

OTDK-thesis

Kristóf Balázs

Beyond Visualization: The Transformative Role of BI Systems in Enhancing ERP Capabilities

FACULTY OF ECONOMICS AND BUSINESS
ADMINISTRATION

BABEȘ-BOLYAI UNIVERSITY CLUJ-NAPOCA



Author:

Kristóf Balázs

Business Information Systems

Supervisors:

Dr. Dénes Csala

Lecturer

Dr. Imre Székely

Assistant Professor

Cluj-Napoca, 2024

Absztrakt

Az elmúlt évtizedben számos üzleti intelligencia (BI) eszköz funkciókészlete bővült, lehetővé téve a kis- és középvállalkozások (KKV-k) számára, hogy ezeket több üzleti folyamat automatizálási (BPA) helyzetben használják. A vállalati erőforrás-tervezési (ERP) rendszerek legtöbb strukturális modellje a BI-t csupán jelentéskészítési és adatvizualizációs eszközként tartalmazza, amely ezen rendszerek legújabb funkciókészletének fényében már nem teljesen pontos.

Szükség van egy olyan keretrendszerre, amely a BI eszközöket használja központi elemként, bemutatva azok képességeit a kezdetben tervezett felhasználási eseteken túl. Mivel ezen eszközök több mint 100 különböző adatkapcsolat-típust kínálnak, és több szolgáltató kínál API-alapú megoldásokat könyvelési és projektmenedzsment szoftverekhez, kiaknázatlan lehetőség rejlik ezen eszközök használatában, olyan megoldások építésére, amelyek a KKV-k specifikus igényeihez szabhatóak. Ezek a megoldások gyakran költséghatékonyabbak és rugalmasabbak, mint a nagy ERP szolgáltatók által kínáltak, még az implementációs költségek figyelembevételével is.

A keretrendszer megalkotásához a Microsoft Power BI és a Tableau, valamint más feltörekvő BI eszközök képességeit használok fel. Ez magába foglalja az adatforrásokhoz való hozzáférés lehetőségeit, valamint az adatok kinyerésére, átalakítására és betöltésére (ETL folyamatok), az adatmodellezés, a gépi tanulás (ML) integráció és a megosztás lehetőségeit. A végső keretrendszer általánosított, amely a két BI eszközre épített modellt használja alapul. Továbbá, hogy gyakorlati példákat biztosítsak a keretrendszerhez, esettanulmányként felhasználok az elmúlt két évben általam készített megoldásokat a Microsoft Power BI-ban (a) az NGI Systems cég belső használatára, valamint (b) a cég ügyfelei számára.

Megállapításaim szerint a BI eszközök figyelembevétele a BPA tervezés során megszokott gyakorlatnak kellene lennie a KKV-knál, mivel számos olyan lehetőséget kínál, amelyek a magas licencköltségek és a nagy ERP szolgáltatók által kínált megoldások bonyolultsága miatt megvalósíthatatlanok lennének. Ezenkívül, a BI eszközökre épülő átdolgozott modell, gyakorlati útmutatót nyújt számos szakember számára, az üzleti elemzőktől a BI szakértőkhig, segítve a vállalatokat abban, hogy költséghatékony, rugalmas és akár skálázható megoldásokat hozzanak létre.

Contents

Abstract	3
Nomenclature	4
List of Figures	5
1 Introduction	6
1.1 Motivation and Background	7
1.2 Research Questions	8
1.3 Methodology	8
1.4 Thesis Structure	9
2 Literature Review	11
2.1 Structural Models of ERP Systems	15
2.2 Factors Related to Successful BI System Implementation	18
2.3 The Feature Set and Capability of BI Tools	20
2.4 Synthesis / Gaps	23
3 The Development and The Current Feature Set of BI Tools	25
3.1 The History and Feature Set of Power BI	26
3.2 The History and Feature Set of Tableau	39
4 Constructing a BI Centred Framework	46
4.1 The Framework for Power BI	47
4.2 The Framework for Tableau	51
4.3 The General Framework	54
5 The Case Study of NGI Systems	56
5.1 The Power BI framework of NGI Systems	57
5.2 HR Compliance and Management Reports	59

CONTENTS

5.3	Production Management Reports	62
5.4	Financial and External Billing Reports	64
5.5	Performance Tuning, Error Handling and Documentation	65
5.6	Interview with the CEO of NGI Systems	67
6	Conclusions	70
	Bibliography	72

Abstract

In the past decade, the feature set of many business intelligence (BI) tools have expanded, allowing SMEs to leverage these in several business process automation (BPA) scenarios. Most structural models of enterprise resource planning (ERP) systems include BI as merely a tool for reporting and data visualization, which is not entirely accurate anymore given the recent development of these tools.

We need to develop a framework that uses BI tools as its core, showcasing their potential beyond their initial intended use cases. As these tools provide over 100 different data connection types and as several plugin providers offer API-based solutions for accounting systems and project management software, there is an untapped potential in using these tools to build solutions that can be tailored to the specific needs of SMEs. These solutions are often more cost-effective and flexible than the ones offered by large ERP vendors, even with implementation costs factored in.

I include the capabilities of Microsoft Power BI and Tableau, along with other emerging BI tools to construct the framework, including options for data sources, extract, transform, load (ETL) processes, data modelling, machine learning (ML) integration and deployment. The final model is generalized, constructed from the models built for each specific tool. Moreover, to provide practical examples for the framework, I use the solutions I have prepared in the past two years in Microsoft Power BI for (a) internal use for the company NGI Systems as well as for (b) their clients.

My findings indicate that the consideration of BI tools in BPA planning should be a common practice at SMEs as it opens many possibilities that would otherwise be unfeasible due to high licensing costs and the complexity of the solutions offered by large ERP vendors. Moreover, the reconstructed model with BI tools at its core provides a practical guide to a variety of professionals from business analysts to BI practitioners, enabling businesses to build solutions that are cost-effective, flexible, and even scalable.

Nomenclature

Abbreviation	Definition
AI	Artificial Intelligence
API	Application Programming Interface
BI	Business Intelligence
BI&A	Business Intelligence and Analytics
BIS	Business Intelligence System
BPA	Business Process Automation
CIO	Chief Information Officer
CRM	Customer Relationship Management
DAX	Data Analysis Expressions
DBMS	Database Management System
DSS	Decision Support System
DW	Data Warehouse
ELT	Extract, Load, Transform
ERP	Enterprise Resource Planning
ETL	Extract, Transform, Load
HR	Human Resources
IoT	Internet of Things
IT	Information Technology
KPI	Key Performance Indicator
ML	Machine Learning
OLAP	Online Analytical Processing
OLTP	Online Transaction Processing
RDBMS	Relational Database Management System
RLS	Row-Level Security
ROI	Return on Investment
RTBI	Real Time Business Intelligence
SaaS	Software as a Service
SCM	Supply Chain Management
SMEs	Small and Medium-Sized Enterprises
SQL	Structured Query Language
SRM	Supplier Relationship Management

List of Figures

2.1	Overview of the main chapters and their contents in the literature review. .	13
2.2	Typical BI architecture (adapted from [6]).	15
2.3	Anatomy of an Enterprise System (adapted from [19]).	16
2.4	Multi-tiered architecture of the 3 largest ERP vendors (adapted from [4]).	17
2.5	Integrating BI back into ERP systems (adapted from [6]).	18
2.6	Analytical maturity levels and the degree of using BI&A in decision making adapted from [3]).	20
2.7	Magic Quadrant for BIA Platforms (adapted from [7]).	22
3.1	Simplified architecture of Microsoft Azure and Fabric services (adapted from adapted from [34]).	27
3.2	Architecture of a DAX query execution (adapted from [28]).	32
3.3	Tableau Online architecture (adapted from [42]).	43
3.4	Tableau Server file types and architecture (adapted from [43]).	44
3.5	Tableau Server architecture approaches (adapted from [43]).	45
4.1	BI centered framework for Power BI tailored to SMEs.	47
4.2	BI centered framework for Tableau tailored to SMEs.	51
4.3	General BI centered framework tailored to SMEs.	54
5.1	The Power BI framework of NGI Systems.	58
5.2	Effective Hourly Rate business logic and calculated column.	60
5.3	HR compliance reports on Confluence.	61
5.4	Recursive exploded BOM Python code.	63
5.5	Sample pages from the ERP report.	64
5.6	Query plans and server timing of the old and the new optimized DAX code.	66
5.7	Interview question for the CEO of NGI Systems.	68

