TFG

Data Management Event 30-January 2024

Module:
(Enterprise) Data Architecture

Ben Teunissen / The Future Group

INDEX

TFG

Introduction

1. Let's define terminology:
   ➔ Enterprise Architecture, Data Architecture (in a narrow sense) and Data Architecture (in a broad sense)
2. Now let's further define Data Management (at scale) based on these terms

ENTERPRISE ARCHITECTURE

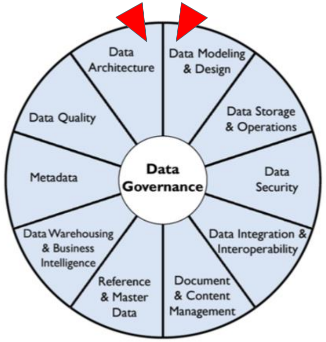
e.g.,
- **Business Architecture**
- **Application Architecture**
- **Information Architecture**
- **Technical Architecture**

e.g.,
- **Integration Architecture**
- **Security Architecture**
- **Etc.**

DATA ARCHITECTURE (Landscape)

Data Modeling

Data Management (at scale)



Source: DAMA

TFG

Definitions

- Let us define Enterprise Architecture, Data Architecture (broad sense); Data Architecture (narrow sense)
- Let us define Data Management (at scale)

Architecture

Source: ISO/IEC 42010:2007 Systems and Software Engineering – Architecture Description (2011)

- The fundamental organisation of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution.
- **Depending on context, the word *architecture* can refer to a description of the current state of systems, the components of a set of systems,** the discipline of designing systems (architecture practice), the intentional design of a system or a set of systems (future state or proposed architecture), the artefacts that describe a system (architecture documentation), or the team that does the design work.
- Architectural practice is carried out at different levels within the organisation ((enterprise, domain, project etc.) and with different areas of focus (e.g. infra, application, data).
- Architecture frameworks bring value because they enable non-architects to understand relationships between levels and focus areas.

Data Architecture (broad sense)

- Artefacts include specifications used to describe existing state, define data requirements, guide data integration, and control data assets as put forth in the data strategy. **An org's DA is described by an integrated set of design documents at different levels of abstraction, including standards that govern how data is collected, stored, arranged, used, and removed.**
- **Data Architecture is fundamental to Data Management**. Because most organisations have more data than individual people can comprehend, it is necessary to represent organisational data at different levels of abstraction so it can be understood, and management can make decisions about it.
- Effective management of data and the systems in which data is used and stored is a common goal of the breadth of architecture disciplines.

Data Architecture (narrow sense)

- **The most detailed Data Architecture design document is a formal Enterprise Data Model. It contains data names, comprehensive data and Metadata definitions, conceptual and logical entities and relationships, and business rules.**
- Physical data models are included, but more as a product of data modelling and design, rather than Data Architecture.
- (Enterprise) Data Architecture is most valuable when it fully supports the needs of the entire enterprise.
- (Enterprise) Data Architecture enables consistent data standardisation and integration across the enterprise.

Data Management

Source: DAMA

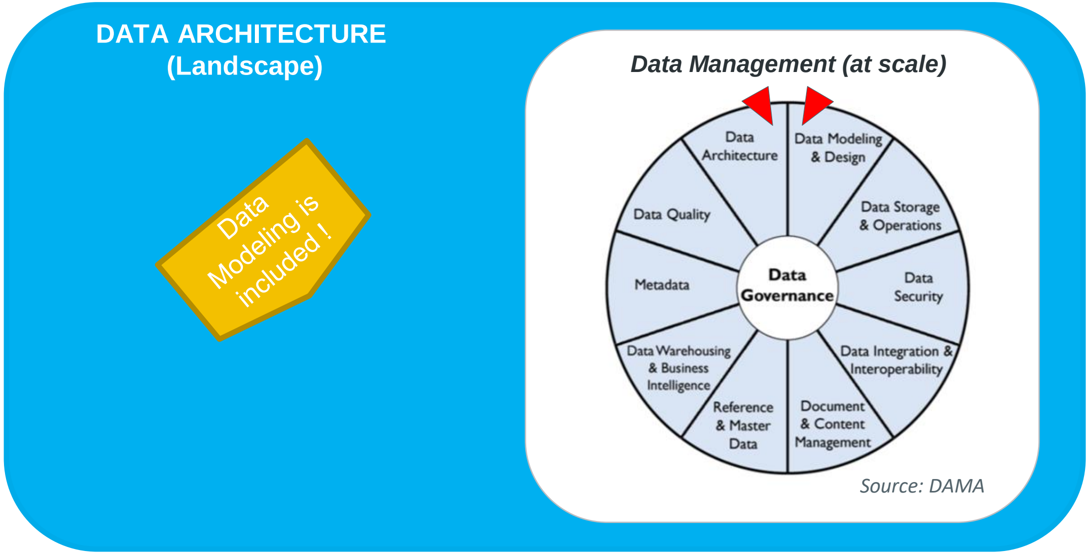
- **Data Management is the development of architectures, policies, practices, and procedures to manage the data lifecycle.**
- It is the process of collecting, keeping, and using data in a cost-effective, secure, and efficient manner.
- Data Architects create and maintain organisational knowledge about data and systems through which it moves.

