POLITECNICO DI TORINO

Master's Degree course in Data Science and Engineering

Master's Degree Thesis

Design and Implementation of a Data Governance Framework for Italian Banks

Supervisor
Prof. Paolo Garza

Candidate
Nicola Orecchini

October 2023

Summary

Data Governance is the discipline of formalizing an organization's data assets to treat data as a real valuable resource to use in business decisions. It involves establishing processes, roles, responsibilities, and policies aimed at ensuring that data is managed, maintained, and used effectively and securely throughout the organization. In this thesis, we address this topic through a practical case study of the implementation of a Data Governance Framework for multiple Italian Banks, in which we had the opportunity to actively participate. After having introduced the main theoretical concepts, we dive into the implementation, reporting in a detailed way the practical activities carried out for the project. Due to confidentiality reasons, no real data is provided, but real-life inspired examples are reported throughout the whole study. We conclude our work by presenting some of the possible metrics that can be defined to quantify the impact of data governance for organizations and by suggesting possible future work in this topic.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 What is Data Governance?

It is no longer news that we live in an era where information flows constantly at lightning speed, affording humanity a multitude of opportunities to harness this wealth of information. In this rapidly evolving environment, characterized by a 'Gold Rush' mentality in the pursuit of harnessing the most powerful data exploitation methods, we find it fascinating to explore one of the many processes that typically occurs behind the scenes but is of crucial importance for managing and developing emerging technologies related to data and, more broadly, information.

To introduce the topic we refer to, we start from the concept of *information assets* in organizations. An organization's information assets represent a diverse range of valuable resources that are integral to the organization's operations, decision-making processes, and overall strategic direction. These assets consist of various types of data, knowledge, and intellectual property that collectively contribute to the organization's competitiveness, efficiency,

and innovation. Information assets typically include:

1. **Data Assets**
   - Structured data (e.g., customer info, sales data)
   - Unstructured data (e.g., text, multimedia)
2. **Intellectual Assets**
   - Patents, copyrights, trademarks
   - Trade secrets, proprietary algorithms
3. **Human Capital**
   - Employee skills and experience
   - Insights for decision-making
4. **Organizational Resources**
   - Processes and procedures
   - Systems and applications
   - Models and algorithms
   - Organizational knowledge
   - Networks and relationships
   - Regulatory and compliance data
   - Financial information

Correctly managing the information assets is crucial for the organization to maintain operational excellence, drive innovation, achieve compliance, and sustain the organization's long-term success.

The effective utilization of these information assets is made possible by *information systems*. Information systems are organizations' hardware and software tools to collect, process, and distribute their information assets [1].

Information systems have gained significant importance due to their fundamental role in facilitating data-driven decision-making, optimizing operational efficiency, and enabling the organization to stay competitive in an increasingly digital and interconnected world.

Ensuring the management of the data processed by information systems is thus strategic for the organization to guarantee (1) the accuracy of regulatory reporting and (2) the robustness of their strategic decisions.

The management of data is formalized in the *Data Management* concept, which, in the corporate world, refers to the practice of collecting, organizing, protecting, and storing an organization's data so it can be analyzed for business decisions [3].

Data management activities are oriented towards the final goal of helping decision-makers in their job, i.e. making business decisions. The most common way decision-makers leverage the value of data is through the analysis and evaluation of some Key Performance Indicators (KPI), which are meaningful metrics describing a certain aspect of a business. Very often, different KPIs (which can vary between industries, and even between companies within the same industry) of a business are collected into business dashboards, which are graphical interfaces describing various aspects of a company, or a process of it, through multiple KPIs.

We can thus understand the importance of data management: an improper carrying out of even a small part of one of the data management activities can lead, eventually, to inaccurate calculation of KPIs, misleading the final decisions taken by business users. The effect of inexact KPIs calculation and consequent inexact decision-making is reflected primarily on the business itself (e.g., through quantifiable financial losses caused by inefficient marketing campaigns, or by unreliable predictive machine learning models), but also on the larger environment the business operates in (e.g., on the customers).

For this reason, data management activities must be carried out according to precise and defined processes and rules. Companies need to establish and set up methods, processes, and tools to govern the data they own. In other words, they need to build extensive knowledge about their information assets. This practice is commonly defined as *Data Governance*.

1.2 Data Governance in the Financial Services Industry

The need for data governance is especially evident in the Financial Services Industry. Indeed, in this field, the "productive sector" is starting to coincide more and more with the corporate information system, especially in relation to the emergence of the so-called *synthesis systems*, which are information systems encompassing both accounting-administrative systems and management-directive systems.

Indeed, these systems are used by financial institutions to generate reports, both for internal and external use (i.e. supervisory reports). These reports include specific KPIs like bank rating scores (useful for customers to know how safe it is to do business with a particular bank), or Economic Capital (the amount of capital a financial institution needs to ensure that it stays solvent given its risk profile [2]), and their incorrect calculation can, in turn, result in significant financial losses extendable to a vast part of the global markets. Take for example the 2008 financial crisis, for which poor risk data management was cited as one of the triggers [4].

For these reasons, in the financial services industry, many regulations like [4] explicitly oblige financial institutions to adhere to specific regulations regarding the governance of data. Thus, in this field, the need for data governance is stronger, as among its implementation drivers there is regulatory

compliance.

1.3 Purpose of this Thesis and Structure of the Content

1.3.1 Purpose

When we first became aware of the topic of Data Governance, we researched the web for more information to explore it. Most of the found resources, though, are confined to a very theoretical, conceptual dimension, and present Data Governance as more of a set of best practices, recommendations, or rules for companies to achieve a higher level of Data Culture within the organization.

While the latter is certainly true, Data Governance needs way more than words to be put into existence in the real world: it needs tangible assets, real processes, and tools, along with clear documentation on how each of these must be used. Data Governance requires a deep understanding of the organization's data landscape, its specific needs and challenges, and its alignment with the overall business strategy. What are the activities that have to be carried out in an organization to establish Data Governance? What tools, people, and processes are involved? What are the final deliverables of these activities and how do they provide value to the organization?

In order to address these questions and enhance the understanding of the subject matter for this thesis, we had the valuable chance to actively engage in a practical Data Governance Framework implementation project. The project was conducted for several Italian Banks and encompassed both the design and implementation phases. In the thesis, thus, we report and explain

the project's real activities in as much detail as confidentiality permits, providing firstly the theoretical concepts underlying the various activities, and then moving to the description of the real activity.

The purpose of this thesis is thus that of researching and documenting the field of Data Governance starting from a real-world experience and trying to join the practical activities carried out in the real project with the underlying theoretical concepts, which we retrieved from resources like company documentation or the web, aiming to provide a guide as exhaustive as possible for designing a real Data Governance Framework and implementing it, starting from the case experience of the Financial Industry.

We note that, previously, other scholars have addressed the very same topic in a thesis, like [6] and [7]. More in particular, both works focus on the organizational point of view of data governance, analysing specific processes and roles/responsibilities needed to set up a data governance framework in an organization. We note how these two works are presented as theses for Master's Degrees respectively in International Management and Management Engineering. Thus, the more organizational point of view and lack of technical details of these works can be explained.

In this context, therefore, our aim is to fill the gap in technical details. Indeed, during our Master's Degree in Data Science and Engineering at Politecnico di Torino, we addressed extensively the topics that serve as a foundation for data governance activities. Thus, in our work, we try to use our acquired knowledge to shed light on data governance from the technical side.

1.3.2 Structure

The structure of the thesis reflects that of a scientific paper: Introduction, Related Work, Methodology, Results, and Conclusions. We made this choice

because this experimental structure allows us to have an exhaustive discussion of the topic, covering both the theoretical aspects and the practical ones. In particular, the Theoretical Background and Related Work will be presented in Chapter 2, the Methodology in Chapter 3, the Results and Analysis in Chapter 4, and the Conclusion in Chapter 5.

In Chapter 2, we discuss what Data Governance is, and what are its objectives, and we present some theoretical concepts needed to formalize the implementation chapter.

In Chapter 3, we address the topic of how to decline theory into practice by designing and implementing a Data Governance Framework for Italian banks. As we anticipated, we do this by reporting the phases of a real project in which we had the honor to participate.

In Chapter 4, we present the results of the Data Governance Framework implementation project, discussing the challenges encountered during its implementation and how our team addressed them. We also analyze the broader impact of Data Governance implementation on the bank's operations, data culture, and decision-making processes, and try to provide insights and recommendations for future Data Governance initiatives within the organization or similar institutions.

Finally, in Chapter 5, we summarize the project and the key concepts that emerged.

Chapter 2

Literature Review

In this chapter, we review some literature about Data Governance in order to exhaustively introduce all the concepts that will be used in the implementation part, i.e. in Chapter 3. More specifically, we examine the literature from the two following points of view.

1. Literature about Data Governance: here we present the concept of Data Governance and how it translates into a real-world application, by reviewing specifically the DAMA DMBOK [5], the most popular manual for Data Management to date.

2. Foundational concepts about Data Modeling: here we go through the basic concepts of data modeling, which are needed to understand a major part of the implementation phase.

2.1 Data Governance

2.1.1 Data Governance Definition and Objective

To understand the concept of Data Governance, we first need to introduce the concept of Data Management. The Data Management Association (DAMA) International [5] defines Data Management as the "development, execution, and supervision of plans, policies, programs, and practices that deliver, control, protect, and enhance the value of data and information assets throughout their lifecycles".

On the other hand, DAMA defines Data Governance as the *"exercise of authority and control (planning, monitoring, and enforcement) over the management of data assets"*.

So, let us make clear the difference between the two: while Data Management refers to any activity related to extracting value from data, Data Governance is one of these specific activities, with the peculiar function of overseeing all the other activities. By *overseeing* we mean, as stated in the DAMA definition, the practices of planning, monitoring, and enforcement, which, in other words, are practices aimed at the systematic management of data assets, ensuring their quality, availability, usability, and security throughout their lifecycle.

The objective of Data Governance is to establish a framework for decision-making processes, responsibilities, and accountability concerning data-related matters. This encompasses creating policies, defining data standards, and implementing procedures that guide data usage, storage, and sharing practices across an organization. Moreover, it involves collaboration between various departments to ensure alignment with business objectives and regulatory requirements. [8]

Ultimately, we can say that Data Governance guarantees that data is treated as a valuable organizational asset, enhancing data-driven decision-making while preserving transparency and mitigating risks associated with data misuse.

In Fig. 2.1, we report a diagram summarizing the concepts we just introduced (ignore Data Quality Management for now, which will be introduced later in this work).

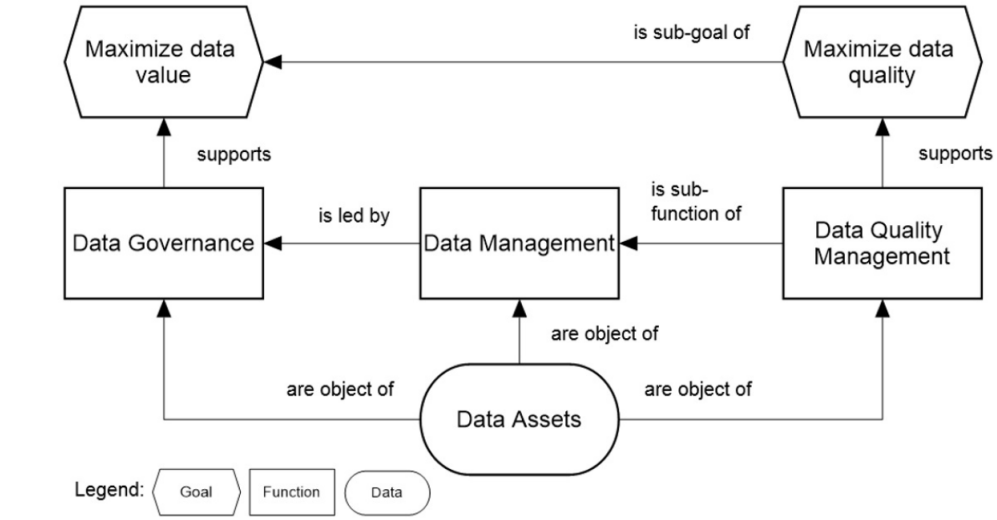

Figure 2.1. Fundamental concepts of Data Governance, Data Management and Data Quality Management. Source: [10]).

2.1.2 Data Governance Framework

After having understood the main theoretical concepts behind Data Governance, we need to translate it into the business world, by transitioning to a more practical, methodological perspective, with the goal of being able to implement the concept in real organizations.

To do so, we can move from the concept of Data Governance to that of a Data Governance Framework: a Data Governance Framework is composed of a set of rules, procedures, and processes designed to enhance the value of data within a business organization, and is responsible for providing a holistic approach to data collection, description, management, protection, and storage [13]. From this perspective, Data Governance is a business function aimed at defining and implementing an all-encompassing strategy for managing data within an organization [12].

In other words, the Data Governance Framework is the mean to implement Data Governance in organizations.

Several models to design a Data Governance Framework have been proposed in the literature [14]. In our work, we analyze that of DAMA, presented in the DAMA DMBOK [5].

DAMA International (Data Management Association) is a not-for-profit, vendor-independent, global association of technical and business professionals focused on advancing the field of information and data management [15]. The organization provides a platform for data management professionals to share knowledge, best practices, and insights related to data management and governance. The DMBOK (Data Management Body of Knowledge) is a publication of DAMA which provides a structured approach to understanding and implementing various aspects of data management. Its goal is to constitute a reference for practitioners looking to develop and enhance their data management capabilities. The DAMA DMBOK is periodically updated to reflect advancements and changes in the field of data management.

The DAMA DMBOK provides a model for designing a Data Governance Framework, represented in the form of the graphic reported in Fig. 2.2. Data Governance is here seen as a hub composed of sections that encompass various knowledge areas, procedures, and methodologies that constitute the

basis for information management in all of its aspects.

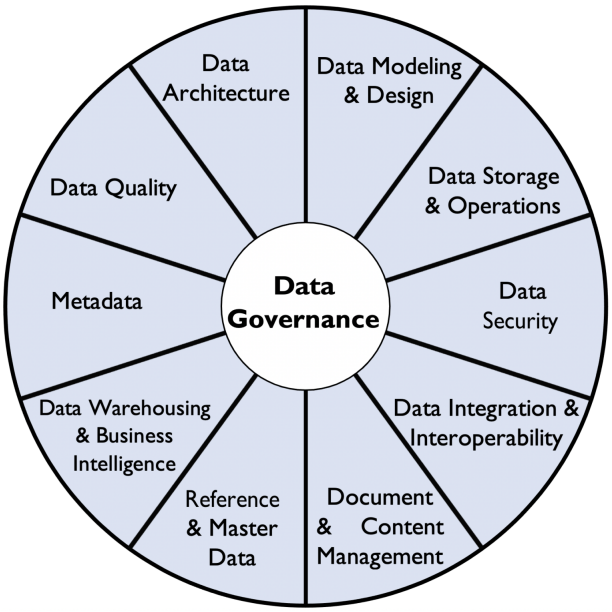

Figure 2.2. Hub Model for Data Governance Framework (Source: DAMA BOOK [5]).