Chapter 1

Introduction

Over the past three decades, Business Intelligence (BI), has surged within both academic circles and the business sector, and in the last decade many new BI tools appeared on the market. The adoption of Business Intelligence and Analytics (BI&A) systems has mainly occurred in large enterprises and consequently, research in the BI&A field has primarily concentrated on these types of organizations [1]. Thus, very little case study-based practical research has been conducted on the capability of these BI tools, especially in the case of small and medium-sized enterprises (SMEs). A report by McKinsey [2] states that in OECD countries SMEs represent approximately 99% of businesses and are responsible for 70% of employment. In high-income nations globally, they contribute over 50% to the Gross Domestic Product (GDP). Despite the enthusiasm about BI&A, the practical challenge for numerous SMEs remains how to effectively utilize analytics to leverage their data [3].

Historically, the largest ERP providers such as Oracle, Sage and SAP included their proprietary business intelligence solutions as part of their ERP offerings. The responsibility for deploying these BI solutions within an organization depended on the expertise and abilities of either the ERP consultants or the customer's information technology (IT) team as solutions offered by large organizations, bundled with underlying ERP systems, were costly and challenging to implement [4]. In the BI&A 1.0 era [5], the results of the Extract, Transform, and Load (ETL) process were loaded into data warehouses (DW), which were the base for online analytical processing (OLAP) servers create a multidimensional view of data, supporting BI operations like aggregation, drill-down, and filtering [6]. Each of these steps can be an obstacle to users who are not data specialists.

The aim of this thesis is to explore the current feature set of Power BI and Tableau, the

two BI tools that are considered market leaders according to Gartner's industry advisory perspective [7] and to build a BI centred framework for both, considering their ability to execute many of the processes mentioned in the previous paragraph, while offering a user-friendly interface for many of these tasks. A case study approach is used to demonstrate the effectiveness of this framework, which uses the solutions I have prepared in the past two years in Microsoft Power BI for (a) internal use for the company *NGI Systems* as well as for (b) their clients.

1.1 Motivation and Background

Following the era of BI&A 3.0 [5], many aspects of the BI processes became much more accessible to SMEs as the cost of acquiring and storing large amounts of data has been declining [3]. SMEs need to leverage this phenomenon to create decision support systems (DSS) and use the data they have in certain business process automation (BPA) cases. As BI&A research was conducted in an enterprise context in most cases, there is a gap in researching how the traditional paradigms of ERP systems, BPA and BI are being reshaped by the advanced capabilities of BI tools.

Research regarding BI software is also lacking, as most of them focus on comparing the tools themselves [8, 7] and only a few mentions specific capabilities or implement out of the box solutions using the options available in each tool [9].

Creating a framework for the use of these tools, with a variety of examples highlighted in the case study provides a practical guide to a variety of professionals such as BI consultants, managers, CIOs, business analysts and BI practitioners, enabling small businesses to build solutions that are:

- Cost-effective: factoring in licensing (Tableau and Power BI licensing costs are included in Chapter 3.1 and 3.2), maintenance, and implementation costs.
- Flexible: changes in the business logic can be easily implemented as most of these solutions are custom built.
- Scalable: BI tools can handle large volumes of data, and ETL, data modelling and tool specific code (e.g. DAX for Power BI) can be optimized (Tableau and Power BI optimization is discussed in Chapter 3.1 and 3.2).

1.2 Research Questions

The objective of this research is to explore the feature set of BI tools and showcase DSS, BPA, and small-scale ERP solutions that can be achieved with the use of these tools. Moreover, the use of a framework is important to generalize these solutions as they might not seem intuitive at first sight, and they can be very specific in their implementation details. To achieve this objective and to address the research gaps highlighted in the previous section, the following research questions are formulated:

1. *In what ways do BI tools contribute to the decision-making and BPA processes within SMEs, and how can these contributions be streamlined through a BI-centred framework?*
2. *What are the key factors that determine if a BI-centred approach can be used successfully for a given set requirements and what are the limitations of these solutions?*

1.3 Methodology

This research employs a qualitative case study-based approach to demonstrate the applicability of the framework and the capability of Microsoft Power BI with practical examples. In more narrow terms, the case study type is exploratory as it aims to reveal and explain the assumed causal relationships in real-world interventions, which are too intricate for survey or experimental approaches, by connecting implementations with effects [10]. Summarizing complex case study findings clearly is crucial for researchers if they aim to engage readers in the research context as though they were part of the study and helping them assess the applicability of findings to their own contexts [11]. Thus, the business requirement for each solution, along with the tools and the initial setup needed is outlined before delving into them.

A qualitative case study-based approach is the most appropriate in answering the research questions as there is a lack in practical research of BI tools in the case of SMEs [1] and most of the solutions using the methods discussed are custom built, thus it is difficult to present an out of the box method that works for most business requirements.

The case study goes over seven internal and external solutions. These solutions were developed over the last two years while I was working in the position of Power BI

Developer at the company *NGI Systems*. An interview with the co-founder and CEO of the company is conducted to accurately assess the impact of the solutions. This provides qualitative data on how they changed business processes at the company, what effects did the external reports have on overall company performance and transparency and what were the main challenges that occurred during the planning, implementation and maintenance phases.

At the cornerstone of this research are qualitative methods. Thus, the first phase involves using a structured approach to collect all relevant knowledge about (a) BI processes, (b) structural models of ERP systems, (c) success factors in BI implementation and (d) the capability of Power BI and Tableau. In the case of the latter, comprehensive analysis of books, documentation and community forums are used as there is very little information in scholarly papers or industry reports. In the case of the first three, research papers, articles and reports are used for the most part as they document the technological development of the processes accurately, along with the economic and business implications of these developments.

1.4 Thesis Structure

The thesis consists of four core chapters, and it begins with the literature review. In the first section, the history of BI processes is presented, and each term used in BI, and each major phase of BI&A [5, 12] is discussed in detail. The second section continues with the history of ERP systems, looking at how the structural models, the offerings of large ERP vendors, and the architecture of these systems has evolved in the past three decades. The third section in the literature review focuses on the success factors related to BIS implementation, detailing the correlation between data-driven solutions and company performance, analytics maturity, and best practices in BI adoption. The penultimate section goes over the main capabilities of BI software, detailing the two most dominant market leaders, Power BI and Tableau. This overview serves as an introduction to understanding the shift to the service-oriented architecture (SOA) of BI tools, the changes in company culture at SMEs and open-source mentality of many companies that are offering BI software.

The first core chapter after the literature review (Chapter 3) demonstrates the feature set of Tableau and Power BI, going over their development phases since they were first introduced. The capabilities of each tool is presented based on their ability to handle ETL

processes, data volume, data types, data connectors, data visualization and both governance and scalable architecture types in the service. The ability to execute more complex solutions in each tool is presented with examples from research papers, books, community projects and from the case study that is discussed later in its own chapter. This chapter is more technical, as it does not provide a holistic view of each solution and it discusses only certain components of them. Its purpose is to showcase every component of the tool that one needs to know about if they want to build scalable and cost-effective solutions using each tool. The next chapter focuses on drawing a more complete picture of the solutions discussed here.