  This knowledge enables an organisation to manage its data and increase the value it gets from its data by identifying opportunities for data usage, cost reduction and risk mitigation.

TFG

Scoping

- The structures necessary for the execution of Data Management (at scale) are considered part of the overall Enterprise Data Architecture. These structures are aimed at making the enterprise more “data driven”.
- Data Management is the development of architectures, policies, practices, and procedures to manage the data lifecycle. It is the collection, usage and keep of data in a cost-effective, secure, and efficient manner.



TFG

Sidestep: Explain Data Modeling – the WHY

- Data modeling is the process of discovery, analysis and scoping of **data requirements**, and then representing and communicating these data requirements in a **precise form (the Data Model).**
- Data modeling is **critical** to Data Management (and often the starting point).

Goals

1. The goal of data modeling is **to confirm and document the understanding of data** (fit, structure) **from different perspectives.**
2. Data models depict and enable understanding of the data assets.
3. Supports the **built up of Metadata; data models** are an important form of Metadata.

Applied Principles

1. **Formalization: documents the concise definition of data structures and relationships.**
   - Enables the assessment of how data is affected by implemented business rules, for as-is / to-be states.
   - Imposes a formal definition and disciplined structure to data that reduces the possibility of data anomalies when accessing and persisting data. Illustration of structures and relationships between data, makes data easier to consume.
2. **Scope definition: explains the boundaries for data context** and implementation of purchased application packages, projects, initiatives, or existing systems.
3. **Knowledge retention/ documentation:** Preserves the corporate memory on systems or projects by capturing knowledge in an explicit form.
   - Is documentation for future projects to use as the as-is version.
   - Helps to understand an organization, business area, existing application, or impacts to existing data structures.

Benefits

1. Applications are more closely aligned with business requirements
2. Sets a **foundation for broad-scoped initiatives** such as Master Data Management (MDM) and Data Governance (DG)-programs.
3. **Reduces support costs and increases reusability** for future initiatives and therefore reduces the costs of building new applications.

TFG

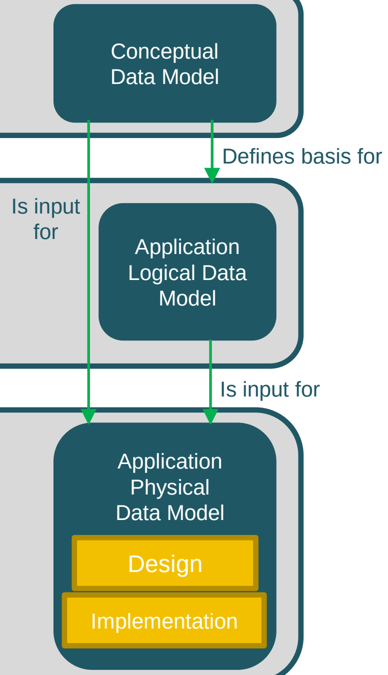
Sidestep: Data Modelling – The WHAT (Levels of Detail)

Within our Data Landscape, we identify 3 model views: Conceptual, Logical and Physical Data Model

Conceptual Data Models: Represent the meaning of and relations between terms as understood by business stakeholders. Underlying technical details are *hidden*, which makes it easier for the target audience to comprehend the model.

Application Logical Data Models: A set of data specifications and related diagrams that reflects the scope and designed functional data structure for a specific application. Used to represent both data at rest (database model) and data in motion (message model).

Application Physical Data Models: A set of data specifications and related diagrams that reflect the physical data being stored or exchanged within a specific scope, and *optimized* for non-functional requirements (performance, security etc.). Enables the generation of database structures for specific database brands, products and versions.