Each knowledge area analyses some fundamental aspects related to data.

1. **Data Modelling and Design**: The process of discovering, analyzing, representing, and communicating data requirements in a precise form, called the data model.
2. **Database Storage & Operations**: Design, implementation, and management of data storage and archiving structures.
3. **Data Security**: Definition, development, and implementation of policies and procedures to ensure proper user authentication, access authorization, and data auditing.
4. **Data Integration & Interoperability**: Processes related to the movement and consolidation of data within and between data stores, applications, and organizations.

5. **Document & Content Management**: Planning, implementation, and control of processes related to data and information of any type and format.
6. **Reference & Master Data Management**: Shared data management aimed at achieving business objectives by minimizing risks associated with redundancies, ensuring quality, and reducing integration costs.
7. **DWH & BI**: Planning, implementation, and control of data production processes aimed at decision support, and assisting knowledge workers in query, dashboard, report, and analysis processing.
8. **Metadata**: Planning, implementation, and control of processes related to metadata.
9. **Data Quality**: Planning, implementation, and control of processes to ensure data quality that aligns with its purpose and user needs, enabling constant monitoring and continuous improvement.
10. **Data Architecture**: Planning the management of data assets, aligning it with the company's strategies, with the aim of defining global requirements and designing solutions to meet them.

This model serves as a starting point for the design of a Data Governance Framework in any organization. The actual design phase should go through the following steps.

1. As a first step, the organization should assess the current governance situation in each of the 10 knowledge areas. Note that each organization is different and thus should, when assessing the current landscape, take into account their challenges concerning data (whether they are related to reporting, data quality, data access, ...) as well as the overall business goals. This means that covering all of the 10 knowledge areas is not mandatory: understanding which knowledge areas should be analyzed depends on the organization.

2. After having identified the meaningful knowledge areas, a prioritization step is required: the organization should identify which knowledge areas they should focus on immediately based on the impact on the overall business strategy and requirements, and the ones to be left for later implementation.

3. As a final step in this framework design part, the organization should start setting down some initiatives for each of the identified knowledge areas, aimed at establishing knowledge about that specific aspect of data, that will eventually constitute the core of the Data Governance of the organization.

We further remark how there is not a single way organizations can build their Data Governance framework starting from the DAMA Hub model: this model proposes a set of knowledge areas to take into account while approaching the framework design phase. Organizations can identify the required activities starting from the indicated knowledge areas, and then expand the list to company-specific ones that could emerge, possibly even outside the 10 presented. Still, the power and importance of the DAMA Hub model for Data Governance lies in its ability to guide from scratch any kind of organization to the realization of a first draft of its Data Governance Framework.

After the design phase of the Data Governance Framework, the implementation phase takes place, which consists of executing the plan by implementing the specific practices, processes, and technologies within each knowledge area. A phase of measurement and improvement follows, where the organization establishes metrics and KPIs to measure the effectiveness of the Data Governance initiatives in each knowledge area, with the possibility of gradually extending the scope of the activities. Following that, a phase of integration takes place: the knowledge areas could often be interconnected, so changes or improvements in one area can impact others, so alignment and coordination should be ensured. Lastly, a communication phase allows to educate employees and stakeholders about the importance of each knowledge area and how

it contributes to overall Data Governance.

We find it interesting to mention here one of the concepts behind this model that makes it so powerful: the *80/20 rule*, also known as the *Pareto principle*. This rule refers to the fact that for many outcomes, roughly 80% of consequences come from 20% of causes [16]. In our case, the Hub model covers 20% of aspects related to data that are able to capture 80% of the initiatives useful to establish a Data Governance framework.

2.2 Data Modelling Fundamentals

As from the DAMA DMBOK, Data Modeling is the process of collecting requirements about the data landscape in an organization, and then representing and communicating them through what is called a Data Model. Data models allow an organization to fully understand its data assets and thus are a crucial tool to set up when designing a Data Governance Framework. In the literature, several data models have been proposed, among which we mention Relational, Object-Oriented, and NOSql.

A data model can be thought of as an architectural blueprint (or a "floor plant") of the data landscape of an organization. Indeed, a data model is usually represented as a visual document containing several diagrams employing standardized symbols to convey information effectively.

A data model is a fundamental tool for activities that need a "view from the top" of the logic of data inside the organization, for example, communication during various business projects, where a common vocabulary about systems and data is required, or during IT projects such as maintenance/upgrade/ replacement of an application, where the starting point is the assessment of the "as is" situation.

The data model constitutes a crucial asset for a Data Governance Framework, as it allows the organization to have a holistic view of the structure of their data, thus allowing them to better understand their data assets.

In the context of our project, the data model serves two goals:

1. Formalizing the architecture of the data assets of the organization, providing a basis for initiating further activities of the data governance framework (e.g., the Data Lineage).
2. Formalizing the architecture of specific deliverables of the data governance framework (e.g., business glossary, data controls). Indeed, we remark how the final goal of the data governance framework is that of designing and building a data governance database, which is a real database, like the ones we started from.

Firstly, we go through the fundamentals of data models.

2.2.1 Levels of Detail

In 1975, SPARC (Standards Planning and Requirements Committee) of the American National Standards Institute (ANSI) introduced the three-schema approach to database management. This framework comprises:

- Conceptual Schema: the representation of the real-world aspects of the organization within the database
- External Schemas: smaller subsets of the total enterprise database system where business users work, according to their specific needs.
- Internal Schema: the machine-oriented depiction of data (how the enterprise's data is stored and organized in the system).

These three levels, in practice, are known as conceptual, logical, and physical layers of detail, respectively. In project contexts, conceptual and logical data modeling fall under requirements planning and analysis, while physical data modeling constitutes a design-oriented activity.

1. **Conceptual** In the conceptual data model, the primary focus is the high-level design of the database. Here, business concepts are captured and expressed through entities and relationships, which we will analyze in the next paragraphs. We report an example in Fig. 2.3.

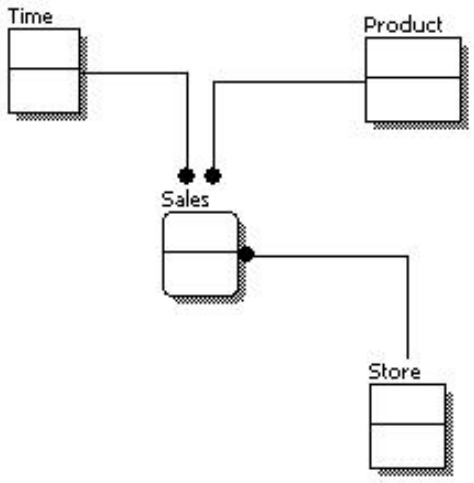

Figure 2.3. Representation of a data model at the Conceptual Level of Detail. Source: [18]

2. **Logical** The logical model comes after the conceptual modeling, with the explicit definition of what the columns in each table are. While designing the logical model, the database system to be used might be taken into consideration, but only if it affects the design (e.g., removing duplicated columns). We report an example in Fig. 2.4.

3. **Physical** Finally, a physical data model portrays a comprehensive technical resolution, derived from the logical data model and subsequently

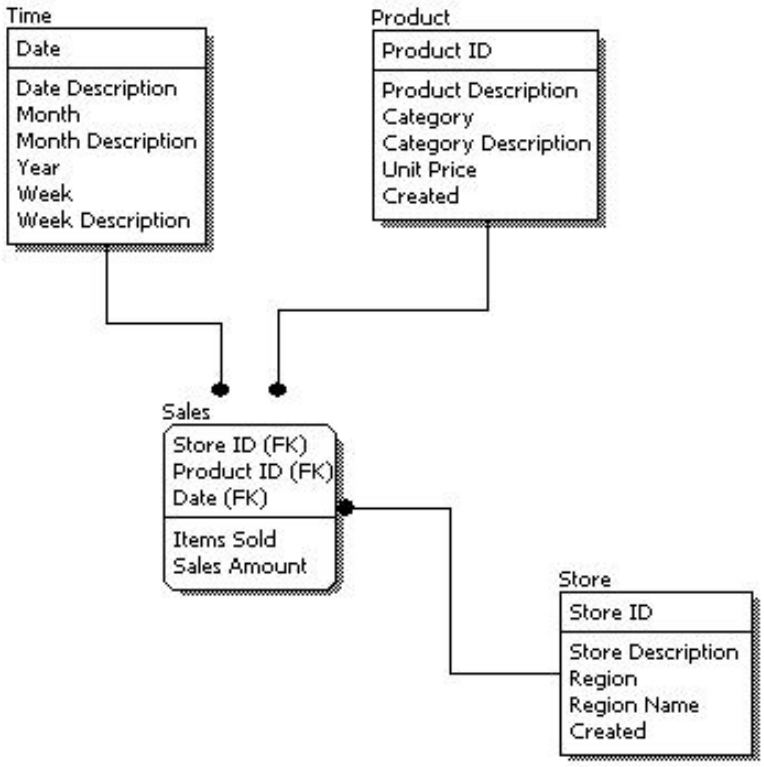

Figure 2.4. Representation of a data model at the Logical Level of Detail. Source: [18]

customized to align with a specific combination of hardware, software, and network utilities. In this level of detail, the characteristics of the columns are made explicit. We report an example in Fig. 2.5.

2.2.2 Components

While different data models represent data through different concepts, there are some basic building blocks common to each: entities, relationships, attributes, and domains.

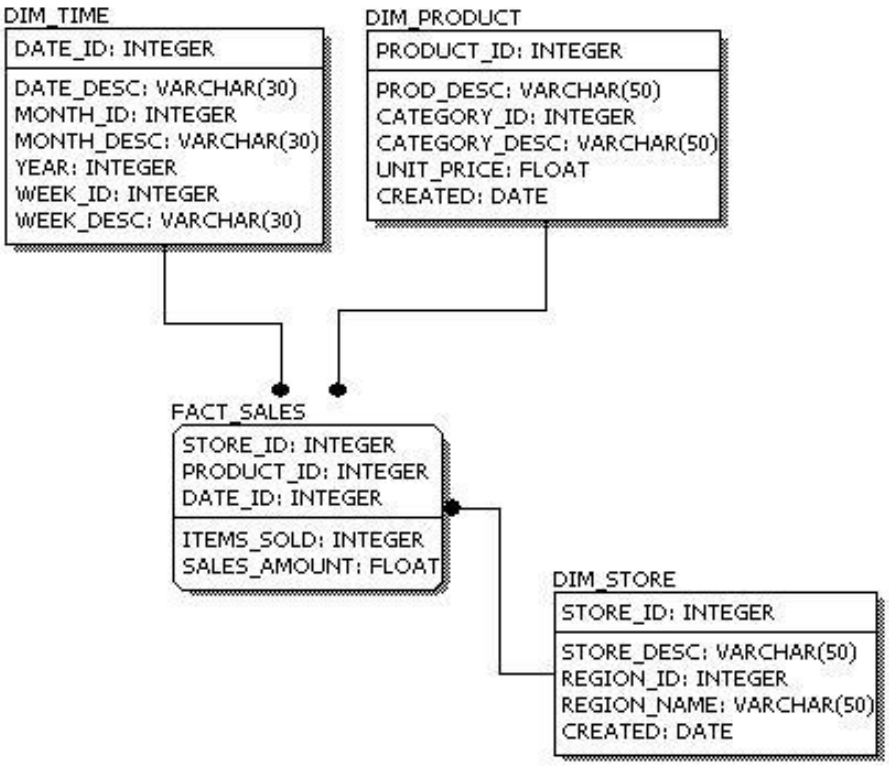

Figure 2.5. Representation of a data model at the Physical Level of Detail. Source: [18]

1. **Entities** Entities are everything an organization collects information about. They often answer questions like who, when, where, what, why, how, and how much. An example is reported in Table 2.1.

   At the physical level, entities correspond to tables, but this might vary based on the data model (for example, in the object-oriented model, entities are represented by classes or objects). The single occurrences of an entity are referred to as instances, and an entity can have multiple instances. For example, the entity Employee can have as instances the names of the employees of a company. In data models, entities are generally represented as rectangles with their names inside, as shown in Fig. 2.6.

2. **Relationships** A relationship is a link between entities, and it expresses

Question	Explanation	Corresponding entity
Who?	Represents the name of a person or an organization.	Employee, Customer, Supplier
When?	Represents a point in time.	Order Date, Event Time, Birthdate
Where?	Represents a location or place.	Address, Destination, Warehouse Location
What?	Represents a thing, object, or concept important to the business.	Product, Idea, Project, Raw Material
Why?	Represents the business reason or purpose behind an event.	Payment, Return, Deposit, Claim
How?	Represents the manner or method of doing something.	Invoice, Contract, Cash Payment
How much?	Represents a quantity or numerical value.	Price, Item Quantity, Due Amount

Table 2.1. Example of entities.

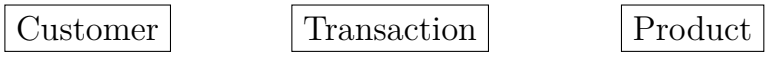

Figure 2.6. Representation of 3 entities.

the interaction between them. Relationships are represented on the data model as lines connecting entities, possibly including a word (usually a verb) that explains the kind of connection. To be exhaustive, a relationship needs another component: cardinality. The cardinality of a relationship quantifies how many entity instances of one entity interact with how many of the other, and this can be: zero, one, or many (zero is used to express possible non-mandatory relationships). Each side of the relationship contains a cardinality in the form $X : Y$, where X, Y is any combination of the possible values that cardinality can take. We report an example in Fig 2.7.