Chapter 4 builds on the previous two chapters to create a BI centred framework for building BPA and other mini-ERP solutions. First, tool specific frameworks are showcased for Tableau and Power BI, and the contents of each element in the framework are elaborated. At the end, a general framework is displayed, which can be used for not only Power BI and Tableau, but also for other BI software.

The last core chapter (Chapter 5) presents the case study of NGI Systems. A total of seven solutions are showcased, which can be grouped into four categories: production management, HR compliance and management, financial reporting, and external billing reporting. First, the business requirements with the initial business logic are presented, along with the setup necessary to perform the given requirements. The setup requirements are put into the Power BI centered framework discussed in the previous chapter to demonstrate its applicability. This chapter is concluded by an interview with the CEO and co-founder of NGI Systems about the process of creating, building, and maintaining these solutions as well as their impact on the firm.

Chapter 2

Literature Review

The systematic literature review covers four major topics: the history and development of BI processes, structural models of ERP systems, factors related to successful BI system implementation and the feature set and capability of BI tools. It defines the scope of the research by linking previous studies, theories, and models to the research objective. The description of each topic, along with the citation of the sources corresponding to the given topic is included in the table in Table 2.1. Note that the table only considers journal articles, books, and conference papers and it does not consider other sources such as BI software documentation or blog posts and other types of articles. One source can appear in multiple categories and sources that do not correspond to one of the four categories are not included.

In 1865, Richard Miller Devens introduced **the term 'Business Intelligence'** (BI) in his book 'Business Anecdote Cyclopedia,' using it to describe how a business financier leveraged information for business purposes [8]. The initial concept of Business Intelligence (BI), as proposed in 1958 [13] in the IBM Journal of Research and Development, was an automatic system designed for disseminating information within any industrial, scientific, or government organization. Thus, literature for the evolution of BI is investigated between 1958 and 2020, with the main events highlighted on Figure 2.1.

The development of Business Intelligence and Analytics (BI&A) across three distinct stages is outlined in [5, 12], with the first stage (**BI&A 1.0**) beginning in the 1970s. This era marked the utilization of relational database management systems (RDBMS) and data warehouses (DW) for storing transactional data, employing statistical and data mining techniques for analysis, with the first OLAP product being introduced in 1975 by Information Resources. The architecture of the first BI systems is discussed later in this

Section	Category Name	Description	Authors	Total
2.1	The History and Development of BI Processes	The history of BI processes and major developments in the field, considering the past three decades, along with each term used in BI: BI, BI&A, BIS, DW, DSS, ETL, OLAP, OLTP, RDBMS, RTBI. Each major phase of BI&A is discussed in detail.	(Luhn, H. P., 1958); (Chou et al., 2005); (Dean and Ghemawat, 2008); (Chaudhuri et al., 2011); (McAfee et al., 2012); (Tutunea and Rus, 2012); (Chen et al., 2012); (Davenport, 2013); (Ransbotham et al., 2016); (Llave, 2017); (Puklavec et al., 2018); (Talaoui and Kohtamäki, 2020); (Eggert and Alberts, 2020); (Nwokeji et al., 2021)	14
2.2	Structural Models of ERP Systems	The history of ERP Systems. Development and changes in structural models, ERP architecture and offerings of large ERP vendors.	(Davenport, 1998); (Yen et al., 2002); (Chou et al., 2005); (Caserio, 2011); (Russman et al., 2017); (Amini and Abukari, 2020)	6
2.3	Factors Related to Successful BI System Implementation	Success factors related to BIS implementation and best practices in BI adoption. Correlation between the implementation of data-driven solutions and company performance, analytics maturity stages.	(Davenport, 1998); (McAfee et al., 2012); (Chen et al., 2012); (Davenport et al., 2012); (Davenport, 2013); (Ransbotham et al., 2016); (Russman et al., 2017); (Puklavec et al., 2018); (El-Adaileh et al., 2019); (Talaoui and Kohtamäki, 2020); (Lin et al., 2022); (Hafeez, 2023)	11
2.4	The Feature Set and Capability of BI Tools	Main components of BI software, detailing Power BI and Tableau. Includes research on optimization, handling of unstructured data, AI integration and BI architecture.	(Russo and Ferrari, 2016); (Edge et al., 2018); (Sleeper, 2018); (Russo and Ferrari, 2019); (Puls and Escobar, 2019); (Sleeper, 2020); (Vashisht et al., 2021); (Magoma et al., 2021); (Srivastava et al., 2022); (Ji and Tia, 2022); (Hafeez, 2023); (Rad, 2023); (Schlegel et al., 2023)	13

Table 2.1: Overview of the main chapters and their contents in the literature review.

section.

The second phase, **BI&A 2.0** started with the internet's rise in the early 2000s, focusing on unstructured data from social media and web analytics, which laid the foundation for new business models and created a demand for text and web mining capabilities [12]). The 2000s also witnessed the introduction of web-based applications and, by 2006, the adoption of cloud BI solutions started, enhancing data visualization and accessibility while providing cost, scalability, and speed efficiencies [8].

In the third phase, **BI&A 3.0**, the focus was on analysing data from mobile and sensor-based devices [5]. Advancements continued into the late 2010s with the rise of augmented analytics and AI, focusing on natural language processing (NLP) and data governance. Entering the 2020s, BI trends have included agile, cloud, and embedded BI, along with real-time analytics [8].

Evolution of Business Intelligence		BI & A Phases
1958	Hans Peter Luhn (IBM set word BI)	BI&A 1.0 (DBMS-based, structured content) • RDBMS & data warehousing • ETL & OLAP • Dashboards & scorecards • Data mining & statistical analysis
1960	Computer- Data Adventure	
1970	Big players into the market	
1975	Streamlining BI Research	
1988	International Conference	
2000	First Wave Applications	BI&A 1.0 (Web-based, unstructured content) • Information retrieval and extraction • Opinion mining • Web analytics and web intelligence • Social media analytics • Social media network analytics • Spatial-temporal analysis
2006	Cloud BI	
2007	Big Data Rising	
2010	In- Chip Technology famous	
2016	Sophisticated BI Tools	
2017	Cloud Application	BI&A 1.0 (DBMS-based, structured content) • Location-aware analysis • Person-centered analysis • Context-relevant analysis • Mobile visualization & human computer interaction
2018	Rise of Augmented analytics	
2019	Rise of Artificial Intelligence (AI)	
2020	Rise of Agile BI, Embedded BI	

Figure 2.1: Overview of the main chapters and their contents in the literature review.

Business Intelligence (BI) architectures within enterprises include a wide range of technologies and processes designed to manage and analyse data effectively, as depicted in Figure 2.2 [6]. Preparing data for BI tasks involves the Extract, Transform, Load (ETL) process, a set of operations for handling data that comes from a variety of sources and has many different formats. The main components of the ETL process are the following:

- The "**Extract**" component of ETL is the initial step, where data is collected from multiple internal and external sources, including but not limited to operational databases (such as ERP or SCM systems), IoT devices and social media platforms. This phase focuses on retrieving the necessary data without changing its content [14].
- During the "**Transform**" phase, this extracted data is cleansed to remove redundancies and inconsistencies, normalization to make sure that data is in a consistent format (the denormalization comes later), and application of business logic to make the data coherent and suitable for analysis. In simple terms, this stage converts raw data into a structured form, ready for analysis [8].
- The "**Load**" stage moves the transformed data into a DW or data mart, which ensures scalable retrieval for the BI processes that are further downstream.