TFG

Back to Data Management (at scale)

- Data Management at scale is the development of architectures, policies, practices, and procedures *to manage data (lifecycle*). It is the collection, usage and keep of data in a cost-effective, secure, and efficient manner.
- For the proper working of the DM-function and its capabilities, *these capabilities need to be reflected in the Enterprise Data Architecture (EDA) - landscape*. So, what would such an architecture look like?

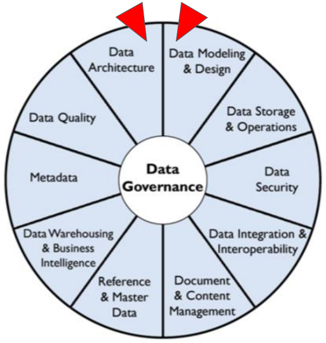
ENTERPRISE ARCHITECTURE

e.g.,
- **Business Architecture**
- **Application Architecture**
- **Information Architecture**
- **Technical Architecture**

e.g.,
- **Integration Architecture**
- **Security Architecture**
- **Etc.**

ENTERPRISE DATA ARCHITECTURE (Landscape)

Data Management (at scale)



Source: DAMA

TFG

The EDA and its Requirements

We gather functional requirements, Non-Functional requirements (NFRs) and identify conditions for Fit for Purpose.

Drivers / Requirements

CTO goals are high level drivers for IT and Data investments; hence any (cloud) investment should contribute to one or more of these goals:

Scalable Growth & Flexibility

- Cost/ Volume ratio
- Speed
- Enhanced decision making

Simplification

- Cost/ Volume ratio
- Reduced # reports/ integrations

Continuity & Compliance

- No business disruption (functionality neutral)
- No surprises on compliance

Non-Functional Requirements

We adopt NFRs for our Data Platform that are generic, technology agnostic and universal:

1. **Simplicity**, limit the # of components and technology
2. **Usability**, provide Domain IT teams with services that they can use without extended technical knowledge
3. **Data Availability**, support fast access to data
4. **Automatability**, aim to automate as much of our data flow processes as possible, driven by standardized patterns and meta data
5. **Extendibility**, onboard new business fast
6. **Governability**, govern the Data Platform in a secure and efficient way

Fit for Purpose - conditions

The following conditions are considered to maintain Fit-for-Purpose:

1. **Company size**, development culture and-resources
2. **Company Type**, e.g., retained organization for outsourcing IT-services
   a) E.g. Microsoft is preferred supplier.
   b) Preference for SaaS solutions.
   c) Preference for proven technology for which adequate knowledge and expertise is available, we prefer a managed service model.
   d) Reduce # of vendors
3. **Use Case types and priority** e.g., first Formal reporting, then AA and ML
4. **Ability to control monthly cost** that are scalable for functionality running on the cloud platform.

TFG

The EDA aligns to the following (Data) Architecture Principles

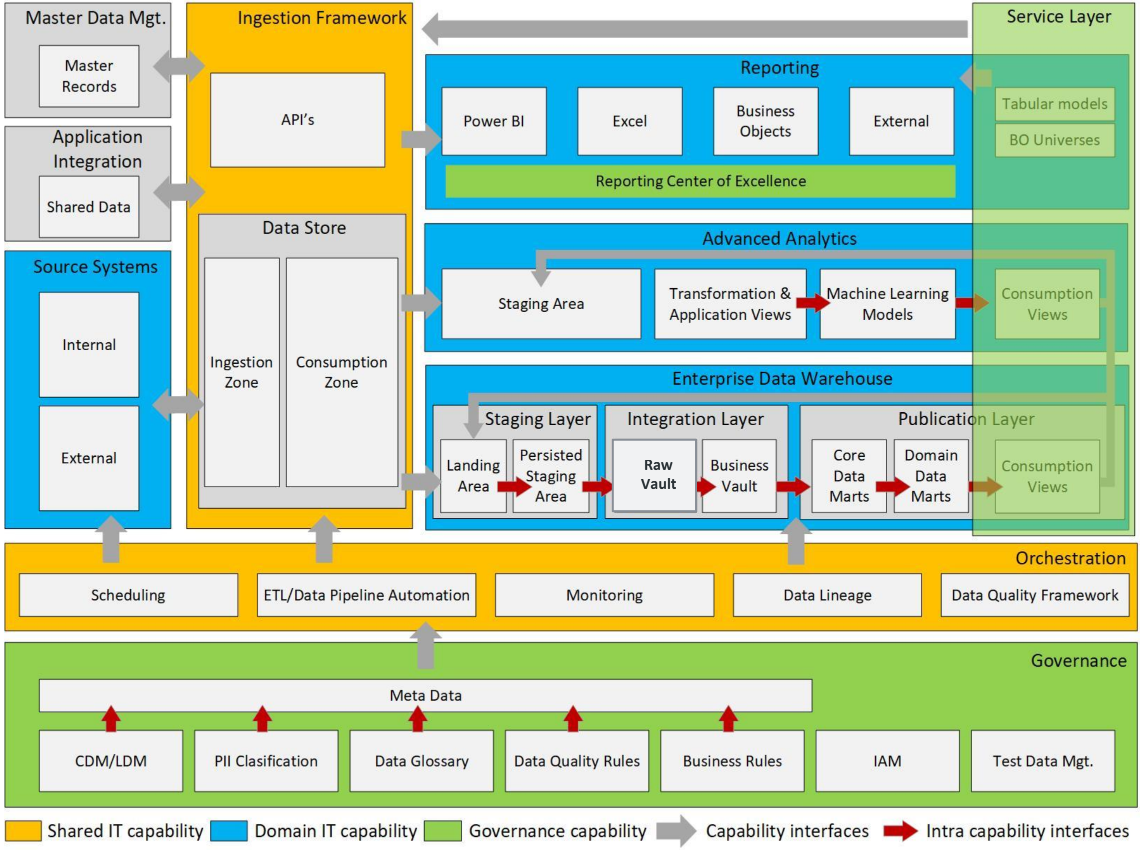
The Enterprise Data Architecture needs to align and support the following principles:

1. We make data fit-for-purpose available at the data source. For this we apply Data Quality at source.
2. We provide rapid, secure access to authorized data.
3. We source our data **once** (ingestion layer) to be consumed many times.
4. We apply **just enough structure** to data to enable Data Consumption (based on purpose).
5. For access we provide IAM (Identity & Access Management) and RBAC (Role Based Access Control).
6. We **empower our users** to make their decisions data driven. For this, we provide BI, Advanced Analytics and Visualization with use of reporting engines such as Power BI.
7. We offer **foundational capabilities** (e.g., Data Modelling) to support data consistency and -coherency in all data activities
8. We bring **strong control** and oversight using the Data Governance capability. We use DDAs (Data Delivery Agreements) between Data Owners and Data Consumers.
9. We drive our data processes (e.g., Operations, Security) **with use of Meta Data**. It is therefore important to build up our meta data.

TFG

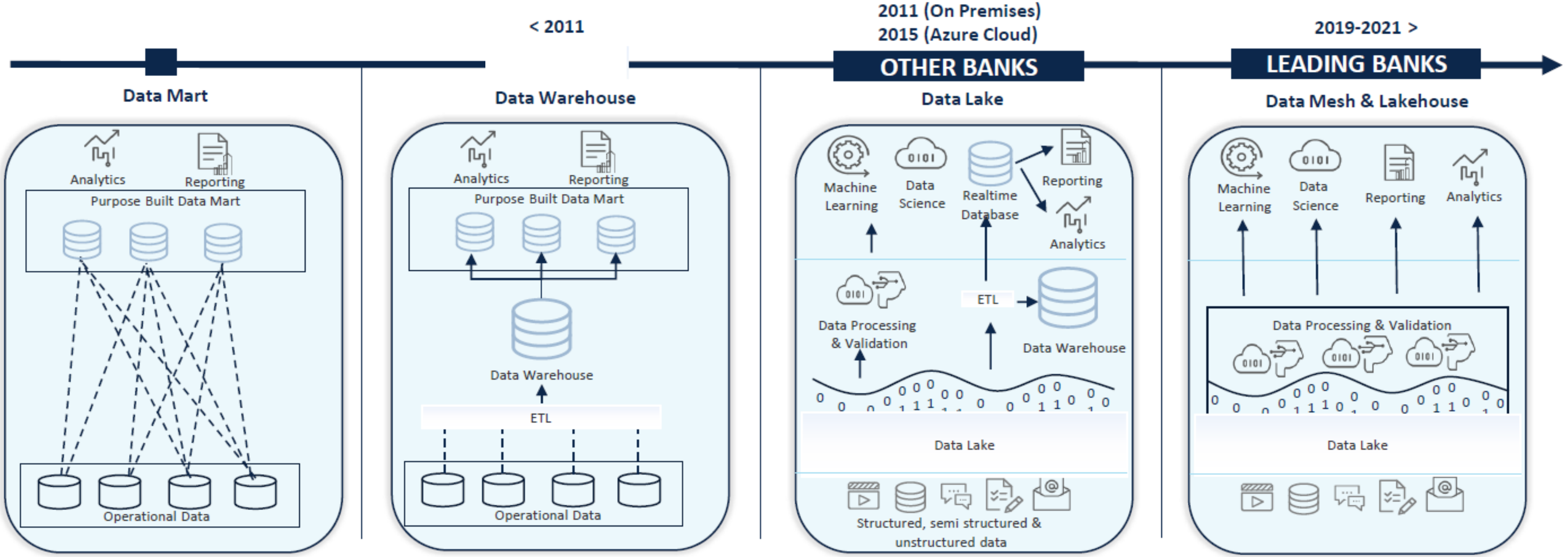
We translate our requirements (DA/DM) and principles into (logical) data capabilities in the EDA.