Explanation: (note that, in this example, we refer to a product not as a

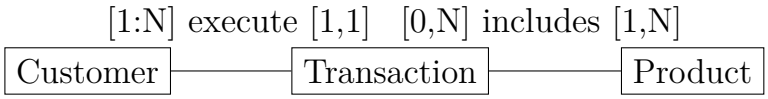

Figure 2.7. Representation of 3 entities with relationships. We indicate zero, one, and many as 0,1,N.

unique piece, but rather as a generic kind of product, like *baseball hat*, or *mouse*)

- **Customer:Transaction 1:N** A Customer can execute many transactions, so the right part must be N. A customer is required in this relationship, because a transaction needs a customer, in order to exist, so the left part is 1.
- **Transaction:Customer 1:1** A specific transaction is made only by 1 customer, so the right part is 1. A transaction is required in this relationship because a customer exists in the database only if they purchased something, otherwise, they would not be a customer, so the left part is also 1.
- **Transaction:Product 0:N** A transaction can include many products, so the right side is N. But a transaction is not required in this relationship, because a product can exist even if nobody purchased it, so the left side is set to 0.
- **Product:Transaction 1:N** A product can be in many transactions, so the right side is N. A product is also needed in the relationship because a transaction cannot exist if it does not contain any product, so the left side is 1.

At the physical and logical level, relationships are made possible with Foreign Keys. A key, in a data model, is an identifier of a specific instance of an entity. In most data models, for example in the Relational, keys are represented with the name of the entity followed by *ID*, inside the rectangle containing the entity, under the name of the entity, and separated by a line. We report an example in Fig. 2.8, indicating the

foreign keys with *FK*. Note that foreign keys appear on the "many" side of the relationship.

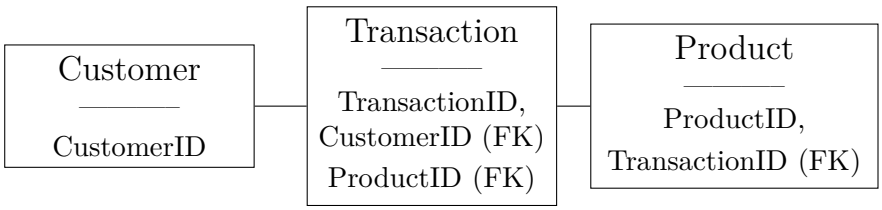

Figure 2.8. Representation of 3 entities with keys and foreign keys (FK).

3. **Attributes** Attributes are properties that characterize or measure entities. The physical correspondent of attributes are columns, in the relational world, or fields, in the NOSql one. Attributes are represented in the data model as a list in the entity rectangle, after the keys. In Fig. 2.9.

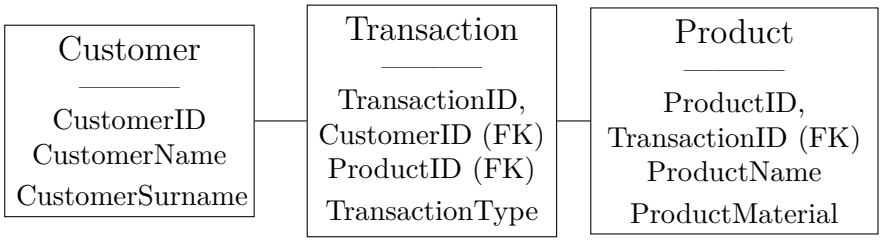

Figure 2.9. Representation of 3 entities with attributes.

4. **Domain** Domain represents the complete set of values an attribute can assume. Domains are important for the standardization of the characteristics of attributes. For example, we could have a domain Date containing all possible values a date attribute can take, independently of what entity that attribute belongs to (for example we could have Purchase date, Birth date, etc.). All values inside the domain are considered valid; the others are considered invalid. Domains can be further restricted by applying business logic: for example, the date of payment of the second installment must be subsequent to the payment of the first.

Domains represent an important aspect and will be useful in the Data Quality part. Domains can be defined in different ways.

- Data type: the standard data type of an attribute. Example: Character(10) indicates the attribute can only contain values formed by at most 10 characters.
- Data Format: patterns of an attribute, for example, the postal code may contain a restricted set of characters.
- List of Values: the finite (and usually relatively small) set of values an attribute can take, for example, Order Status: Shipped, Canceled, Delivered.
- Other logic: business-specific logic, for example, the Order Delivery Date must be subsequent to the Order Shipping Date.

After having discussed the main components of a data model, in Table 2.2 we join this information with what was previously presented in the Levels of Detail paragraph, analyzing the use of the various components in the Conceptual, Logical, and Physical levels of detail for a Relational database data model.

Component	**Conceptual**	**Logical**	**Physical**
Entity Names	X	X	
Entity Relationships	X	X	
Attributes		X	
Primary Keys		X	X
Foreign Keys		X	X
Table Names			X
Column Names			X
Column Data Types			X

Table 2.2. Comparison of data model components in Conceptual, Logical, and Physical Data Models. An "X" indicates that the component is used in the data model at that specific level of detail.

2.2.3 Schemes

The most used schemes to represent data are Relational, Dimensional, Object-Oriented, Fact-Based, Time-Based, and NoSQL. Here we discuss the Relational and the Dimensional schemes, focusing on the differences between them, as they are the two data models that we encounter in our project when dealing with bank institutes' data architectures.

- **Relational**: Relational scheme focuses on reducing the redundancy of storage. This is achieved by having one fact in one single place: each type of information, such as a customer name, is stored in one unique location called a *table*. All the other customer details, e.g. sales dates or purchased products, will each have dedicated tables. These tables will establish *relationships* with the main customer table, eliminating the need to repeatedly input the customer's name whenever new data is appended. So, here, relationships capture the business rules connecting facts. This kind of scheme provides an efficient way to capture business transactions and is mostly used for Online Transaction Processing (OLTP) databases, like Transactional or Operational databases.

- **Dimensional**: Dimensional scheme focuses on structuring data to optimize querying and analysis. This is achieved by using two concepts: *fact* and *dimension* tables. Fact tables are composed of rows representing specific numeric measurements, such as amounts, quantities, or counts, and take usually roughly 90% of the database space [5]. Dimension tables consist mostly of textual explanations representing business objects, and are the main references for answering *query by* questions about the fact tables. They usually take roughly 10% of the database space. This kind of scheme is designed for business reporting purposes and is mostly used for Online Analytical Processing(OLAP) databases, like Reporting databases.

We report a quote that effectively summarizes the main difference between Relational and Dimensional schemes: *"Relational database models are optimized for getting your data in, while dimensional database models are optimized for getting your information out"* [17]. Understanding these two types of schemes is needed in the context of our work as they are the most used by banking institutes.

2.2.4 Data Modeling

Data modeling refers to the process of creating a data model for a business. It involves activities aimed at evaluating an organization's requirements in order to apply the concepts we just introduced. We do not go into details here but only mention the two ways in which the data modeling can be executed: forward engineering and reverse engineering.

- **Forward Engineering** is the creation of a database starting from the specified requirements. The initial step regards the design of the Conceptual Data Model, which aids in comprehending the project's extent and the essential vocabulary associated with it. Subsequently, the Logical Data Model is established to document the business-oriented solution, succeeded by the Physical Data Model to present and document the technical solution.
- **Reverse Engineering** is the comprehensive documentation of an already existing database. The initial stage involves the writing of the Physical Data Model, enabling the understanding of the technical blueprint of the current system. This is succeeded by the elaboration of the Logical Data Model, which outlines the business-oriented solution that the existing system fulfills. Subsequently, the Conceptual Data Model is written to capture the business terminology within the established system. Note that while various data modeling tools facilitate the process of reverse

engineering from a range of databases, the creation of a coherent layout for model elements, especially for the Conceptual Data Model, still necessitates the involvement of a manual modeler, which executes tasks including grouping of entities based on subject area or function.

Reverse Engineering is indeed the technique used by our team to understand the organization's data assets, where existing documentation about the data model is not available.

Chapter 3

Methodology

In Chapter 2, we introduced the concept of the Data Governance Framework and presented one of the many possible models to realize it, specifically the DAMA Hub Model. In this chapter, we present the model specifically implemented for the project, which we will see involves the DAMA Hub Model while enriching it through specific techniques.

Before moving to the presentation of our approach, we report the problem statement of the project: our client is represented by multiple Italian banks, which, after the issuance of some regulations (Bcbs 239 and Circolare 285 of Bank of Italy), found themselves in having to make their information systems compliant to specific constraints, including having proper documentation and knowledge of their data assets. Our job is then represented by developing a framework to help them solve this problem, i.e., bringing their information systems to regulatory compliance.

When addressing this problem, we emphasize that while the ultimate objective is to achieve regulatory compliance, we do not confine the project's scope solely to this aspect, but rather try to establish a high-level framework that

enables the enhancement of information systems management in general, and which, as a side effect, can ensure their compliance with regulations. This approach, as we will see, ensures (1) more long-term oriented benefits and (2) benefits outside the compliance scope.

Moving to the presentation of our approach, we start with a graphical representation of it in Fig. 3.1. The subject of the problem is represented by the organization's *information assets*, including, as we say in the introduction, various types of data, knowledge, and intellectual property that collectively contribute to the organization's competitiveness, efficiency, and innovation.

Organizations can better manage their information assets by establishing processes, policies, standards, and frameworks that guide the overall management of data across the organization, which, as we have seen in Chapter 2, is exactly the definition of data governance. Thus, as we report in Fig. 3.1, *organization's information assets are enhanced by Data Governance*. We have also seen how data governance is applied specifically through a data governance framework, as we report on the diagram.

Continuing in Fig. 3.1 from the *Data Governance Framework* node, we have seen how the DAMA Hub Model provides a basic framework, based on 10 knowledge areas, to identify areas of the organization to be analyzed in order to establish knowledge about information assets. Thus, in the figure, we report this relationship between the data governance framework and the DAMA hub Model.

Here, we note how a relationship can be established also from the *Organization's information assets* node and the DAMA Hub Model: in this context, in fact, the DAMA Hub Model serves as an approximation of all the possible sources of information assets in an organization. Indeed, in the real world, every organization is different and has its own information assets, but still, as we mentioned, the DAMA Hub Model identifies the main areas where the

majority of information assets are contained, independently of the organization.

From here, the methodology part of the project begins: the data governance framework that we would like to design and implement in the project comprises tools aimed at analyzing, or better, building knowledge about the organization's information assets. More specifically, we see how the indicated tools are *Data Discovery*, *Metadata Management*, and *Data Quality Management*. Even if these tools present names similar to some knowledge areas of the DAMA Hub Model, we remark how these are actual tools, that serve to build knowledge about the corresponding DAMA Hub knowledge areas, but also to build knowledge about all the other areas, as we will see.

In the diagram, we report the principal deliverables produced by the data governance framework's tools too: *Data Model*, *Data Lineage*, *Business glossary*, *Data Dictionary* and *Data Controls*.

Finally, we report in the diagram the last part of the project: collecting all the project deliverables into one single structure, represented by a real database, called *Organization's Data Governance Database*, which constitutes the principal source of knowledge about the organization's information assets.

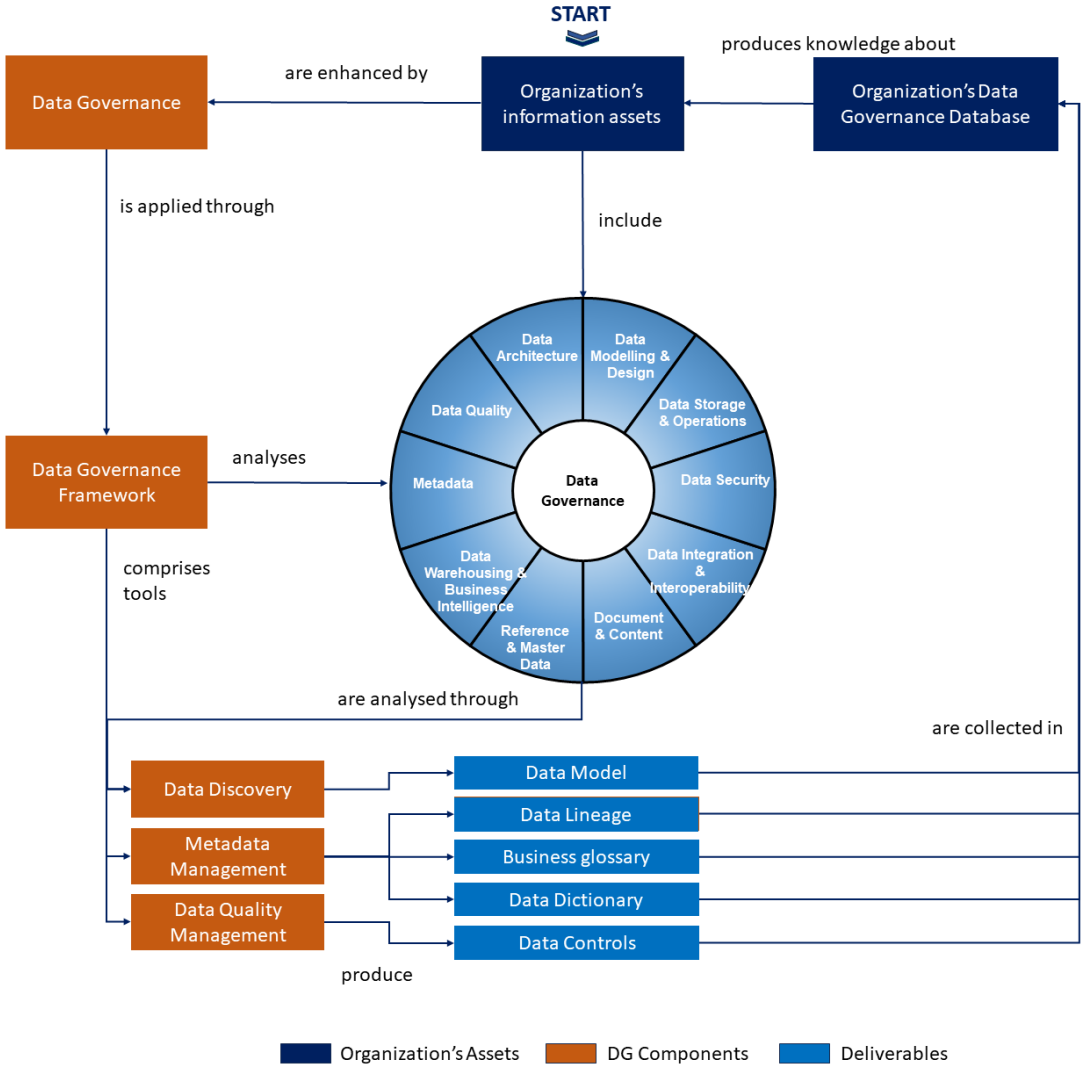

Figure 3.1. Flow diagram illustrating the scope of the project.