A variety of digital interactions and devices generate vast, unstructured data, providing valuable insights for analysis despite being complex to manage [15]. These ETL processes are very important in addressing and tackling the challenges posed by **big data**, characterized by **5 Vs**: volume, velocity, variety, veracity, and value. Traditional RDBMSs are often very inefficient for this task. Besides the need for persistent storage, this was the main need for the integration of more advanced technologies capable of handling the complexities of big data [14]. Systems based on the MapReduce model and other big data technologies are used to manage the scale and complexity of enterprise data. These technologies are used to support the execution of complex SQL-like queries and other analytical operations on large datasets [16].

Further downstream in the BI processes, the architecture includes mid-tier servers designed to support specific BI functionalities. **Online Analytic Processing (OLAP)** servers present a multidimensional view of data, which is the foundation for operations such as filtering, aggregation, and pivoting, necessary for most BI tasks. **Data mining** engines can perform advanced data analysis beyond the capabilities of OLAP and reporting servers, enabling the development of predictive models to forecast a variety of scenarios, such as the customer reactions to marketing initiatives. **Text analytic** engines analyse large volumes of text data, such as customer feedback, to extract insights, like frequently discussed topics. This eliminates the need for manual effort, which is very extensive. [6].

Frontend applications are interfaces through which users interact with BI systems. This lets users perform a variety of BI tasks such as data visualization, performance track-

ing, and putting together ad-hoc queries. These tools play a vital role in allowing decision-makers to dynamically explore data, identify trends, and make data driven strategic decisions [6].

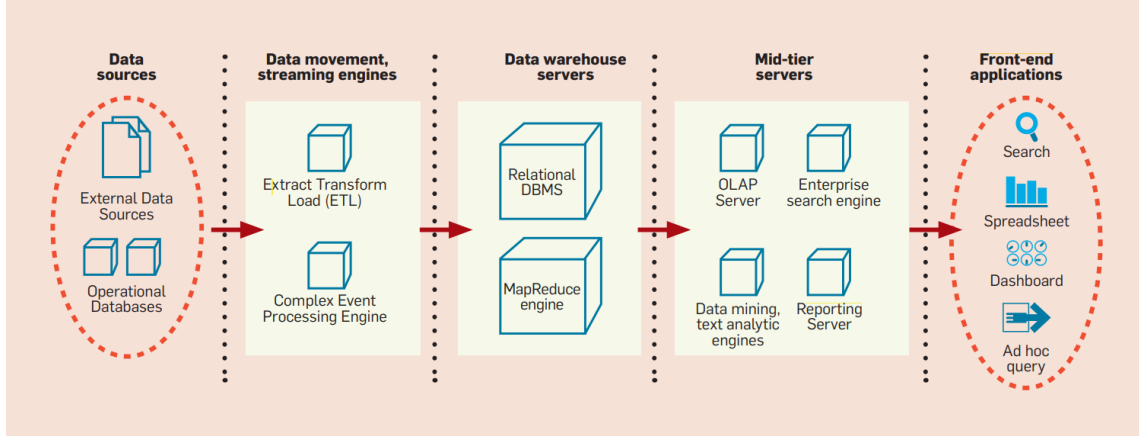


Figure 2.2: Typical BI architecture (adapted from [6]).

The research on **BI&A within SMEs** reveals a gap, as most studies concentrate on larger corporations rather than SMEs. There is a need for more industry-specific studies to enhance the integration of BI&A in SMEs, given its potential to improve decision-making processes [1]. SMEs tend to rely on internally developed tools or straightforward analytical solutions like spreadsheets (MS Excel, Open Office Calc, Lotus 1-2-3, etc.), which reflects a trend towards adopting Software as a Service (SaaS) BI solutions that are cost-effective and require minimal training [17].

The challenge for SMEs goes beyond just using BI&A tools. It involves the need for significant investments and changes in company culture to make the most out of analytics. This challenge is about expanding the skills of managers at SMEs to make decisions based on a combination of data and intuition [3]. This is discussed in more detail in section 2.3. Research emphasizes the need for BI&A solutions that can make the adoption process for SMEs simpler, highlighting the importance of making these tools cost-effective and functional for small businesses.

2.1 Structural Models of ERP Systems

By 1997, Over 20,000 companies globally invested billions in ERP system installations, with around 70% of Fortune 1000 firms having implemented ERP solutions for key business areas such as manufacturing, finance, and human resources [18]. One of the first

enterprise system architectures, the ‘Anatomy of an Enterprise System’, conceptualized in 1998, is centred around a **central database** that consolidates data and interfaces with modules that are specialized. Each of these modules is responsible for one distinct business area such as finance, sales, manufacturing, and human resources [19]. This model introduces the monolithic nature of ERP systems, where the focus is on data centralization for transaction management and reporting [19]. Each module's interaction with the central database is designed to be uniform, with the goal of maintaining data integrity and supporting the enterprise’s activities, which are often interconnected. This is also known as having a Single Source of Truth (SSOT). The ‘Anatomy of an Enterprise System’ is visible on Figure 2.3.



Figure 2.3: Anatomy of an Enterprise System (adapted from [19]).

Initially, the focus of ERP systems was on transactional capabilities, focusing on daily business operations such as recording data, managing transactions, and generating reports. However, this structure was rigid, and limited the flexibility required for more complex data analysis [20]. In response, BI systems were introduced to extend the functionality of ERP systems by adding layers of BI&A, aimed at enhancing decision-making processes [21]. The ability of a DW to store historical data is very important. For example, consider a company that sells shoes and does not have a specific model on sale anymore, its historical sales can be retrieved from a DW, which you cannot do with a transactional RDBMS. Early integration strategies varied: some companies first defined their BI requirements and then customized their ERP systems accordingly, while others implemented a standard ERP system followed by a bespoke BI system to interpret the data [21].

Modern ERP architectures are primarily multi-tiered, with separate layers for data management, application processing, and user interface (UI) operations. These systems are designed to support business processes and typically include a central database, business logic encapsulated in application servers, and a web-based UI for accessibility [4]. Service-Oriented Architecture (SOA) is integral, dividing business logic into services for modularity and reuse. Cloud computing has become a major component, offering ERP solutions as scalable services over the internet, which aligns with the industry's continuous move towards SaaS models. These modern ERP systems are more flexible in deployment, whether on-premises or cloud-based, to suit various business needs and integrate with technologies like IoT and mobile applications [4]. Three examples of modern multi-tiered ERP architecture offered by the three largest ERP vendor (SAP, Sage, and Oracle) are visible on Figure 2.4.

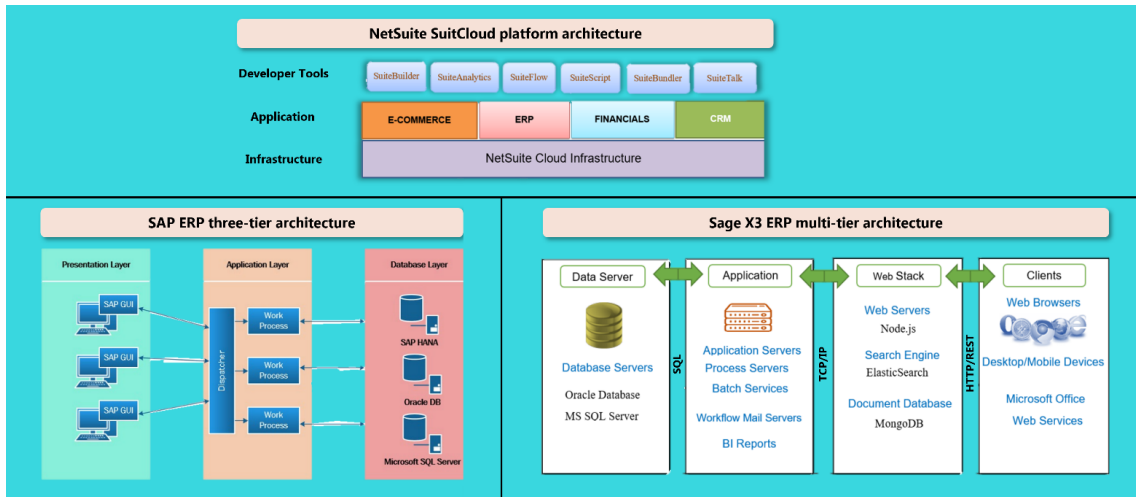


Figure 2.4: Multi-tiered architecture of the 3 largest ERP vendors (adapted from [4]).