Capability	Description
Advanced Analytics	Capability where the benefits of advanced analytics concepts such as Machine Learning, AI and Data Discovery can be leveraged.
Application Integration	The process of enabling independently designed applications to work together
Data Sources	Transactional (OLTP systems) that support the operations of the bank.
Enterprise Data Warehouse	Capability where data from multiple data sources **is integrated**, business rules are applied, and a single point of truth is established.
Governance	Managing data availability, usability, integrity and security, based on standards and policies.
Ingestion Framework	Capability responsible for making (source) data available to downstream consuming capabilities such as Advanced Analytics, Application Integration, EDW and Reporting. This capability should ensure that data is sourced and consumed in a secure, consistent, and efficient way.
Master Data Management	Ensure the accuracy, stewardship, uniformity, semantic consistency and accountability of official shared master data assets.”
Orchestration	Automate and **streamline the process of taking siloed** data from multiple data storage locations, combining and organizing it, and making it available for data analysis tools.”
Reporting	All public-, regulatory-, management- and operational reporting activities that take place in enterprise XYZ.
Service Layer	Data layer where data produced by Analytics capabilities can be consumed directly by other capabilities, and directly by end-users. The service layer is highly governed ensuring safety, but still offers seamless availability of data.



The data capabilities are written out in detail in the White Paper; here we only provide a short summary per capability.

Looking back and forward at (solution) Developments



As a part of this architecture

- **Purpose-built database** were built optimized for **analytics and reporting.**
- Source often for these databases were transactional systems having **point to pint connect.**
- This resulted **in enhanced complexity and reduced re-usability.**

As a part of this architecture

- An **additional layer data warehouse** is introduced which is a **database optimized** to analyze relational data coming from transactional systems.
- The **data structure, and schema are defined** in advance to optimize for fast SQL queries,
- Data is cleaned, enriched, and transformed so it can act as the **"single source of truth"** that users can trust

With data sources going beyond structured sources,

- Data Lake is a storage repository that can store **large amount of structured, semi-structured, and unstructured data.**
- It is a place to store every type of data **in its native format.**
- It offers high **data quantity** to **increase analytic performance and native integration**

Data mesh and lakehouses have implementations

- Similar to data warehouse in terms of structures and data management features but runs , **directly on the kind of low-cost storage used for data lakes.**
- Merging them together into a single system means that **data teams can move faster** as they are able to use data **without needing to access multiple systems**

<u>Source:</u> *Accenture / Cloud Architectures*

Obviously we are considering Cloud solutions

1. We look at common cloud scenarios that fit Enterprise XYZ and align with EA-future states. Going for an industry standard ensures adequate support and development resources in the market.
2. The proposed scenarios look at <u>four</u> reference architectures:
   a) Data Lake, Data Lakehouse, Data Mesh and Data Fabric
3. The proposed scenarios look at <u>three</u> cloud technology stacks:
   a) MS Azure, Snowflake and Data Bricks.
4. Current industry standards typically combine reference architectures and technologies.
   a) Note that the technology primarily refers to the key <u>integration</u> tech, not the complete stack. E.g., Data Bricks typically runs on MS Azure using Blob Storage for the Data Lake, same goes for Snowflake.
5. The described scenario's will be input for a cloud data technology selection (RFP) process

*Cost is a key factor when deciding **what cloud architecture and technology** to select. When analyzing costs, we focus on four key elements:*

1. ***Development**: complex technology, that is hard to implement, will result in high costs.*
2. ***Licenses**: annual license costs for the cloud services.*
3. ***Compute**: compute is scalable and flexible with cloud, but also very expensive.*
4. ***Storage**: storage is cheap with cloud*

.

TFG

Cloud Architectures (1) DataLake

We focus on three Data architectures: Data Lake, Data Lakehouse, and Data Mesh. Per architecture we describe its benefits.