This chapter, thus, is structured following the flow diagram presented for the project structure: we will go through all the three tools needed to build the organization's data governance database: *Data Discovery*, *Metadata Management* and *Data Quality Management*. For each tool, we will go through 3 sections: Definition, Technique, and Deliverables. In doing so, we will both

reference the DAMA DMBOK [5] and report knowledge taken from the real project experience. For confidentiality reasons, no real data will be provided, but instead, we will use example data to give an idea about the concepts.

3.1 Data Discovery

3.1.1 Definition & Objective

The first tool the project relies on is Data Discovery. It can be described as the process of comprehending an organization's data assets. This involves discovering which databases, applications, or business processes the organization owns, with the aim of creating a map of the whole landscape.

The objective of data discovery, in the context of the implementation of a data governance framework, is identifying all data assets within the organization, which is essential for initiating any data governance activity. More specifically, data discovery makes possible the two subsequent activities in the implementation of the data governance framework: metadata management and data quality management.

Even if details are discussed in the two following sections, we anticipate how data discovery makes these two steps possible:

- **Metadata management**: the big picture of the organization's data assets constitutes a first form of data lineage, which enables the setting up of a higher granularity level data lineage.
- **Data quality management**: having a high-level representation of all the information assets allows us to easily identify grafting points for data quality controls, making sure they cover the most relevant systems.

3.1.2 Technique

Data discovery involves activities like the examination of *data models*, which are a core concept of the Data Modelling and Design knowledge area of the DAMA DMBOK we have seen in Chapter 2. More specifically, data discovery involves:

i. **Initial stakeholder engagement**: preliminary discussion with key people owning information assets, like Chief Data Officers (CDO), or database administrators, aimed at identifying the subject of study of the data governance activities.

ii. **Gathering documentation**: after having identified the information assets, the next step is to gather any existing documentation related to them. This can include data models, data architecture representations, and any other relevant documents or files, e.g. sample extractions from the database. This documentation aims to provide context on how data is structured, stored, and utilized within the organization.

iii. **Data relationship mapping**: this step involves understanding how different data elements are related. Data relationship mapping involves identifying dependencies between systems and applications, such as data flows between systems or transformations they go through during various processes. This step aims to comprehend the data ecosystem and its interconnections.

iv. **Data landscape visualization and presentation to stakeholders**: this is the last step, and involves the creation of a visual representation of the data architecture studied, illustrating the principal systems and processes producing and using data in the organization. The graphic representation is then shared with the key stakeholders of the organization, and here they can confirm the architecture and approve the project, if the architecture is exhaustive and correct, or, if some integration of the

architecture is needed, they give indications on which corrections to be applied.

3.1.3 Deliverables

As we anticipated, the project deliverable at this stage is a visual representation of the data architecture of the organization. We report an example of this in Fig. 3.2. Please note that this is not meant to represent the real underlying architecture used in the project, but rather an illustrative example.

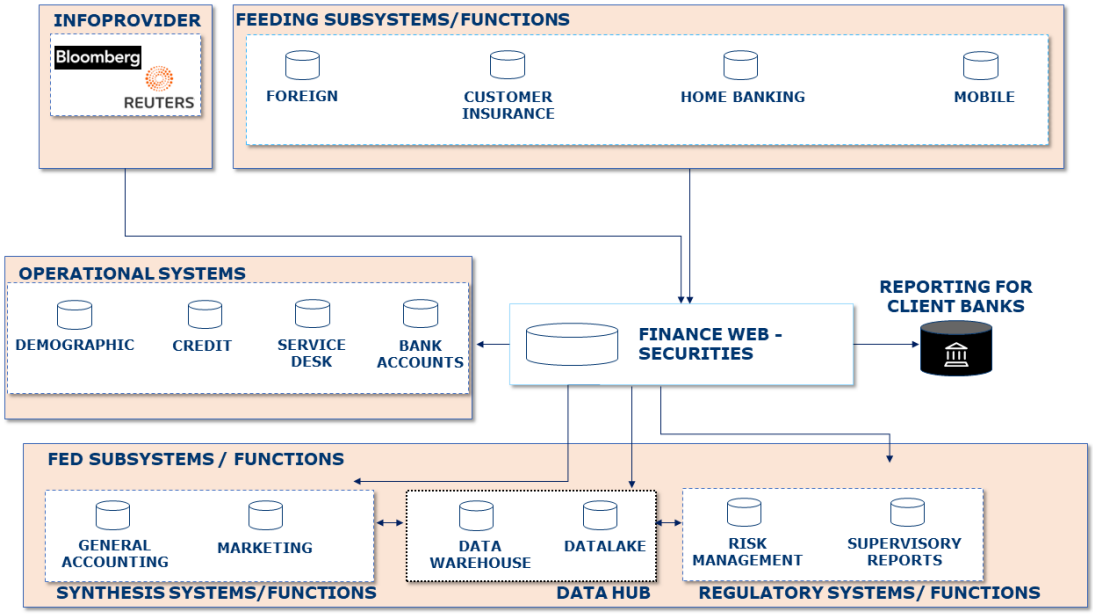

Figure 3.2. Example of data architecture visualization for the Financial Instruments branch of a bank.

The proposed example reports the architecture of data assets pertaining to a specific division within a financial institution, specifically one that specializes

in the management of financial instruments (e.g. stocks, bonds, derivatives, insurance, etc.).

Before going through the description of the elements of the architecture, we remark how the frames *Feeding subsystems/ functions* and *Fed subsystems/ functions* specify two types of concepts: *systems* and *functions*. To understand the difference between these two, but also the need for differentiation, it is worth introducing here a concept that will be more explicitly defined in the metadata management section: the dual aspect of information assets in an organization. As we have seen in the introduction, the information assets of an organization are not limited to data and systems processing them but include intellectual, human, and organizational assets. Thus, when approaching data discovery, we have to make sure to include in the architecture not only the systems processing data but also the *business processes* involving them. This explains the role of the term *functions* in the architecture: it allows us to include in it data elements that, if we just looked at systems, would not have been reported.

Moving to the main components, they include:

1. **Finance Web - Securities**: the central database. It stores the core information related to financial instruments.
2. **Infoprovider**: a system responsible for gathering, managing, and distributing financial market data and information from external sources. The Infoprovider integrates data from *Bloomberg* and *Reuters*, two well-known providers of financial market data, and stores this data before making it available to the central database.
3. **Feeding Subsystems/ Functions**: these are the subsystems or functions that provide input to the central database. In our example, they include:
    i. Foreign: data related to foreign financial instruments or assets (e.g.

currency exchange rates, foreign financial markets indices, etc.)

ii. Customer Insurance: insurance-related data for the bank's customers (e.g. data related to insurance policies, coverage details, customer insurance profiles, etc.)

iii. Home Banking: data related to online banking transactions conducted by customers (e.g. account balances, bill payments, etc.)

iv. Mobile: data generated through mobile banking applications and services (e.g. transaction data, mobile app usage statistics, etc.)

Note here the concept of system and function: each of these 4 data assets can be seen both as the technical databases storing the information and as the business processes producing them.

4. **Operational Systems**: systems managing the day-to-day operational processes and transactions of the financial institution. These systems, which are demographic, credit, service desk, and bank accounts, receive data from the central database and store them in the corresponding one.

5. **Fed Subsystems/ Functions**: subsystems or functions that receive data or information from the central database. They are subdivided into 3 categories:

I. Synthesis Systems/ Functions: these systems or functions are used for report creation. They join multiple data received from the central database and calculate KPIs that are eventually collected into reports. In the example, two elements are reported: *General Accounting* and *Marketing*

II. Data Hub: this system represents a comprehensive data management and analytics solution that supports various data-related needs within an organization. It could include, for example, a Data Warehouse, used to store and manage structured data, and a Data Lake, a scalable and flexible storage system that can hold vast amounts of structured, semi-structured, and unstructured data.

III. Regulatory Systems/ Functions: these elements are also used for report creation, but specifically for regulatory reasons. As an example of these kinds of entities, we included *Risk Management* and *Supervisory Reports*.

Note how these 3 entities, in the figure, are interconnected: this is to represent the exchange of information happening between the data hub and the two other entities of the fed subsystems/functions.

6. **Reporting for Client Banks**: systems handling additional reporting requirements specifically for client banks

3.2 Metadata Management

3.2.1 Definition & Objective

The second tool for the implementation of the data governance framework is metadata management.

As from DAMA, metadata is *"information about technical and business processes, data rules and constraints, and logical and physical data structures"*. In other words, metadata is data describing characteristics of other data. For example, the names of the columns of a table of a database, as well as information about the format of the data each column contains (e.g. integer number, 10-character long string, etc.).

Metadata can be categorized into the following three types.

- **Business metadata**: they describe the content of the data using non-technical terms. Examples include entities and attributes, valid domain

values, definitions of the content of database tables, data models, data lineage (i.e. the path of the data, this will be examined in detail later), and algorithms used to create some data.

- **Technical metadata**: they describe the technical details of data, like systems storing the data and processes moving them between systems. Examples include table names, column names, column properties ETL (Extract, Transform Load) job details, and data access.

- **Operational metadata**: they describe properties of the processing of the data. Examples include: log data about job executions, backups, date created, history of database extractions.

It is thus very clear why metadata is critical to Data Governance: it constitutes the basis for building knowledge about data. More in particular, metadata is fundamental to organizations as it allows the understanding of their data, thus making possible a number of activities in Data Governance, among which are the following.

- Bringing knowledge about data: what data does the organization have? What do data represent? Where do they originate? How do they move through systems? Who has access to them?

- Assessment of data quality (which is later discussed as a separate knowledge area): what does it mean for data to be of high quality?

- Risk management: what personal data is processed in the organizations? What data is subject to regulations? What data is used to evaluate risk exposure?

Without metadata, the answer to all these kinds of questions would not be possible.

3.2.2 Technique

Moving on to more practical terms, how is metadata managed in an organization? There are two steps to follow in order to establish knowledge about metadata: first collecting them and then storing them.

1. **Collecting metadata**: as we have seen, metadata comes from many different processes. To effectively collect all the different kinds of metadata, a prior step of discovering the data underlying the metadata we are interested in must be carried out. In other words, all the organization data systems (e.g., databases, applications) must be registered, along with the contained entities (e.g., tables) and with the attributes contained in each entity (e.g., columns). This is achieved through a process called **Data Lineage**, which will be discussed in the next paragraph. Once an exhaustive list of all the data is obtained, the metadata can start to be collected. In this phase, a reverse-engineering approach can be adopted, i.e. collecting metadata from existing documentation (e.g., definitions of columns of a table), but this can present some risk, the biggest one being the possible lack of care at the moment of the documentation was created, resulting in improper metadata definition. Thus, a better approach [5] is to develop metadata by analyzing the underlying data and establishing brand-new definitions. This, of course, requires time and writing skills but results in a more robust metadata definition.

2. **Storing metadata**: once the metadata has been collected, it must be stored in specific structures identified according to the type of information they provide. We report here the two main metadata repositories used: the **Data Dictionary** and the **Business Glossary**.

Data Lineage

Data lineage is a tool that allows the tracking and tracing of the origin, movement, and transformation of data throughout its lifecycle in the organization [19]. In other words, data lineage is a map showing how data evolves from its sources to its destination [20].

As a first example, we report a visual data lineage in Fig. 3.3. We denote with the term *node* the different stages where data passes from one source to another. Nodes can represent both jumps between different databases and jumps between business processes, e.g. from one company department to another. In our example, we trace back the path of the *Amount* attribute, indicating with the pink arrow the dependencies across the different stages, and thus the Data Lineage.

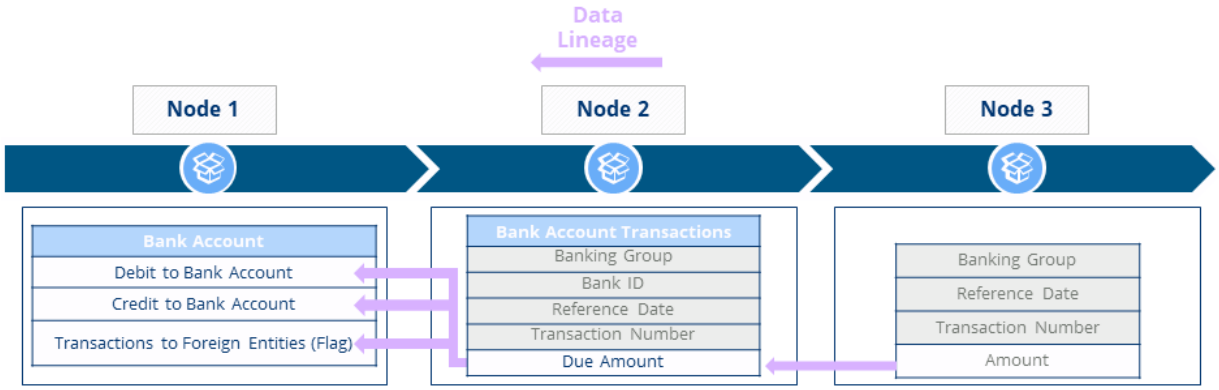

Figure 3.3. Example of Data Lineage.

Data lineage is considered the primary tool in data governance, as it makes possible the following:

- Exhaustive documentation of the data landscape of the organization (with significantly more detail than the data discovery phase, which only

serves as a starting point for the implementation of data governance)

- Impact analysis, i.e. analysis of the root cause of a data problem (e.g., some KPI starts showing too many null values)

- Identification of grafting points for data quality controls (we discuss this in detail in the Data Quality Management section)

- Identification of the so-called *master data*, that is the data representing the source of truth, when multiple sources of the same data are available across the organization

Data lineage can be performed according to different points of view, granularity levels, direction, and alignment. Each combination of these four criteria leads to a different data lineage.

1. **Points of view**: this is related to the data concepts subject to the lineage. Are we tracing the path of data from a **business** point of view (i.e., data seen as business processes) or from a **technical** one (i.e., data seen as tables and columns)?

   a) **Business/logical** point of view captures the path of data in terms of business processes producing it. We report an example of business data lineage for a financial institution in Fig. 3.4: the *Family Balance Reporting*, which is the process of generating reports summarizing the combined financial status of a customer's family for assessment (e.g., amount spent in a month, type of expense, etc.), contains data directly coming from two processes: *Customer Information Integration* and *Transaction Recording*. These two processes represent respectively the processes of integrating customer data from various sources into demographic databases, and that of storing detailed records of credit card transactions for record-keeping and reconciliation. These

two processes depend, in turn, on other two processes, namely *Customer Data Entry*, which is the process related to capturing and entering customer information during onboarding, and *Credit Card Usage*, which represents the processing of credit card transactions at the point of purchase.