There is very little research on **integrating BI back into ERP systems**. However, this process has the possibility of enhancing the operational level of enterprise systems by enabling real-time decision-making directly within the ERP systems. This enhances the operational efficiency and responsiveness of an organization [22]. Structurally, this integration requires both a robust and dynamic architecture that can handle the flow of information from BI tools back to the transactional systems of an ERP. This feedback loop improves and streamlines business operations in many cases [22].

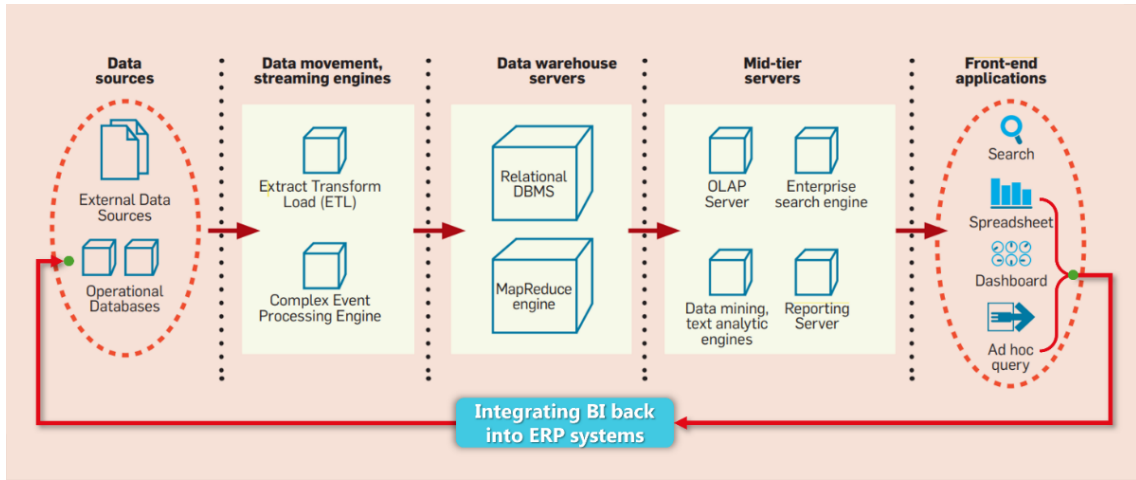


Figure 2.5: Integrating BI back into ERP systems (adapted from [6]).

2.2 Factors Related to Successful BI System Implementation

There are many success factors related to successful BI implementation, and there are many approaches to adoption both in the case of enterprises and SMEs. This section collects these, going in chronological order, highlighting crucial and often recurring factors.

It is highlighted that the successful implementation of ERP systems relies on a firm's ability to **align the technical capabilities of the system with its business requirements** [19]. It emphasizes the risk of ERP having a rigid structure that may not align with a company's unique strategic goals. This issue was present in most ERP systems before the 20th century. It has been suggested that benefiting from the big data wave demands a shift in decision-making culture towards data-driven approaches and the **acquisition of new talent skilled in data science** [15], which are crucial before integrating such approaches into BI systems and addressing the early issues associated with big data, as mentioned in section 2.1. This is the main reason why the data scientist role was named 'The sexiest job of the 21st century' [23] as there was a lack of data professionals capable of handling these challenges compared to the demands of the industry.

In the era of BI&A 3.0, highly regarded articles on the topic [5, 12] highlight the following success factors related to BI&A implementation, focusing solely on the business implications (with strategic alignment and talent recruitment excluded, as they are mentioned earlier):

1. **Data-Driven Mindset:** Organizations should prioritize data in decision-making, adapting to the new real-time, predictive tools.
2. **Data Management:** High-quality data and strong governance, given the diverse data sources like mobile and IoT sensor data.
3. **Technology Integration:** An infrastructure capable of handling big data, leveraging cloud computing and scalable analytics platforms.
4. **User-Friendly Tools:** Interfaces should be accessible to non-technical users to allow easy interaction with the data.
5. **Collaboration:** Joint efforts are required across IT, analytics, and business units.
6. **Agility:** Businesses must be able to adapt to the rapidly evolving BI&A field.
7. **Legal Compliance:** Companies must understand data privacy and protection laws.
8. **ROI Measurement:** Establishing clear metrics to measure the success of BI&A investments to justify the implementation costs involved.

Companies' success with analytics varies significantly across **three maturity levels: Analytically Challenged, Analytical Practitioners, and Analytical Innovators** [3]. These levels show an organization's ability of using analytics for competitive advantage (which is increasingly harder to achieve due to increased market adoption) and innovation as visible on Figure 2.6. The figure also indicates that among organizations with a higher level of analytical maturity, managers rely more heavily on analytics than on intuition compared to less mature organizations. The study concluded that only a fraction of the firms has built a strategic analytics plan to achieve their analytical objective, which was determined to be the most important factor in successful BI implementation. [3].

A study of 38 articles [24] identified ten success factors that appeared in most of the articles reviewed: management support, data source system, organisational resources, IT infrastructure, vision, champion, team skills, project manager, user participation, and change management. The two aspects not discussed before are the presence of a project champions and the degree of user participation. A **project champion** with leadership skills, deep knowledge of business processes (preferably in an executive role) and understanding of technological innovations is crucial for the successful implementation of BI. This is discussed in depth in the case study at section 6.6. **User participation** refers to involving users in the development and use of BI systems, which ensures that the BI system aligns well with the organization's operations and analytical goal. Moreover, it facilitates the use of these systems in the long run.

In the case of **SMEs**, only one study focuses on the success factors in BI adoption.

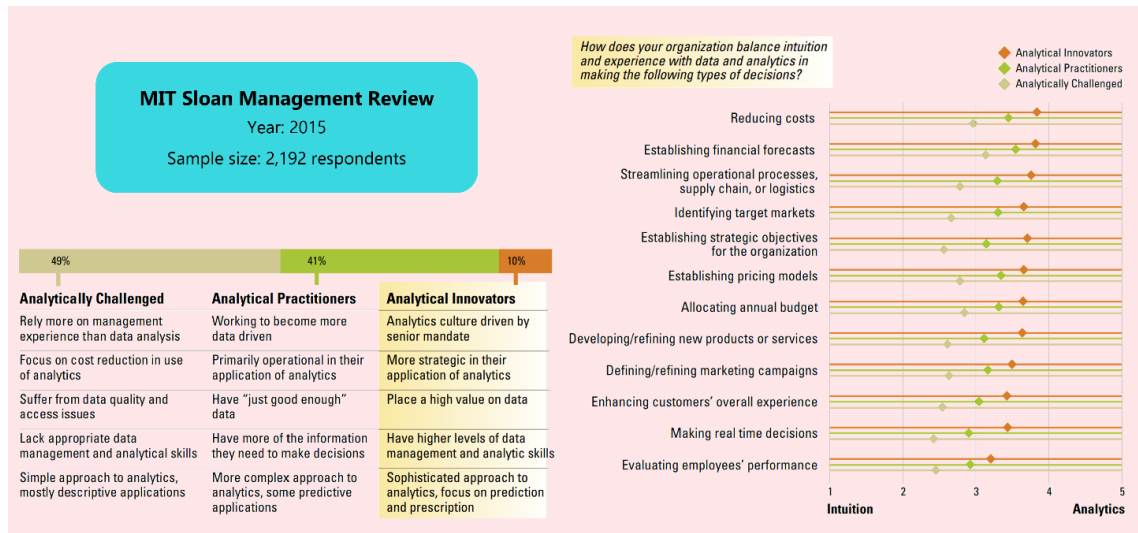


Figure 2.6: Analytical maturity levels and the degree of using BI&A in decision making adapted from [3]).