Scenario	Benefits	Recommendations
• DataLake (DL) is the architecture that **emerged when big data** and its applications such as Hadoop File System (HDFS) became available to data teams. • The idea is that **all sorts of structured and unstructured data become available** for Analytics and operational processing, this data is than stored centrally in the Data Lake for consumption by an EDW, Machine Learning and AI.	Compared to EDW: • Datasets can be made available for consumption fast since no expensive and time-consuming integration in the EDW is needed. • **No data transformation is needed, data is schema less** which allows data to be made available fast. • **Storage in Data Lake is cheap**, allowing storages of data sets that otherwise would not be included in EDW. • Separation of Storage and Compute, allows for faster ETL at limited extra cost.	• DL is being replaced by DLH and DMe, but is for a smaller enterprise still a good fit. • DL supports easy lift and shift migration of EDW to the cloud and is a good foundation for Advanced Analytics suites such as Dataiku.

TFG

Cloud Architectures (2) DataLakeHouse

We focus on three Data architectures: Data Lake, Data Lakehouse, and Data Mesh. Per architecture we describe its benefits.

Scenario	Benefits	Recommendations
Data Lakehouse (DLH) is the **evolutionary next step from DL**, addressing some of the shortcomings from DL and further leveraging the potential of modern data technologies. • The general idea is that data is **stored only once** on low-cost storage **while subsequent, meta data driven layers are built on the physical layer without actual physical data replication.** • This architecture became available thanks to technologies such as DataBricks and Snowflake. • **Integration takes place in three layers of DLH**, often labeled as Bronze (Raw), Silver (Filtered, Cleaned, Augmented) and Gold (Business level aggregates). • Then all sorts of outputs (e.g., Data Marts) are built on these layers.	Compared to DL: • No physical replication of data. • Out of the box functionality such as time travel, delta load handling, governance. • Strong support for machine learning and AI.	• DLH is also a good fit. It is basically a DL with lots of improvements. The main challenge is that in an ideal situation an existing EDW would be rebuilt based on the principles of a DLH. • A solution here would be to lift and shift the current EDW, based on the replication principles of DLH.

TFG

Cloud Architectures (3) DataMesh

We focus on three Data architectures: Data Lake, Data Lakehouse, and Data Mesh. Per architecture we describe its benefits.

Scenario	Benefits	Recommendation
Data Mesh (DMe) is the most modern data architecture implemented by most large banks and tech firms. DMe is a **decentralized, federated architecture that can be compared to micro services in software architecture.** Two of the core principles of Data Mesh are: (1) Domain ownership of data and (2) Data as a Product. All data has a single authoritative source and domain owner; domains are responsible for their data, and data producers treat their Data as a Product to the outside world (consumers). • Each domain handles its data the way that best fits their functionality, both from an operational and analytics perspective. **However, the way data is exposed to other domains is centrally implemented, regulated and governed.** • This means that all data that is exposed to other domains must be delivered in a re-described format, have meta data descriptions available and be described in a data catalog.	Compared to DL and DLH: • **Domain driven architecture that is all about ownership of data.**	• DMe requires large commitment from domains to provide **meta data** for their data and make it available for external consumption. • DMe is a better fit for large organizations where domains have their own IT capability. • SMBs not large enough **can not commit** to this.

TFG

Technologies, that could be the backbone of our future data cloud platform.