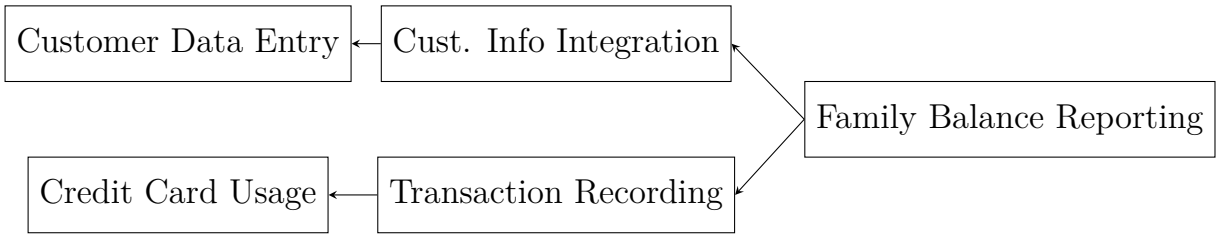

Figure 3.4. Data lineage example from a business point of view.

b) **Technical/physical** point of view captures the specific names of the objects implemented in the databases/applications. We report the same example as before but from a technical point of view., in Fig. 3.5: *Reporting and Analytics* is the system generating reports that summarize the financial position of a customer's family; *Customer Demographics* is the database containing information about customers; *Transaction Database* is the database storing transaction data (e.g., amounts, dates, etc.); *CRM System* is the system facilitating data collection and entry during customer onboarding; *POS System* is the system processing credit card transactions.

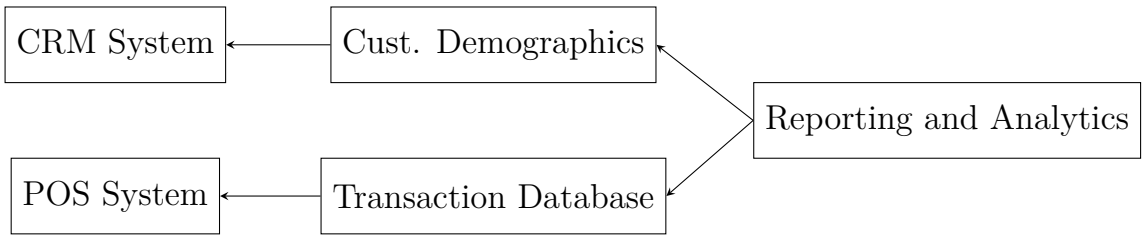

Figure 3.5. Data lineage example from a technical point of view.

Thus, the same data lineage is now expressed in terms of systems/applications producing the data.

2. **Granularity levels**: this refers to the "zoom" level of the data lineage. Are we tracing the path of data in terms of systems/macro processes, in terms of the single tables/processes contained in them, or in terms of the single columns/detailed process contained in each one? We adopt the following terminology to indicate the granularity level:

   a) **Scope**: the bigger set of data sources inherent to a given business area or physical system.

   b) **Entity**: the single processes producing data in a given business process, or the single data structures composing a system (e.g., tables).

   c) **Attribute**: the most granular concept of a business process, or the actual data contained in the data structures (e.g., columns).

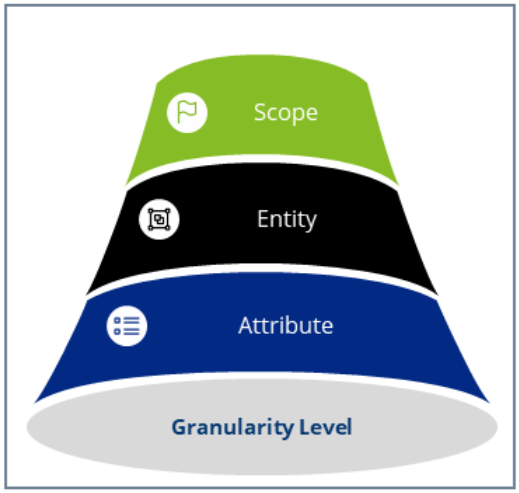

Figure 3.6. Representation of the three granularity levels for data lineage.

More details on the granularity level structure are reported in the Business Glossary section, as that is the tool where it was originally introduced.

3. **Direction**: this refers to the direction followed when tracing the data.

   a) **Forward**: expresses how data moves from source to destination

   b) **Backward**: reveals the origin of data

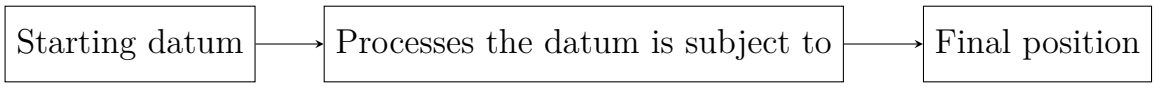

Figure 3.7. Example of Forward Data Lineage.

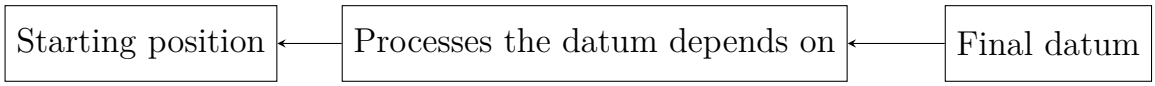

Figure 3.8. Example of Backward Data Lineage.

4. **Alignment**: this refers to the fact of whether the lineage is made across systems/processes inherent to a specific point of view or whether the lineage is made across different points of view.

   a) **Horizontal**: the lineage considers data entirely from a business point of view or rather entirely from a technical point of view.

   b) **Vertical**: the lineage functions as a join between the business and the technical points of view. This can be made after both the business and technical data lineages have been made and is crucial to connect the two worlds to give an extensive view of the data landscape of the organization.

   The concept of vertical data lineage is reported in Fig. 3.9. To better understand it, we also report the vertical data lineage that would result from the two horizontal data lineages seen in Figures 3.4 and 3.5, of which the first is from a business point of view whereas the second is from a technical one.

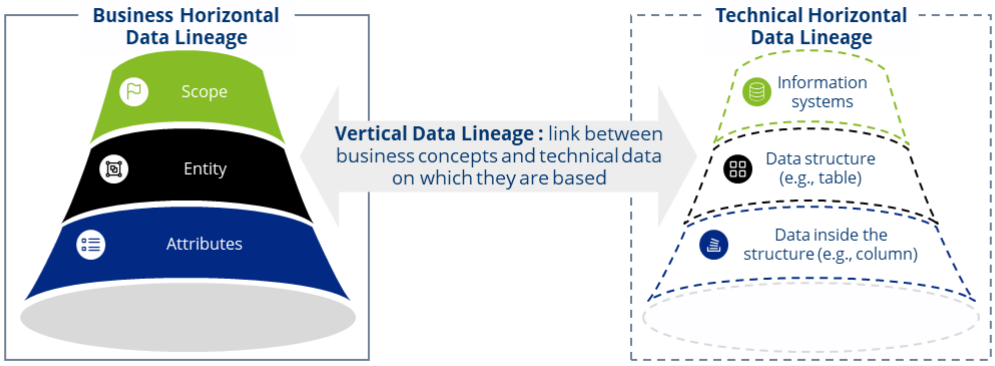

Figure 3.9. Representation of Horizontal and Vertical Data Lineage.

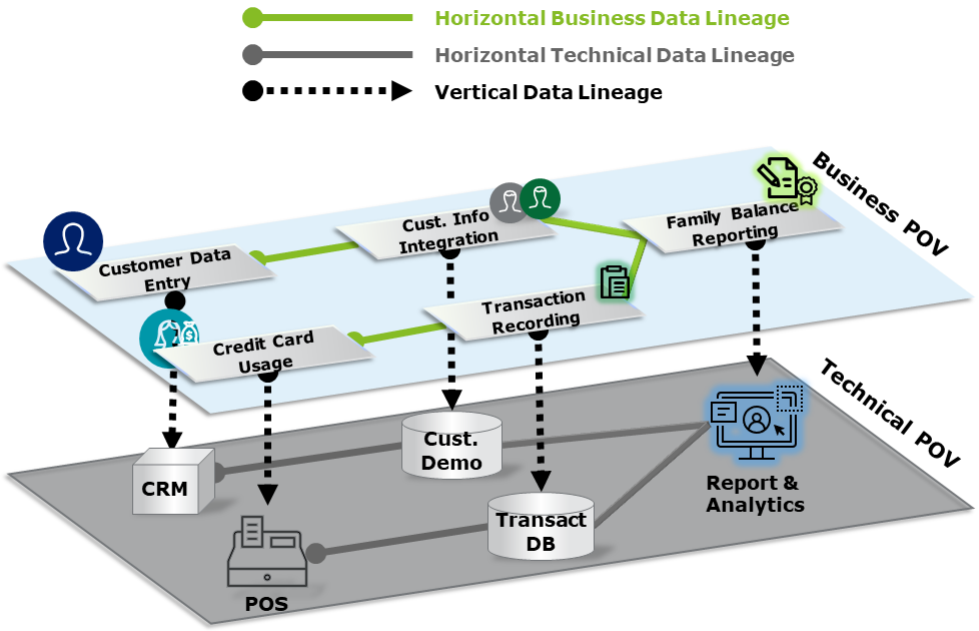

Figure 3.10. Visual example of Vertical Data Lineage.

To complete this section, we would like to propose an analogy to better understand data lineage: a geographical map. A geographical map is a tool to represent the world, and it can do so by expressing multiple characteristics, like elevation, streets, and others. Each of these characteristics brings to a different map, while the underlying world remains the same. Thus, we can imagine all the produced maps as layers that can be put on one another, and that collectively bring extensive information on the world. This concept is illustrated in Fig. 3.11. Data lineage, on the other hand, mimics the same process: at the base, there is a data warehouse, or, more generally, the data landscape of a company. On top of that, multiple versions of lineages, that are the equivalent of the map layers, can be created, by combining the 4 different criteria seen for the data lineage.

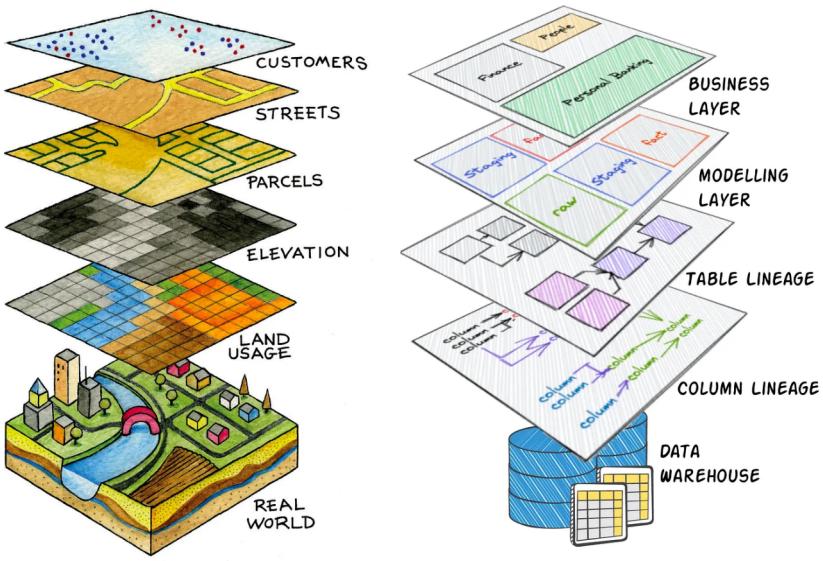

Figure 3.11. Analogy between the layers of a geographical map and those of a Data Warehouse (Source: [20]).

Even if this analogy can help us comprehend data lineage, we specify that, in practical terms, data lineage is represented by a document, often a spreadsheet, indicating all the stages the data passes through. The spreadsheet contains multiple columns, each indicating a *node*, which is a stage in the

data path, and the values contained in that column represent the name the data gets in that specific stage. Nodes can be identified by observing, in the path of data, the moments at which data:

- moves from a consistent and persistent database to another
- transitions from one procedure/system to another
- shifts from one organizational structure to another

Then, by analyzing a single row, one can trace the path of that data across the different nodes.

At first glance, data lineage might appear less useful, as one might anticipate that the name of the data remains consistent across various stages. Indeed, this isn't accurate, as data can undergo name changes as they progress through each stage of their journey. For a real example of this, see the Deliverables section.

Business Glossary

Now we have understood how metadata can be collected, we move on to metadata storage. We start with the first structure, the Business Glossary.

A business glossary is a spreadsheet containing the organization's business concepts and terminology, definitions, and the relationships between those terms [5]. It represents metadata, as the name suggests, from the business point of view. This means that, in the business glossary, there are no technical terms related to databases or systems.

More in particular, the business glossary describes, at a functional semantic

level, the organization's information assets, and is constructed through a multi-level logical structure, comprising the following:

- Information Map: it contains the definitions of the scopes, entities, and attributes relevant to business management, along with their relationships.
- Data management responsibilities: the structure of the responsibilities of data management (i.e., who is responsible for which part of the data landscape).
- Classification: details on the information assets object of the glossary, such as sensitivity and usage.
- Control rules: rules on how the definitions of the information map are given.

A visual representation of the business glossary is reported in Fig. 3.12.

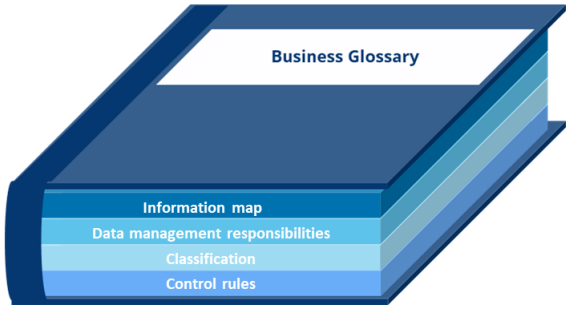

Figure 3.12. Representation of the Business Glossary.

The business glossary uses the same hierarchy adopted when implementing the granularity level in the data lineage. As anticipated, the granularity level structure was originally introduced in the business glossary implementation. In Fig. 3.13, we can see the E-R model for the business glossary.