This study on Business Intelligence System (BIS) adoption in SMEs [25] highlighted several significant determinants across different adoption stages. The study used hypothesis testing of 11 hypotheses. Out of these, **management support**, the role of a **project champion** (throughout the adoption process), and **organizational readiness** (particularly in the early stages of adoption) were identified as crucial factors influencing BIS adoption. Furthermore, the **integration of BIS within ERP systems** and the **quality of the organizational data environment** were identified as determinants that affect adoption and use stages, but these were only partially supported [25].

Before concluding this section, it is worth noting that, most of the current BI research primarily explores how BI delivers actionable intelligence rather than the organizational context influencing its success [26].

2.3 The Feature Set and Capability of BI Tools

This last section explores the current research on BI tools, with a focus on Power BI and Tableau, which are explored in the beginning. As the research on the specific BI tools is lacking (they are reviewed in the end of the section), most of the sources used in this section are books.

As technical details about the feature set of **Power BI** are mostly found in documentation and community forums, there are only a few renowned books on the topic. The book 'Introducing Microsoft Power BI' is an introductory guide to utilizing Power BI for

BI&A. It provides a high-level overview of tool's capabilities, from data modelling and report creation to features like custom visualizations and integration with various data sources [27]. 'The Definitive Guide to DAX: Business intelligence with Microsoft Power BI, SQL Server Analysis Services, and Excel' is a detailed book on DAX (Data Analysis Expressions). It covers the basics of DAX, namely table functions, evaluation contexts, CALCULATE and CALCULATETABLE, variables, time intelligence, filter context and hierarchies. Later, it details the preliminary knowledge on DAX optimization, such as the DAX engines and query plans [28]. 'Pro Power BI Architecture' explores developing, deploying, sharing, and securing Power BI solutions. It covers a lot of topics like data import methods, DirectQuery, live connections, composite models, dataflows, shared datasets, data marts, paginated reports, and real-time streaming datasets, along with integration with Excel and performance tuning. The book also goes over deployment and collaboration, detailing the Power BI service, Power BI Report Server, gateways, licensing, administration, sharing methods, workspace organization, security features like row level security (RLS). This book provides the most detailed information on the capability of Power BI [29].

As the documentation of **Tableau** is very detailed, there are only two distinguished books, one aimed for beginners, the other is for advanced users. 'Practical Tableau' is a beginner's guide on Tableau for data visualization that starts with basic principles and gradually moves to more advanced chart types and visualization techniques. It covers topics such as data preparation, dashboard design, and storytelling with data, while also giving into practical tips for enhancing visualizations, like creating dynamic charts, custom maps, and implementing effective colour schemes [30]. 'Innovative Tableau' delves into more advanced functionalities like parameter and set actions, new mapping capabilities, and dynamic visualization techniques. It explores the utilization of Tableau Extensions and offers a guide on using Tableau in conjunction with other products for advanced analytics. Key topics include custom geographic maps, trellis, and tile maps, diverging bar charts, performance indicator titles, and integrating Google Forms for commenting systems [31].

There is only a limited number of papers detailing the feature set and capabilities of BI tools. Firstly, there are the **comparative** ones. A comparative study of fifteen BI tools, namely Zoho Analytics, Sisense, Power BI, Board, Tableau, Looker, Datapine, Microstrategy Analytics, Klipfolio, Birst, ClicData BI, kyubit BI, Dundas BI, SAP Business Objects BI, and Cluvio BI only compared the tools based on user reviews, and

although the comparison of the technical details were mentioned, no conclusions were drawn from it. Moreover, it was mentioned that there is now a possibility for SMEs to “deploy BI for the first time to address diverse needs” [8]. A yearly study by Gartner, ‘Magic Quadrant for Analytics and Business Intelligence Platforms’ compares BI tools based on ‘Completeness of Vision’ and the ‘Ability to Execute’. For 16 consecutive years, Microsoft (Power BI) has been positioned as a leader as visible on Figure 2.7. This is mainly attributed to its integration with Microsoft 365, Teams, and Azure, competitive pricing, features including metric-tracking, and low-code data mart [7].



Figure 2.7: Magic Quadrant for BIA Platforms (adapted from [7]).

Only one research paper was found that detailed the **application of a BI tool’s feature set specifically for SMEs**. The research focused on the IBM Cognos tool, highlighting features like data visualization, access to information, and data encryption are valued for decision-making processes. While SMEs recognize the benefits of BI tools, their adoption is influenced by the tools' affordability and their capacity to provide consistent, intuitive, and cost-effective solutions [32].

The **integration of AI and BI** was studied in the context of Power BI. A case study from Microsoft Research outlines the integration of AI with Power BI to analyse unstructured data, including text and metadata from social media, news, and cyber intelligence. The approach involves creating back-end pipelines that use AI to structure unstructured

data and developing front-end visualizations founded in Visual Analytics [9]. These out of the box solutions are available at Power BI Solution Templates. There was only one study exploring predictive analytics with Power BI through integration with Python, specifically employing linear regression and Support Vector Regressor (SVR) algorithms to forecast sales data [33]. However, paper is quite limited and only explores two simple machine learning (ML) algorithms. It is important to note that entry level ML algorithms are much easier to access thanks to the availability of certain ML libraries in Python.

2.4 Synthesis / Gaps

The systematic literature review presented investigates the evolution, structural models, success factors, and capabilities of Business Intelligence (BI) systems, while going over their development from the late 20th century to the latest advancements such as the integration AI for predictive analytics. It showcases the history of BI from its beginnings, focusing on key milestones across three distinct phases: BI&A 1.0, 2.0, and 3.0, transitioning from RDBMS and DW to the incorporation of unstructured data and AI technologies. It explores the structural models of ERP systems, highlighting a transition from monolithic, transaction-focused architectures to modern, multi-tiered, cloud-based solutions that facilitate real-time decision-making.

Key factors contributing to the successful implementation of BI systems are identified, including the need of a data-driven mindset, data management, technology integration, user-friendly tools, collaboration, agility, legal compliance, and clear ROI measurement. The literature review also addresses the challenges and requirements for SMEs adopting BI solutions, highlighting the importance of affordability, ease of use, and the role of leadership in the creation of a data-centric culture.

Finally, the feature set and capabilities of BI tools are mentioned with a focus on Power BI and Tableau, and their development to support advanced analytics, real-time data processing, and integration with machine learning algorithms like linear regression and SVR through Python for predictive analytics.

Despite everything that is mentioned above, there is still a **clear gap in the literature**. As most of the studies from the past focus on enterprises, especially before 2015, there is little practical, case study-based information mentioned about SMEs, even in literature reviews collecting them [1]. Moreover, many papers on SMEs focus on the economic and business side of BI adoption and do not provide a practical framework, with the feature

set of these tools. Thus, there is a clear to specifically address the application of BI tools' features for SMEs, especially for the market leaders Tableau and Microsoft, for which during this literature review, there was no such study found.