MicroSoft Azure	Databricks	SnowFlake
Azure supports many different programming languages, tools, and frameworks, including both MS-specific and third-party software and systems. Amazon and Google offer similar platforms. The Azure platform can be the base for the Data Architecture based on its core services. Core services (required for all scenarios) e.g. • Azure Analysis Services • Azure Cosmos DB • Azure Purview • Azure SQL • Azure Synapse	**Offers a "lakehouse" platform** that is based on open source Apache Spark framework that allows analytical queries against semi-structured data without a traditional database schema. • Data is physically stored in cheap ADLS gen2, with the Delta Lake as Storage layer on top. • Delta Lake supports SQL and has features like snapshot isolation which helps concurrent read/write operations and enables efficient insert, update, deletes, and rollback capabilities. • Allows background file optimization through compaction and z-order partitioning achieving better performance improvements. • Developers use Python and SQL in Databricks notebooks (much like Jupiter) for creating ETL and Machine Learning workflows, Scala and R also supported in these notebooks. • Its ecosystem includes a Data Catalog "Unity Catalog", that can be used as interface for data discovery. Databricks scenario specific services are: • Databricks Delta Lake, • Databricks Unity Catalog	Is a **cloud native Data Analytics platform** (unlike Synapse that is based on on-premise SQL server technology). It offers many unique benefits compared to traditional Databases. • Data is stored in the background on Azure Blob Storage in so called **micro partitions**, these small units of storage are immutable, **data changes are stored in a meta data layer.** This means that **physical data replication is not needed,** a database can be cloned without any copying of data. Users of a cloned database can then make changes that are only implemented on their database. • Data can only be stored in a cheap and easy way with internal users, **it can also be stored without replication with external users.** Think of vendors who sell datasets or need to share for other reasons. • Micropartitions are stored in multiple clusters that can receive concurrent requests, the location of data is stored in the meta data layer https://docs.snowflake.com/en/user-guide/tables-clustering-micropartitions.html • **Separation of compute and storage**, t-shirt size compute units can be used by end users and systems based on task requirements. This allows for very scalable and efficient allocation of expensive compute resources. • Snowflake optimizes queries and data structures under the hood, that results in ultra-fast performance for large data sets and even for poorly written SQL. • The Variant data type can be used **to store unstructured data** (Parquet, Avro, Json, etc.), this offers great benefits for implementing the data lake in Snowflake. • Snowflake has multiple **out of the box methods and interfaces for data ingestion**, the vert simple "copy in" command extracts data from data lake, external tables can be created on data lake files and multiple connectors exists for Kafka, ALD and ODBC. • **Old school Wherescape** is Snowflake compatible, allowing out of the box deployment of the Data Vault ETL packages. • Delivered as SaaS application, which **does not require any ALM**, upgrading or any other maintenance.

TFG

Then we look at scenario’s

Based on architectures and technologies, we describe three scenarios, which are industry standards and fit for Client XYZ. For the cloud architecture we either use Data Lake or Data Lakehouse and as potential platforms we use Azure Synapse, Databricks or Snowflake.

Scenario #1: Data Lake with ADLS gen2 or Blob and Synapse.	Scenario #2: Data Lake house with MS Azure and Data Bricks.	Scenario #3: Data Lake with MS Azure and Snowflake.
A (cheap) alternative could be to use Azure SQL at the beginning and move to (the more expensive) Synapse when the business case arises. For machine learning we can either use Dataiku or Azure Machine Learning. The Data Vault can be lift and shift migrated to Azure SQL or Synapse.	In this architecture there are typically three layers, the Data Lake (Bronze), the Delta Lake (Silver), plus the analytics products (Gold). These layers are not physical layers, but meta data on top of the physical layer, this keeps storage cost very low. Data Bricks has an integrated Data Catalog, it is geared towards Machine Learning and AI rather than EDW. Migrating the Data Vault goes against the architecture of Databricks.	We opt for either building the Data Lake in ADLS or in Snowflake. In the last scenario the ingestion layer would be in Azure, the consumption Layer would be in Snowflake. Dataiku would be a good Machine learning solution in this scenario.

➔ *Following we describe each of these scenarios based on reference architectures provided by the vendors. There are many variations possible given the tools and concepts available. For our purpose, we focus on scenario #3.*

TFG

SCENARIO #3, Data Lake House with MS-Azure and Snowflake

Snowflake, just as Databricks, claims that it has invented the Data Lake House architecture.

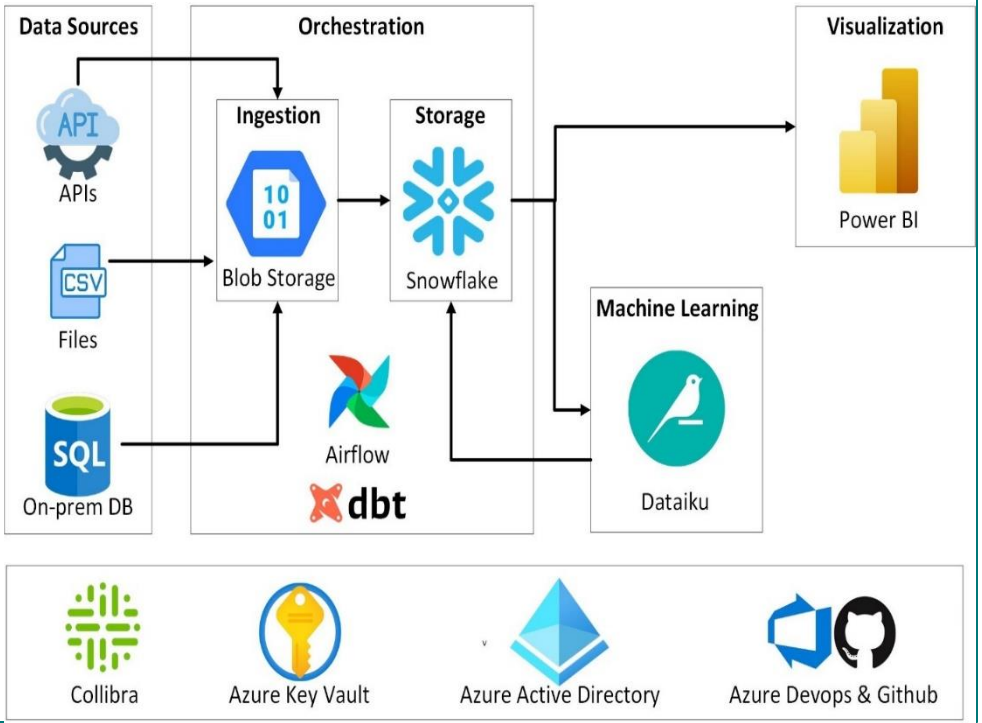
Setup:

- Data is sourced (most likely from a landing zone in Azure) into the Snowflake Data Lake.
- Data is Integrated into EDW with use of Wherescape.
- Data from the Data Lake is consumed for other purposes (e.g., by Dataiku for Analytics)

Pros / Cons:

- SaaS, no Application Lifecycle Management
- Easy to load data into Snowflake and storage is cheap
- Easy sharing of data (internal + external) without replication
- High performance without the need for specialized developers
- Lifts and shifts old legacy out of the box (WhereScape)
- **May be the most expensive solution (see appendix 2). Adds an additional vendor.**

Overview: Snowflake Reference Architecture



TFG

Appendix 1: Descriptions NFRs

More extensive descriptions of Non-Functional Requirements (NFRs)

Requirement	Explanation
Automatability:	We aim to automate as much of our data flow processes as possible, driven by standardized patterns and meta data. However, we should assess for each use case if the process of automating makes sense. It could be the case that developing the automation framework is more expensive that coding on a case-by-case base.
Data Availability:	A general shortcoming of the current data platform is that stakeholders do not have access to the data they need or must wait for long time before they can use the data. Our data platform should support fast access to data for our stakeholders.
Extendibility:	Bank XYZ aims to grow by acquiring new asset portfolios in the market. These portfolios than need to be onboarded quickly in our enterprise.
Governability:	The ability to govern the Data Platform in a secure and efficient way is a critical success factor for the Data Platform.
Simplicity:	Bank XYZ is a small bank with limited development resources. For the EDA we should limit the number of components and technology where possible. New solutions should integrate without complex development effort in the larger data platform. Complex and hard to implement technology should also be avoided whenever possible.
Usability:	The aim is to provide Domain IT teams with services that they can use without extended technical knowledge. The focus for domain teams should be on understanding the data and business of the bank, while the Shared IT teams provide a service that is easy to use, preferably by configuring, rather than coding.

Appendix 2: Costs Comparison Matrix

Cost is a key factor when deciding what cloud architecture and technology to select. When analyzing cost, we should focus on the following four key elements:

1. Development: complex technology that is hard to implement will result is high costs
2. Licenses: annual license cost for the cloud services.
3. Compute: compute is scalable and flexible with cloud, but also very expensive.
4. Storage: storage is cheap with cloud.

Tech stack per scenario Cloud cost factors

	Scenario 1: Synapse	Scenario 2: Databricks	Scenario 3: Snowflake
Data Pipelines	Azure Data Factory	Azure Data Factory	Azure Data Factory or Airflow + DBT
IAM	Azure Active Directory Azure Key Vault	Azure Active Directory Azure Key Vault	Azure Active Directory Azure Key Vault
Ingestion Layer	Azure Blob Storage	Azure Blob Storage	Azure Blob Storage
Data Lake	ADLS gen2	Databricks Delta Lake	Snowflake
Storage	Synapse or Azure SQL	Databricks Delta Lake	Snowflake
EDW ETL	Wherescape	Databricks Workbooks	Wherescape
Analytics	Azure Analysis Services or Datalku	Data Bricks MLflow	Datalku
Reporting	Power BI	Power BI	Power BI
Data Catalog	Microsoft Purview	Databricks Unity Catalog	Collibra

	Synapse	Databricks	Snowflake
Complexity	EUR …		
EDW lift & shift			
Annual cost			

TFG