- Clearly define what constitutes a data product.
- Determine the appropriate level of maturity and automation required, for example, opting for no metadata-driven framework, creating scripts and pipelines manually, and provisioning by hand.
- Determine the appropriate level of central oversight, balancing between strict standardization and the proliferation of tools.
- Determine the appropriate level security and data privacy measures, such as choosing between coarse-grained data consumption or fine-grained security using row-level filtering with dynamic data masking.
- Value the implications of bypassing the principle of first creating a source-aligned data product, which can lead to lower data reuse and ownership issues.
- Determine relaxing data governance principles, which may include having less metadata, no lineage, limited data quality rules, and limited administration.
- Determine the possibility and implications of reusing existing legacy point-to-point interfaces.
- Determine if data products may be perceived as too raw or technical.

Conclusion

- Progression over perfection. Start using smaller iterations.
- Develop a strategy to align business domains with application domains, data domains, and data products.
- Determine the most suitable trade-offs or design considerations for the initial implementation phase.
- Adopt design best practices, including the use Lakehouses and Medallion architectures.
- Promote clear guidelines for data products to encourage extensive data reuse.
- Cultivate a culture focused on data product thinking and robust governance. The latest release of Microsoft Purview provides valuable best practices.
- As you progress, the future of data architecture lies in policy and configuration management, rather than infrastructure management.

Rate Our Session