Chapter 3

The Development and The Current Feature Set of BI Tools

Since the introduction of the first BI tools in the 1970s, the feature set and the targeted user group of BI software have expanded. The most remarkable growth happened in the past decade, where many tools appeared on the market. Most of these modern BI tools offer a user friendly and intuitive UI, making it easier for users with less technical knowledge to leverage many of their features, which is especially important in the case of SMEs, where human capital is a scarce resource and outsourcing is not a financially viable option. However, many aspects of BI tools require extensive knowledge to use them to build more complex solutions such as mini-ERP systems, complex DSSs, BPA solutions and dashboards with predictive analytics. Moreover, data governance, access control, the correct flow of data and efficient data models are often overlooked and can cause major performance and safety issues. If SMEs want to leverage these BI tools, they need to have a solid understanding of the tool's capabilities. The framework that is discussed later aims to fulfil this exact purpose.

In this chapter, we explore the capabilities of the two most dominant market leaders, Power BI, and Tableau. I go over the broader architecture, talk about their components such as data connector types, custom connector options, ETL solutions, data modelling, downstream calculations, model, and calculation optimization, built in and custom visuals, documentation, community forums, AI integration, licensing, and limitations. After presenting the tools, I showcase solutions that are pushing their limitations from books, research papers, blog posts and community forums. The solutions that I built are only be showcased in the case study sections. After presenting the two tools in detail, other

emerging BI tools are also briefly discussed.

Note that this chapter is more technical, as its goal is not to provide a holistic view of each solution, but to discuss the components of tools, giving a broader overview of their capabilities. These capabilities serve as a foundation for the upcoming section, where whole solutions are showcased, focusing on specific ERP and BPA capabilities.

3.1 The History and Feature Set of Power BI

Although the main tools that Power BI (PBI) contains have existed before its appearance (e.g. Power Query), Power BI's public release was in 2015. Since then, the tool has maintained its position as a market leader [7]. This section explores the broader architecture of Power BI, then it discusses its components and limitations and finally, it showcases solutions that push the tool to its limits.

When discussing the broader architecture of PBI, it is impossible to not mention Azure. Since this section focuses solely on the capabilities of PBI, the architecture of Azure and the newly released Microsoft Fabric are only discussed. The differences between Azure and Fabric are visible on Figure 3.1.

Azure's structure has a wide range of services including Azure Data Lake for data storage across lifecycle stages, Azure Synapse Analytics for data processing, and integration capabilities with Power BI for visualization and analytics. A lot of companies face challenges with data silos, multiple copies of data, varied data formats, and complex system integration due to their organically grown architecture. **Fabric** aims to solve these issues by centralizing data in OneLake to eliminate silos and ensure that only one dataset copy exists (SSO). Fabric standardizes all tabular data in a format known as Delta Parquet to solve data integration and format uniformity issues. As an example, it offers a unified user experience similar to Microsoft 365. It simplifies access control and security with a single model, and has a simplified billing and licensing structure, which is separate in the case of the services offered by Azure [34] Since scope of this thesis is to explore the impact of BI solutions in the case of SMEs and these solutions are targeted for enterprises, Microsoft Fabric and Azure are not discussed in more detail.

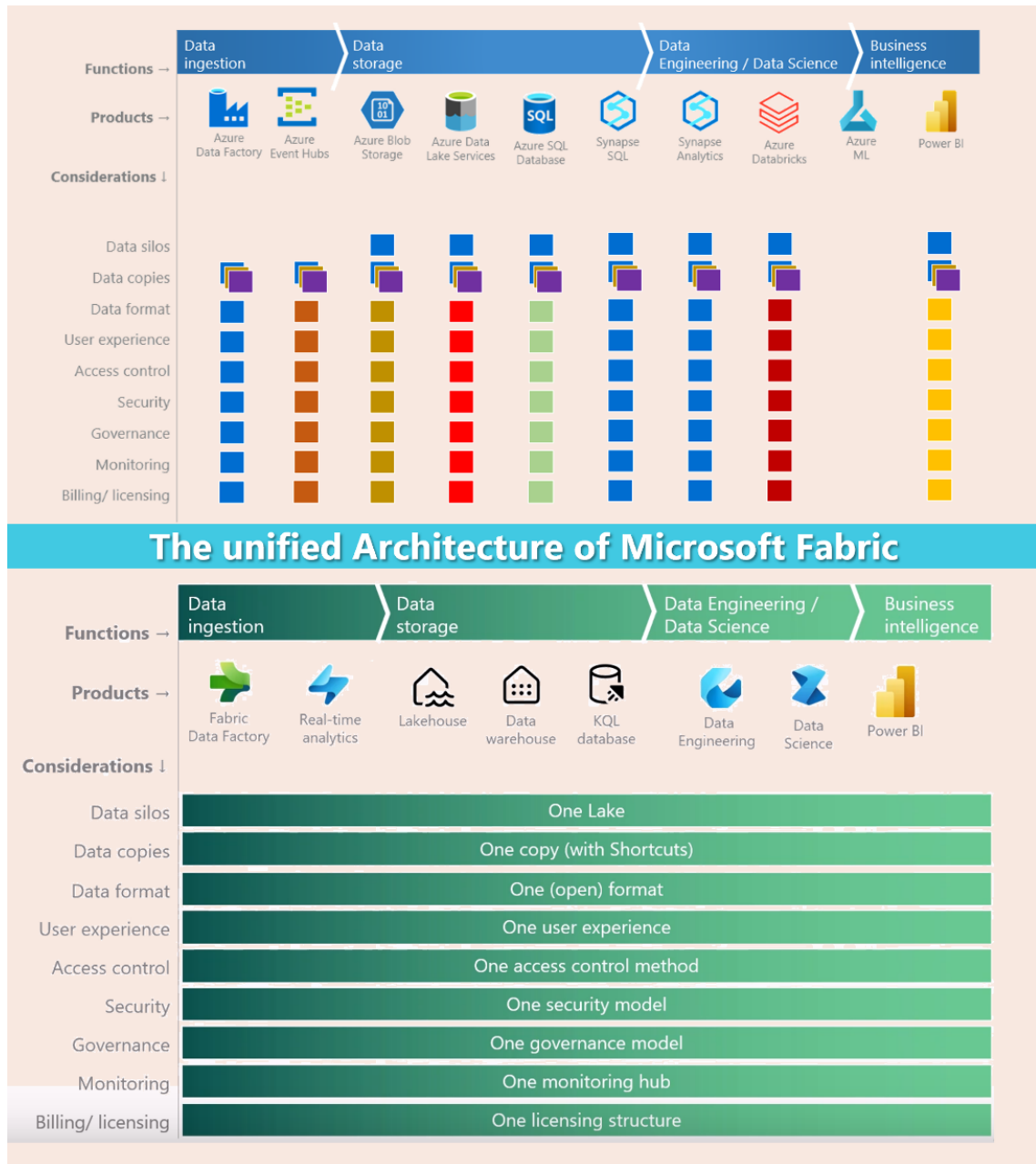


Figure 3.1: Simplified architecture of Microsoft Azure and Fabric services (adapted from adapted from [34]).

The first elements of PBI are the **data connectors**, **custom connectors** and the **connection types** themselves. At the time of writing, PBI offers 160 data connection types, ranging from simple Excel Spreadsheets to an Azure SQL Data Warehouse. It is important to note that connection type is different from the data source type. Every connection has its strengths and weaknesses, some perform better on smaller datasets, others on larger ones, and the amount of flexibility is also different between them [29]. Depending on the data source type, PBI can establish three types of connections with the data:

- Import: loads the entire dataset into memory of the host machine, whether it is

the device running Power BI Desktop locally or a server hosting the dataset after publication. Every data source type supports this connection type.