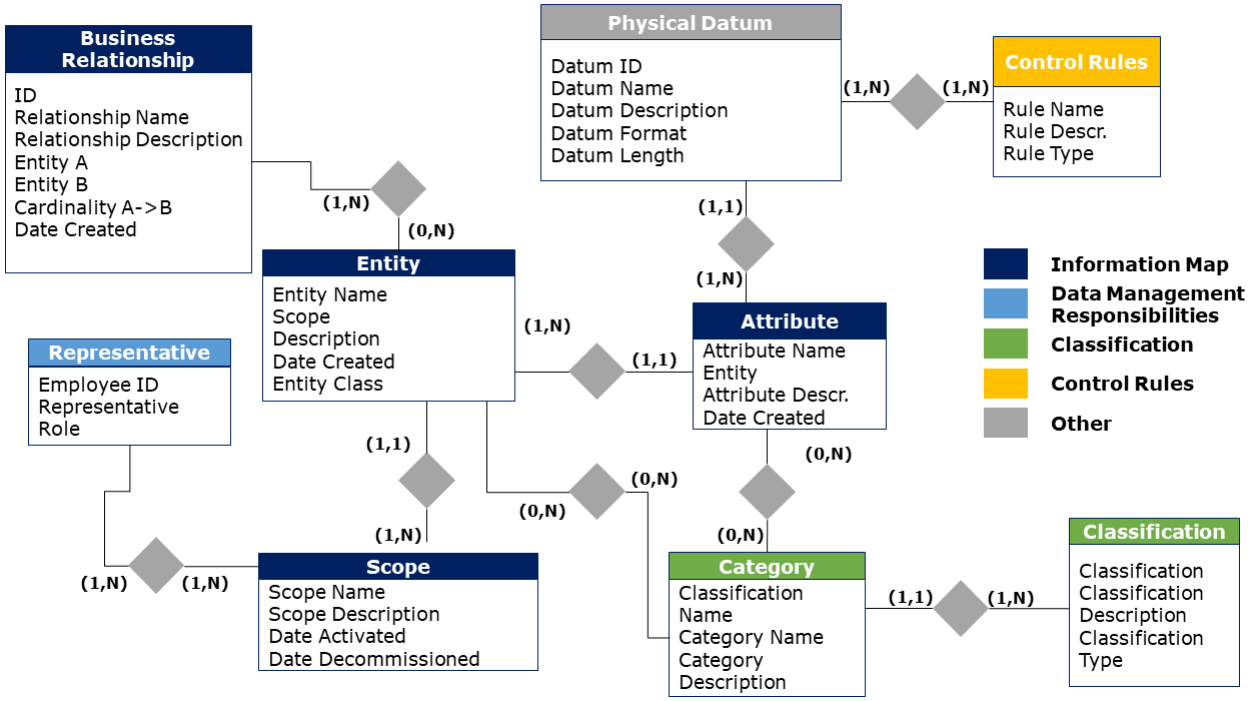

Figure 3.13. E-R Model for the Business Glossary. Primary and Foreign Keys are not reported for simplicity.

An important aspect of the business glossary regards its implementation: seen its business nature, a *top-down* approach must be used for its creation, meaning that one should start defining terms starting from the high-level business processes, and not from the information systems. This logic is maintained also in the hierarchy scope, entity, and attribute presented in the data lineage section: scopes are firstly defined starting from the high-level business processes of the organization; then, a set of entities, attributes, and relationships are defined for each scope, considering the smaller sub-processes contained in it.

Data Dictionary

The other main structure used to store metadata is the Data Dictionary. A data dictionary defines the structure and contents of data sets and can be used to manage the names, descriptions, structure, characteristics, storage requirements, default values, and other attributes of each data element in a model [5]. It can be thus seen as the technical-side equivalent of the Business Glossary.

The kind of information contained in the data dictionary is defined by data architects during the original Data Modeling phase, precisely at the logical level, but often is lost when transitioning to the physical level. A data dictionary can prevent the complete loss of this information within the organization, and it can also contribute to maintaining alignment between the logical and physical models subsequent to the deployment in a production environment.

Finally, we remark that the data dictionary, contrary to the business glossary, is implemented with a *bottom-up* approach, meaning that definitions for data are collected starting from the attributes, then moving up to entities, and then scopes.

3.2.3 Deliverables

Three are the project deliverables of metadata management:

1. **Data Lineage document**
2. **Business Glossary**
3. **Data Dictionary**

Starting with the data lineage document, we report the following example.

Consider a scenario where a Machine Learning model is employed by a financial institution to determine Risk Class for customers. If the model generates a singular numerical output (or a string denoting the Risk Class) based on several input data, the resulting output data assumes an entirely new identity – representing a distinct concept – even though it remains dependent upon the original input data. Additionally, data may undergo transformations between two nodes; for instance, normalizations or percentage calculations. All these concerns are indeed targeted by data lineage. The resulting data lineage spreadsheet for this example is reported in Fig. 3.1, whereas the corresponding visualization of the underlying ML model is in Fig. 3.14.

We complete this example by reporting a question the company stakeholders could ask about the Customer Risk Class model and that data lineage could help answer: *The Customer Risk output is not consistent with our expectations. Which data does it depend on?*. Of course, if the company only has access to a visual scheme like that of Fig. 3.14 drafted by the Data Scientists who developed the model, this question might not be properly and exhaustively answered. Thanks to the spreadsheet document in Table 3.1, indeed, it is very easy to have the complete list of input data to the Customer Risk Class model (listed in *Node 2* column of the table), as well as the original data the inputs depend on (listed on *Node 1* column), and thus answer the stakeholders' question.

Moving to the business glossary, we report an example of the Information Map part, which is usually its most substantial part, in Table 3.3: we identify three terms that are found to be significant for the company and narratively define them, We also indicate, in the middle column, the granularity type, with reference to what we said regarding the data lineage. The business glossary itself does not necessitate the presence of this extra information, but, in real-life implementations, including it is a good practice to reach the

Node 1	Node 2	Node 3
Customer Debt	Customer Debt as % of Customer Balance	Customer Risk Class
Customer Balance	Customer Debt as % of Customer Balance	Customer Risk Class
Customer Education	Normalized Ed. Level	Customer Risk Class
Customer Salary	Customer Salary	Customer Risk Class

Table 3.1. Example of a Data Lineage spreadsheet.

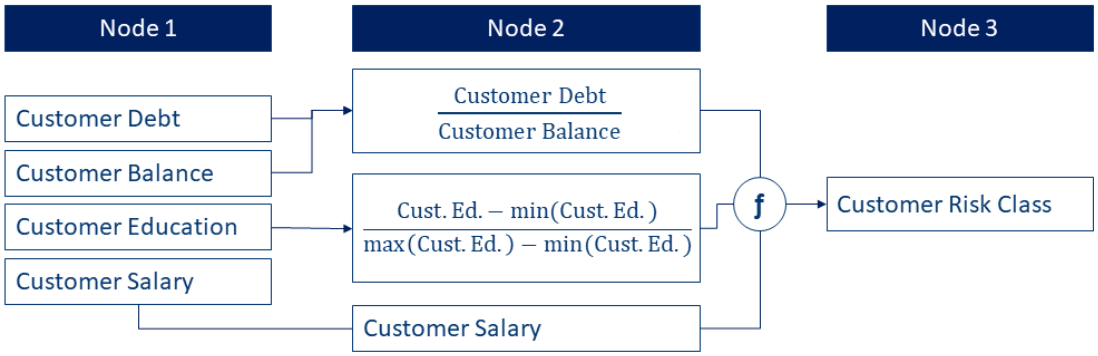

Figure 3.14. Representation of the ML model underlying the example of Table 3.1.

full potential of this metadata storage tool.

Finally, we report a very simplified example of a data dictionary in Table 3.2: each row corresponds to a specific field in a database, and information like Field Description, Size, and Data Format are reported, as well as the Domain set, when applicable.

DB	Field Name	Field Label	Field Description	Field Size	Data Format	Domain
HRM	EmpID	Employee Identification Number	Identification number assigned to employee at time of hire	8	String	
HRM	SepReas	Separation Reason	Reason an employee has separated from the agency	2	Numeric	1- Abandon 2- Retirement 3- Seasonal
XYZ	IDcode	Employee code	Unique code of the employee	8	String	

Table 3.2. Example of a Data Dictionary.

Term	Granularity Type	Definition
HR	Scope	Company's HR dept
Employee Registration	Entity	Employee registration activities
Employee ID	Attribute	An identification number assigned to an employee at the time of hire

Table 3.3. Example of a Business Glossary.

3.3 Data Quality Management

3.3.1 Definition & Objective

Our third tool to implement a data governance framework is Data Quality Management.

In the context of organizations, the basis for claims regarding the value of data is the assumption that data is reliable and trustworthy. This essentially means that the data possesses a high level of quality.

The concept of *data quality* is the same as that of *quality* when applied to various other goods or products: it involves ensuring that the product is reliable and fit for its intended use. Thus, data quality refers to the degree to which data accurately and appropriately represents the real-world entities or phenomena it is meant to depict.

More in particular, data quality is the collection of models, methodologies, and tools that ensure the information contained within the corporate information system maintains a level of reliability suitable for ensuring customer relationship management and supporting governance activities and external reporting (mandatory regulatory compliance).

A program aimed at ensuring data quality in an organization is called Data Quality Management, and it oversees data throughout its lifecycle by establishing standards, integrating quality into the processes responsible for data creation, transformation, and storage, and assessing data against these standards.

Data Quality Management is a function in organizations, in the same way as Data Governance is. Indeed, these two functions are often implemented

in different programs. Nevertheless, they remain strictly linked, and Data Quality Management can be considered part of the overall Data Governance Framework, as its final objective is to ensure the integrity, reliability, and usability of data assets throughout their lifecycle, aligning with the overarching goals and strategies of the organization, and thus falling under the scope of Data Governance.

Needs for Data Quality Management include:

- **Regulatory Compliance**: Implementation of Data Quality systems to meet supervisory requirements of various regulations (Finrep/Corep, Basel III, MiFID, Circular 263 B.I.).
- **Business Enhancement**: Effective, efficient, and structured data quality control to ensure the accurate reliability and timeliness with which information is made available to Top Management. Such support enhances business decisions and safeguards against losses arising from incorrect decisions.
- **Cost Efficiency**: Reduction of operational costs through optimization of control processes, minimizing rework activities, and decreasing the number of Full-Time Equivalents (FTEs) involved in certain Back-Office processes.

3.3.2 Technique

Data Quality Management involves three dimensions of implementation aimed at establishing a Data Quality Management System:

1. **Organizational Model**: the model that aims to empower business structures regarding the data quality management process. It involves

setting up roles (e.g., who is responsible for the supplied data, who manages the quality, and others), and also establishing an organizational structure for coordinating and overseeing the data quality management processes

2. **Functional Model**: the analytical model that identifies specific control types that extensively represent all aspects of data quality, along with quantitative alert thresholds for each. It also addresses the procedures for implementing controls.

3. **Architectural Model**: the comprehensive model for the implementation of data quality control applications. This model includes the following steps:

   i. Defining the control framework and the grafting points

   ii. Defining Control Libraries for all major areas of interest

   iii. Creating a dashboard for monitoring and analyzing control results, as well as visualizing reports

We only focus on the functional and architectural models. The reason for this is that the organizational model analyses the data quality management implementation from a management perspective, addressing methods to structure people and processes, while our aim is to provide details from the technical point of view.

Functional Model

As anticipated, the functional model formalizes the concept of data quality control by means of an analytical model.

Before defining the fundamental elements of the model, we provide some context by presenting the three types of data quality controls that can be

identified.

1. **First-level controls**: controls performed during the data input phases by the employees in charge
2. **Second-level controls**: controls performed on data flows exchanged between systems (both operational and synthesis systems) by the data quality management system
3. **Third-level controls**: controls performed by the Internal Audit unit on the whole data quality system, aimed at verifying its related benefits

We report visually in Fig. 3.15 the three types of data quality controls.

The controls of the data quality management system refer exclusively to data quality assessment, i.e. they are second-level controls. They are also called *merit-based* controls and, as anticipated, they are performed on data flows exchanged between operational and/or synthesis systems.

The two fundamental elements of the functional model of a data quality system are the *Control Logic* and the *Data Quality Control*:

- The **Control Logic** is the set of operations involving data extraction, combination, and mathematical calculations that allow the computation of the indicator.
- The **Data Quality Control** is the operation of comparing the quality indicator with one or more predefined threshold values. The control identifies one and only one type of anomaly, i.e., it is related to a single aspect of data. It comprises three parts:
  I. An **indicator**, i.e. the metric representing specific elementary information regarding a single piece of data or a set of data. Indicators can be:

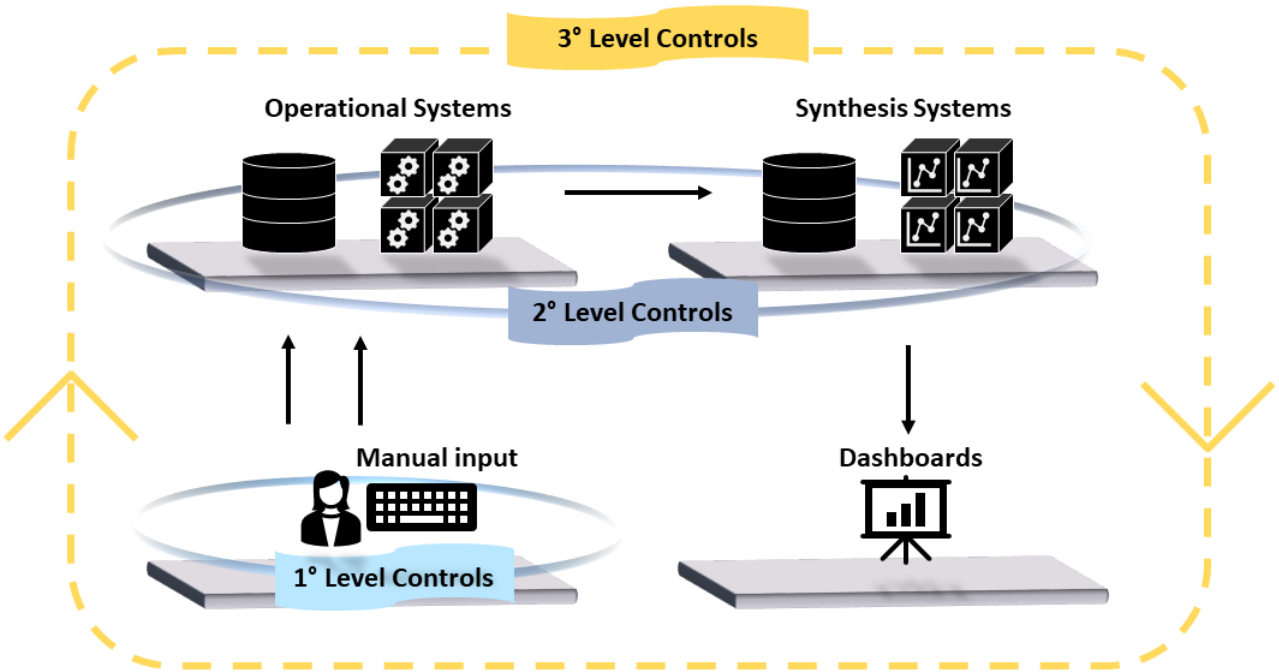

Figure 3.15. Illustration of the three types of data controls visualized in a simplified data lifecycle.

i. *Indices*: the ratio between favorable cases and total observations (ranging from 0 to 1)

ii. *Temporal metrics*: compliance with time-related conditions (YES / NO)

iii. *Percentage deviations*: the percentage difference from reference values, trends, or historical averages

II. The **thresholds**, defined by the Data User (i.e., the people that formally use a specific dataset), represent the minimum value that the indicator must reach in order to correspond to a specific level of quality. The DQM model defines two thresholds: *Non-Admissibility*

Threshold (NAT) and *Conformity Threshold (CT)* (see later for explanation)

III. The **target value** (optional) is used to set maturity goals to be monitored and improved in line with the measured quality level. It can be changed from time to time, in order to increase the target quality of the control. Unlike the Thresholds, though, it does not factor into the quality evaluation.

The outcome of the control expresses whether the quality level of the dataset is good enough for it to be released or not. More specifically, the outcome can be:

- *OK*: the value of the indicator is above the CT. For the considered control, the monitored dataset does not exhibit anomalies (or these are present to an insignificant extent for data quality purposes). The dataset can be released.
- *Warning*: the value of the indicator is below the CT but above the NAT. For the type of control considered, there are anomalies that limit the quality of the final result within the scope of the control. If the timing requirements for data flow release permit, rework activities will be carried out. Regardless of their outcome, however, the data flow can still be released
- *KO*: the value of the indicator is below the NAT. For the considered control, there are anomalies that inhibit the release of data related to the company's scope, compared to the defined quality level. If the timing requirements for data set release permit, rework activities will be carried out. In cases where this is not possible or the outcomes of the checks do not improve with reworking, it is necessary to initiate the *escalation process* for (1) potential exceptional release of the dataset, or (2) commencement of additional rework activities (even if the timelines do not allow compliance with pre-established deadlines).

An example of a functional model for data quality management is reported in Fig. 3.16.

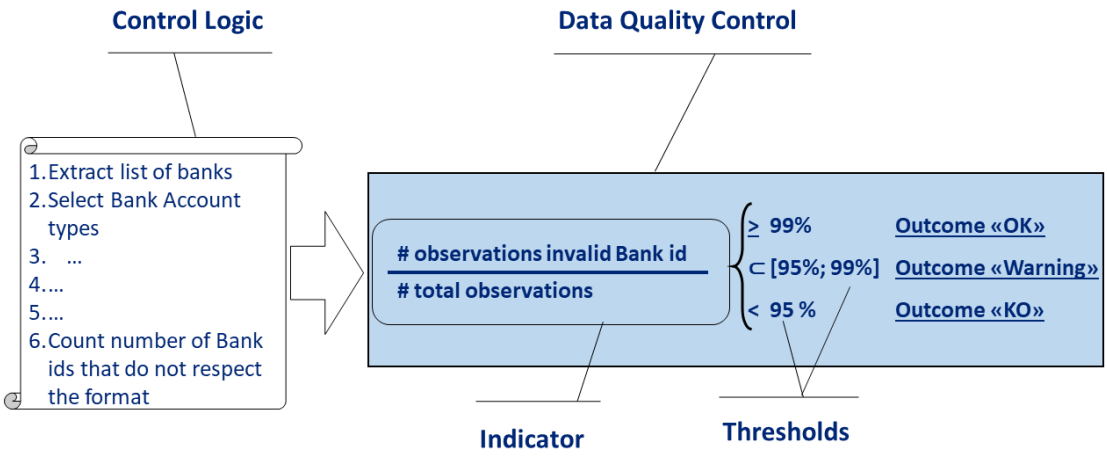

Figure 3.16. Example of fundamental elements of the Functional Model for the Data Quality Management System.

It is thus clear how the core component of the functional model is represented by the indicator, as it defines the aspects of data to be analyzed. Thus, the definition of indicators is crucial for the implementation of data quality systems.

The common approach is to create lists of indicators through an unstructured procedure, with the consequence that, often, incomplete and inadequate indicators are obtained. It is therefore essential to arrive at a structured definition of the concept of data quality indicators in order to avoid potentially partial and subjective evaluation models.

To achieve this, it is necessary for the indicators to be defined based on specific *quality dimensions* that allow for adequate formalization (indices,

temporal metrics, percentage deviations, variations compared to historical averages) and weighting (highlighting what is truly important and what is not) of the indicators.

In our approach, we classify data controls by introducing the following tree diagram to identify the quality dimensions:

1. **Data-Centric Controls**: controls that focus solely on the piece of data itself
    I. **IT-related controls**
        i. **Uniqueness**: verify that entities are identified with the correct key and are therefore not duplicated; the analysis should be focused particularly on domain tables (e.g., verify that there are no multiple records with the same credit line number for the indicated customer ID)
        ii. **Integrity**: verify that data belong to their class of allowable values. (e.g., the country must take values in ['IT', 'BG', ...])
    II. **Single-point controls**
        i. **Completeness**: verify the presence of all necessary data for analysis and ensure that they have an adequate historical depth
        ii. **Accuracy**: controls on the number of records extracted and sent to reports. These controls alert users to the existence of issues even before contextualizing the data
2. **Contextual Controls**: controls that consider external factors and the context of the piece of data
    I. **Timeliness controls**: Controls aimed at verifying the ability to execute the process on up-to-date data capable of representing events and phenomena in their immediacy of occurrence (e.g., absence of update delays)

II. **Logical/ Coherence**: verification of compliance with interdependency constraints among correlated variables in the database (e.g., the installment payment date must be greater than the date of mortgage agreement date)

III. **Trend**: relation of the data from a stream to the same data in the same stream produced in the previous month or the same month of the previous year, in order to neutralize any seasonal effects. For this type of check, the identification of the tolerance threshold is crucial

Summarizing, we identify 7 classes of controls: Uniqueness, Integrity, Completeness, Accuracy, Timeliness, Coherence, and Trend. These quality dimensions are to be used by organizations to identify a list of control indicators that ensure comprehensive coverage of all aspects of process quality.

Architectural Model

The architectural model model presents the concrete functions needed in order to set up the functional model.

The architectural model provides 5 steps:

1. Set up a procedure for data controls implementation
2. Set up a data model for storing the results
3. Set up a procedure for running the controls
4. Set up problem handling procedure for KO controls
5. Set up a dashboard for results inspection

We focus on the fundamental activities, thus briefly going through steps 1 and 2, which constitute the building blocks for the subsequent 3 steps.

Step 1: data controls implementation. This step is composed of the following two parts.

1. Selecting data control types: in this sub-step, the data architecture of the specific financial institution must be analyzed at the most granular level (e.g., database columns), in order to identify the most meaningful quality dimension for the controls to be implemented.

2. Identifying grafting points: this sub-step also involves the analysis of the underlying data architecture, but now with the aim of optimizing the positioning of the data quality controls. For example, it is advisable to implement diagnostics in all *self-consistent* processing phases (i.e., phases that require more processing): indeed, there, it is recommended to anticipate all possible controls by positioning them at the early stages of the data path, in order to avoid costly rework.

Step 2: the data model for storing the results For what concerns the storage of the data quality control results, the E-R model in Fig. 3.17 is adopted.

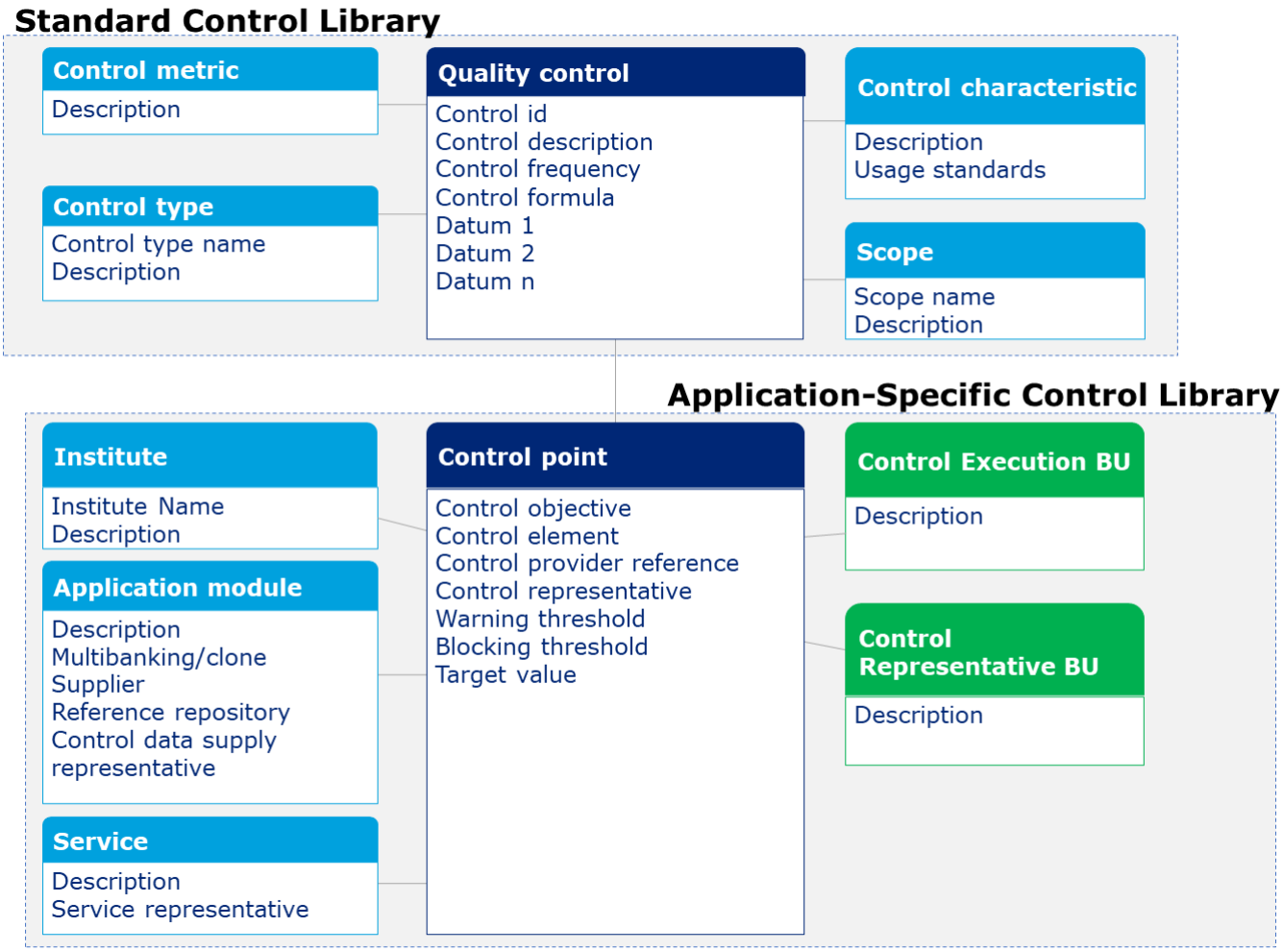

Figure 3.17. E-R Model for the Data Controls, subdivided into two parts: Standard Control Library part (top frame) and Application-Specific Control Library part (bottom frame). Primary/ foreign keys and relationship cardinality are not reported for simplicity.

3.3.3 Deliverables

Deliverables of the data quality management activity of the data governance framework implementation are represented both by the obtained dataset,

which contains the outcomes of the data quality controls and has the structure illustrated in Fig. 3.17 and a presentation document summarizing the dataset in a more easy to understand format.

We report in Fig. 3.18 an example structure of the presentation document. The presentation focuses on 4 aspects:

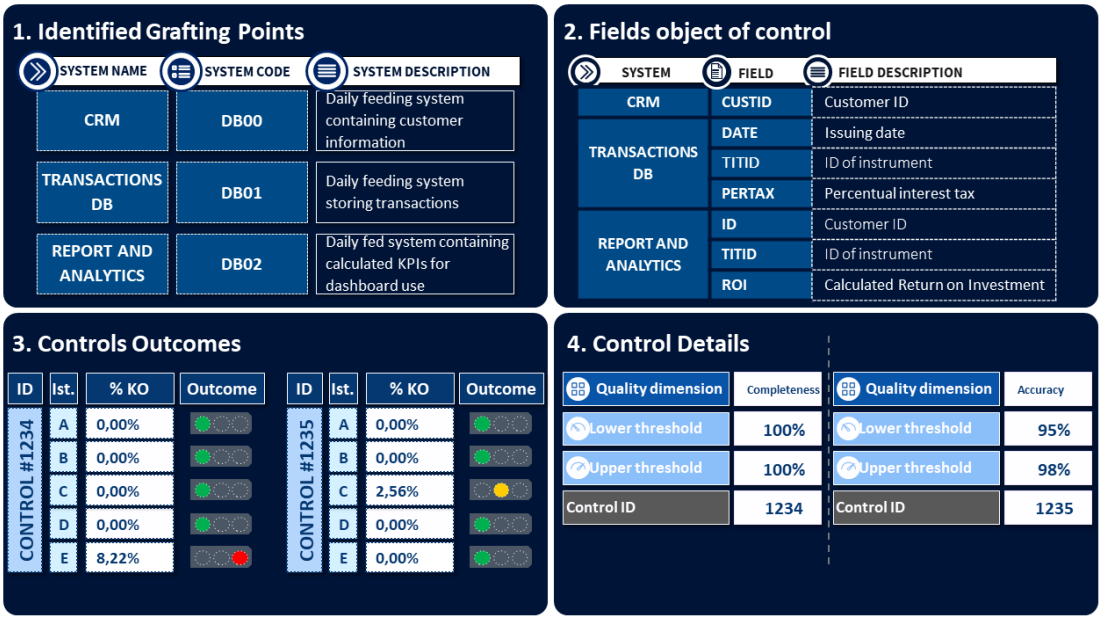

Figure 3.18. Example structure of the presentation document for the outcomes of the data quality controls.

1. **Identified Grafting Points**: here the systems where the controls are positioned are summarized.
2. **Fields Object of Control**: a more granular description of the grafting points for each system.
3. **Controls Outcomes**: the actual, numerical outcome of the controls.
4. **Control Details**: specific information on the control (e.g., NAT and CT thresholds, Data Quality Dimension)

Chapter 4

Results and Analysis

Following the *Method* part, a chapter regarding the results is pretty natural. However, we remark how this thesis reflects a real project, and thus, due to confidentiality reasons, no data about the results can be reported. Consequently, a real *discussion of results* is not possible for this project.

Nevertheless, we aim to conclude this work with an explanation of how the data governance framework implementation can be assessed. Thus, even without providing any data, we report in this chapter the methods used for this scope.

4.1 How to measure implications of Data Governance?

Implications of a data governance framework implementation are related to the organization's knowledge about its information assets: the more extensive

the framework is, the more knowledge it brings.

Thus, in order to quantify the benefits brought by the data governance framework to the organization, we must determine specific metrics expressing the level of knowledge an organization has about its information assets.

To monitor the results of data governance in a comprehensive way, we use a specific framework developed in 2022 by the Information Governance Observatory of ABILab [1], which identifies a set of Key Performance Indicators (KPIs) useful for providing reliable feedback on the achieved results and suggesting improvement actions.

The ABILab framework structures the metrics as follows:

I. *Context Metrics*: metrics related to the establishment of data governance at the corporate level. The goal is to assess the state of the data governance framework in terms of strategic objective coverage and roadmap development. These metrics are categorized into 3 dimensions:

   i. *Activation*: they assess the methods used to establish the corporate Data Governance framework (policies, implementation plans, data governance processes, organizational model, and ownership) for consistency with the Data Strategy. Example: *Coverage of Data Governance areas compared to the planned ones.*

   ii. *Architecture*: they evaluate the corporate architectural model, technological infrastructure for data management, data security and protection capabilities, and enabling tools. Example: *Percentage of coverage of the technological capabilities required to support Data Governance.*

[1] Abi Lab is an establishment known as the Banking Research and Innovation Center, founded by the Italian Banking Association (ABI). Its primary objective is to foster connections between banks and information and communication technology (ICT) firms.

iii. *Culture*: they assess the availability of data-related skills, initiatives to promote a Data Culture, and awareness campaigns for personnel. Example: *Percentage of training hours dedicated to data-related topics out of the total training hour.*

II. *Perspective Metrics*: Metrics pertaining to the data governance operations. The objective is to assess the execution capability of data governance based on the implemented components and the results achieved across various domains. These metrics are categorized into 5 dimensions:

i. *Metadata and Data Model*: Verify the level of coverage and effectiveness of Metadata Management activities, as well as the benefits associated with these activities. Example: *Degree of coverage of the Business Glossary*, or of the *Data Dictionary*

ii. *Quality*: Evaluate the extent of Data Quality Management activities conducted, associated performance, and the added value obtained from these activities. Example: *Percentage of data passing controls*

iii. *Remediation*: Assess the completeness of anomaly remediation activities, the effectiveness of their management, and the business benefits derived from them. Example: *Time saved on rework*

iv. *Reporting*: Evaluate the scope of reporting within the Data Governance domain, the effectiveness and efficiency of the reporting process, and the benefits associated with this activity. Example: *Increase in effectiveness in communication with regulators/top management*

v. *Access and Usage*: Evaluate the extent of the Data Governance framework for business users, the support for data access and usage needs, and the benefits associated with these activities. Example: *Enabling of Analytics and AI capabilities*

III. *Value Metrics*: Metrics related to the benefits of Data Governance for

the company. The goal is to understand the value of data for the company and evaluate the contribution of Data Governance. These metrics are divided into 2 dimensions:

i. *Value of data for the company*: they interpret, measure, and communicate the value of data for the company, taking into account the potential uses of data and the underlying business objectives. Example: *Intrinsic value of a dataset for a specific data/business domain* (e.g., Customer, Product, etc.)

ii. *Value of data governance for the company*: they evaluate the contribution made by data governance in increasing or preserving the intrinsic value of data, relative to their dedicated purposes. Example: *The value of data governance activities in mitigating the risk of "decay" in the intrinsic value of a dataset and/or increasing that intrinsic value, promoting greater diffusion and pervasiveness* (i.e., defensive vs. aggressive approach)

Within the scope of this project, we focus on the perspective metrics of **Metadata and Data Model** and **Quality**, as they directly pertain to the activities we carried out in Chapter 3 (note that also the other metrics can be calculated starting from our implementation, but additional information regarding the organization's processes are needed).

4.2 Metadata and Data Model Metrics

We report here some of the metrics used for measuring the impact of data governance, from the point of view of metadata management.

- **Horizontal Business Data Lineage Coverage**: defined as the ratio between the number of Business Glossary (BG) data elements for which

a lineage path has been traced and the total number of BG data elements considered relevant for regulatory and strategic purposes.

$$\frac{\#BGDataElementsWithTracedLineagePath}{\#TotalBGDataRelevant}$$

A value equal to 100% (best) indicates that a Lineage path has been effectively traced for all relevant data. The calculation of the indicator needs the Business Glossary deliverable from the data governance framework.

- **Vertical Data Lineage Coverage**: defined as the ratio of the number of BG data elements for which a link has been defined to the physical data that represents them to the total number of BG data considered relevant, with the same meaning as before.

$$\frac{\#BGDataElementsForWhichIsDefinedLineageWithPhysicalData}{\#TotalBGDataRelevant}$$

A value equal to 100% (best) indicates that a linkage to physical data has been defined for all relevant data. As for the previous, the calculation of this indicator needs the Business Glossary deliverable from the data governance framework.

4.3 Quality Metrics

We report here some of the metrics used for measuring the impact of data governance, from the point of view of the data quality.

- **Data Coverage**: The percentage measurement of data coverage for critical data managed by the organization is defined as the ratio between the number of data elements considered critical and subject to controls and the total number of critical data elements of the organization:

$$\frac{\#CriticalDataElementsSubjectToControls}{\#TotalCriticalDataElements}$$

In this formula:

- *subject to controls* refers to data on which at least one control has been implemented
- *critical data elements* represents data that are considered important for the organization. Note that the concept of *criticality*, as well as its interpretation, may vary among different organizations based on the maturity level of the Data Governance framework

This indicator reflects the level of coverage of the bank's core data under control. The higher the value of the indicator, the more consistent the activity of monitoring and coverage of data considered more *critical* is. To calculate this indicator, two deliverables from the Data Governance framework are required: the Data Dictionary and the Data Quality Controls.

- **DQ Dimensions Coverage**: The indicator measures the percentage level of coverage of dimensions from the Data Quality (DQ) perspective (Completeness, Accuracy, Consistency, Integrity, etc.). A DQ dimension is considered controlled if it is covered by at least one control. It can be calculated as the ratio of the number of controlled DQ dimensions to the total number of DQ dimensions:

$$\frac{\#ControlledDQDimensions}{\#TotalDQDimensions}$$

We note that:

- The indicator can be calculated at various levels of aggregation (e.g., data-centric vs. contextual controls, or it-related vs. single-point controls, etc.).
- A DQ dimension is considered controlled if it is covered by at least one control.
- DQ dimensions may be defined differently in different contexts.

The higher the ratio, the more the definition and implementation of specific controls on various DQ dimensions are observed. Evaluation may involve the definition of appropriate thresholds. Required deliverables for its calculation are the data quality controls.

4.4 Approach to Results Interpretation

After defining and explaining the metrics used to evaluate the impact of the Data Governance Framework on the organization, a discussion of the results, i.e. the values obtained for the metrics, could contribute to a better understanding of their interpretation. Unfortunately, though, we cannot provide such results in this thesis due to the sensitive nature of the data involved. However, we would like to report a possible approach for the interpretation of the calculated metrics.

To effectively interpret and leverage the Data Governance implementation metrics for enhancing the Data Governance initiative within the organization, it is fundamental to consider the following aspects:

- Firstly, these metrics should not be viewed in isolation but rather as part of an ongoing process. Regularly tracking and analyzing the metrics over time can reveal trends and patterns, providing valuable insights into the effectiveness of the framework.

- Secondly, it is essential to establish clear benchmarks and goals for each metric, aligning them with the organization's overall data governance objectives. By setting specific targets and expectations, one can understand whether the initiative is meeting its intended outcomes. If certain metrics are falling short, it's crucial to conduct root-cause analyses to identify areas requiring improvement.

- Thirdly, communication and collaboration among all stakeholders are key. Sharing the metric results with relevant teams and individuals within the organization fosters transparency and accountability. It enables data governance teams to work collaboratively to address any issues and implement necessary enhancements.

Thus, the metrics should be continually computed, monitored, and analyzed with respect to some pre-established benchmarks. Insights should then be shared with key people involved in the organization's data assets in order to foster a culture of data governance and to ensure that the Data Governance initiative remains adaptive, effective, and aligned with the organization's evolving needs and priorities.

Chapter 5

Conclusion and Future Recommendations

Data Governance addresses the topic of formalizing an organization's data assets, with the aim of turning data into a valuable resource. In the Financial Services Industry, the need for formalization is further driven by regulations that have been issued after the 2007 financial crisis. This is why banks seek to integrate a data governance function as a part of their business.

To successfully establish data governance practices, a Data Governance Framework must be designed and implemented, according to a structured and exhaustive approach. Our proposed approach articulates three major activities to be carried out:

1. Data Discovery: this activity aims at understanding the general data landscape of the bank
2. Metadata Management: following the first, this activity serves as an enhancement of the data landscape understanding. This is achieved by

(1) tracing the path of critical data across the various bank components, both at a business processes level and at a technical systems/applications level (*Horizontal Data Lineage*), (2) by formalizing the business processes taking place in the bank (*Business Glossary*) and (3) connecting them with the technical ones (*Vertical Data Lineage*), eventually achieving a complete view of the organization.

3. Data Quality Management: this third activity complements the previous two in providing an understanding of the organization's data assets by measuring the quality level of the data (*Data Quality Management*).

Finally, to numerically quantify the impact of the data governance framework on the organization, some metrics can be defined, calculated, and periodically evaluated to assess the overall data governance maturity level of the organization, and to gradually integrate more and more practices aimed at enhancing the value of data.

As for further work directions, during our work, we identified the following possible needs for improvement:

- Starting with the Metadata Management activities, regarding the Data Lineage, we note how this activity is limited to tracing the *path* of data, ignoring the *transformations* it goes through (e.g., aggregations, disaggregations, functions, etc.). While having clear knowledge of the path of data is already of high value compared to having no information at all, in some situations this could not be enough, and tracing transformations could become critical, especially in the context of Machine Learning models, that could take lots of data from a source, apply very complicated functions and then only output a single number or string. Having information about some of the logic the data goes through could bring an enormous advantage.
- For what concerns the Data Quality Management, we specify how all the

data quality controls must be designed following a logic. For example, to identify data elements on which to put a completeness control (that we remember is a control of the presence and historical depth of all necessary data for analysis), one must first find out what data elements these are applicable to. Having to often deal with very large datasets, this task is time-consuming and, in a time-constrained project, leaves less time for the definition of more sophisticated controls, like the Coherence controls. Indeed, for time reasons, these kinds of controls are often limited to checking coherence between dates (e.g., the date of contract sign must be less than or equal to date of payment), but could encompass much more complicated logic, which could put light on some critical data quality problem. To address this, there are two possibilities: reducing the time needed on other controls or extending the project deadlines. Being the second option very distant from reality, to proceed with the first one, a way of automating the "easier" types of quality controls could be a solution to leave people more time to focus on more logic-driven controls.

- Finally, as a general aspect of the project, we note how all the activities illustrated in the project are carried out by a team of people, who have to manually process very big datasets. This aspect is mainly related to the textual form of the metadata (i.e., tables and column names are strings), and thus the difficulty in automating the processing. Nevertheless, with the emergence of new Large Language Models (LLMs), experimentations could be carried out on trying to automate the processing. Still, we think manual work will always be necessary in this field, as it is almost counterintuitive to bring exhaustive and rigorous knowledge about an organization's data assets without having people directly analyze them.

Bibliography

[1] Wikipedia, *Information system*, https://en.wikipedia.org/wiki/Information_system.

[2] Investopedia, *What Is Economic Capital (EC)? How to Calculate and Example*, https://www.investopedia.com/terms/e/economic-capital.asp.

[3] Tableau Articles, *Data Management: What It Is, Importance, And Challenges*, https://www.tableau.com/learn/articles/what-is-data-management.

[4] Basel Committee on Banking Supervision, *Principles for effective risk data aggregation and risk reporting*, January 2013.

[5] Data Management Association, *DAMA-DMBOK2: DAMA International's Guide to the Data Management Body of Knowledge.*

[6] Luca Robimarga, *Data governance: securing the future of financial services.*

[7] Nicola Ierinò, *Financial Services embrace Data Governance: a real case study on a domestic Bank Group.*

[8] Informatica, informatica.com/resources/articles/ *What Is Data Governance and Why Does It Matter?*, https://www.informatica.com/resources/articles/what-is-data-governance.html.

[9] Piotr Pietrzyk, *Implementation of the Data Governance / Data Services / Data Factory / Data Mesh / GRC - black magic or rather a straightforward approach?*, https://www.linkedin.com/pulse/implementation-data-governance-black-magic-rather-piotr-pietrzyk/.

[10] Christian Bruck, *Challenges and opportunities of Data Governance in private and public organizations.*

[11] Talend Resources, talend (a Qlik Company), *Data governance framework – guide and examples*, https://www.talend.com/resources/data-governance-framework/.

[12] bnova, *Un framework per la Data Governance 2.0*, https://www.bnova.it/data-governance/data-governance-framework/.

[13] sadasdb, *Data Governance: le nuove strategie per una migliore gestione e valorizzazione del patrimonio informativo aziendale*, https://www.sadasdb.com/data-governance-data-strategy-big-data/.

[14] atlan, *Data Governance Framework — Examples, Templates, Best practices & How to Create a Data Governance Framework?*, https://atlan.com/data-governance-framework.

[15] Data Management Association, *DAMA International - About Us*, https://www.dama.org/cpages/mission-vision-purpose-and-goals.

[16] Wikipedia, *Pareto principle*, https://en.wikipedia.org/wiki/Pareto_principle.

[17] Doug Wilson, Praxent (praxent.com/blog), *Dimensional and relational database modeling systems: The right tool for the job*, https://praxent.com/blog/dimensional-relational-database-modeling-systems-better-business.

[18] keydata, *Data Modeling - Conceptual, Logical, And Physical Data Models* https://www.1keydata.com/datawarehousing/data-modeling-levels.html.

[19] atlan, *Metadata Management and Data Lineage: How Their Synergy Enhances Data Understanding and Data Governance* https://atlan.com/metadata-management-and-data-lineage/.

[20] Medium, *The many layers of data lineage* https://medium.com/data-monzo/the-many-layers-of-data-lineage-2eb898709ad3.