- **Direct Query:** establishes a live connection to the data source, eliminates the need for storing a second copy of the data in memory; visualizations in reports show data retrieved directly from queries sent to the data source (through T-SQL scripts), each time a user interacts with the report.
- **Live Connection:** does not store data in memory. Queries directly and gets the current data from Azure Analysis Services, SQL Server Analysis Services Tabular and datasets from PBI Service. This connection type uses the SQL Server Analysis Services technology. It is the most suitable for scenarios where large-scale data models and real-time analytics are a requirement.

Direct Query and Import connection types can be combined in a single report, into a composite model. Their behaviour combines two connection types, and it is beyond the scope of this thesis.

Besides the built in PBI connectors, a lot of external vendors offer **custom plugins**. In the case study, the following two plugins are presented: Power BI Connector for Jira and OdataLink for Xero. This plays a crucial role in allowing SMEs to integrate the data from their systems with PBI. Thus, in the next chapter, one of the requirements for determining if a BI solution can be implemented is checking if there are custom plugins for the given software. These external vendors usually offer a separate licensing plan.

Power Query is an ETL tool that is part of Power BI Desktop. After connecting to the data sources, Power Query Editor can perform data transformations before loading the data into Power BI, using the M language for performing the transformations. Power Query has three layers: the M Engine for executing queries, the M Query for writing the script in the M language, and the Power Query UI which is a visual interface for creating, modifying, and managing M queries. When connecting to a data source, Power Query has four separate steps: settings, authentication, data preview and query destination [35].

In Power BI, data sources can be assigned **Privacy Levels**—Private, Organizational, or Public—to control the flow and visibility of data. These levels indicate the data interactions, where **Private** data remains isolated, **Organizational** data can interact with other Organizational sources and finally, **Public** data is accessible to all levels. This setting is important in ensuring data security during the query operations performed in the tool.

Note that privacy levels have performance trade-offs, generally the stricter the privacy level is the slower the performance.

As a best practice, the **ETL process in Power Query** can be segmented into three separate queries: the 'Raw Data' query for extraction, the 'Staging' query for transformation, and the 'Data Model' query for final preparations prior to the load stage [35]. It is important to mention that most functionalities discussed here are also available in Excel's Power Query. This means that Excel queries can be copied to Power BI and vice versa (only for data source connectors available in Excel). When connecting to Excel data, it is preferable to do it through tables rather than named ranges or entire worksheets as it makes setup, maintenance, and transparency much easier [35].

Power Query can also un-pivot pivoted data, which addresses one of the major issues of working with Excel files. Pivoting data is also important in certain scenarios (e.g. Sales and Target are values of a single column). A variety of filtering and sorting options are also available in Power Query. Later, with the introduction of Query Folding, these become very important for performance optimization. Grouping data allows users to only load the data in the granularity specifically needed for the report (e.g. Sales by Year), making the whole process more optimal than doing this further downstream with DAX as it saves precious memory and load speeds on the visuals. Queries can also be appended in the tool, which can group transactions for example from different months, stored in separate sheets/tables.

Merging data is possible using join types between the sources, with the following options: Left Outer Join, Right Outer Join, Full Outer Join, Inner Join, Left Anti Join, Right Anti Join and Full Anti Join and Cross Join. The “Use fuzzy matching to perform the merge”, Power Query uses the Jaccard similarity algorithm to identify matches by flagging pairs of instances that have an 80% or higher similarity (this can be adjusted if needed). The use of a transformation table is also an option with fuzzy matching.

A form of **web scaping** is also possible when using Power Query, especially by using the “Add Table Using Examples” feature. Note that building business logic on external web sources poses a risk as changes by the data provider can disrupt queries without notice, often at inconvenient times.

Query folding in Power Query is an optimization process. User interface-driven steps are translated into the native query language of the data source, ideally SQL, to be processed by the server instead of locally. This enables a more optimized data handling by reducing unnecessary data loads and processing times. It's a dynamic feature that can be

affected by the order of query steps and may not persist if the sequence is altered (always check the “View Native Query” option). The availability of query folding depends on the data source and the type of connector used, with databases generally supporting this feature, while flat files do not [35]. When using Direct Query, the focus should be on maintaining query folding as long as possible to leverage the performance benefits it provides.

With the “Custom From Examples” feature in Add Columns, users can add the value of a the desired column row by row, until the engine comes up with a code that fits the specific pattern. This is an extremely useful feature for less experienced users.

It is also important to mention that there are a lot of complexities in Power Query when working with data types like currencies and dates, which are not mentioned here. This often includes **conditional logic** (as it is a logic-based expression builder), which is also an integral part of the M language.

In Power Query, **iteration** in a table on a row-by-row basis is present, often to evaluate new columns. **Recursion** in the M language is also possible, mainly through functions like List.Generate() for pagination and List.Accumulate() for aggregating values through repetitive calculations. These functions often have more efficient results compared to traditional recursive methods marked by the @ sign. However, iterative, and recursive functions are seldom used in practical scenarios.

Parameters are another useful tool in Power Query as they can later be reused in functions. For example, in the case study, the parameter of the page number is used for scraping a given number of pages off a website, assuming that only the page number changes in the URL.

Python and R scripting is available in Power Query for PBI, and their use can be extremely powerful in certain situations. Python and R scripting is discussed later in this section. In the case study, there is a recursive bill of materials (BOM) algorithm inside Power Query, which is discussed in detail.

As a final remark to Power Query, when it comes to making the decision between the use of DAX or Power Query for a specific problem, the most optimal solution in most cases is to solve as much of the requirements **upstream** as possible. Among the other benefits mentioned above, this is the most important reason to master Power Query.

DAX (Data Analysis Expressions) is the programming language used by PBI, Analysis Services, and Power Pivot for Excel. It was first introduced in 2010 with PowerPivot for Excel and it is utilized across BI communities for creating data mod-

els. DAX is part of the internal engine named Tabular that supports many Microsoft tools. DAX is designed for computing business formulas across a data model, which consists of interconnected tables that organize data into rows and columns, with relationships linking these tables [28]. In the PBI architecture, DAX is a downstream language so its use might be expensive for memory and computational times. That said, it is a very powerful tool that has many hidden complexities, making it the most complicated part of PBI for beginners and intermediaries. Unlike Excel or many programming languages where formulas can often be used from online solutions, DAX requires a fundamental understanding of its evaluation contexts and other theoretical concepts.

In PBI, DAX code can be written as a **measure** and as a **calculated column**. **Calculated columns** are values computed during data refresh and are used within the context of their specific row in a table (row context). This means that calculated columns are computed during process time and not query time, thus they use RAM which can be very expensive [28].

Measures, on the other hand, are dynamic calculations that aggregate data based on the evaluation context defined by user interactions or report filters (filter context), changing as the context changes. Measures are computed at query time, meaning that they are more flexible and efficient when working with large volumes of data. The evaluation contexts and context transitions are mandatory to understand for any DAX developer.

The **row context** identifies a specific row for computation within a table, which enables calculated columns to compute values row-by-row as each row creates its own context. This is important for DAX expressions that require the value of a column for a specific row, as it functions like a cursor iterating over rows during calculations [28].

The **filter context** applies filters to tables, affecting the result set over which DAX expressions are evaluated. This changes dynamically based on user interactions in reports, such as slicers or other visual elements. This context is crucial as it determines the data subset that is visible and computable within each visual component of a PBI report [28].

DAX has many **functions** like iterators, aggregations, logical, information, table, mathematical, trigonometric, text, date and time and relational. It also uses variables, which are useful in optimization and in writing clear and readable code. The presentation of the specific DAX functions is beyond the scope of this thesis, so the only function that is discussed separately is **CALCULATE**, while some of the others are only mentioned briefly.

The **CALCULATE** function is the most complex and powerful function in DAX. It

operates within an existing evaluation context, creating a new filter context for evaluating its arguments while discarding any existing row contexts due to a mechanism called **context transition**. It accepts three parameters: an expression to evaluate, explicit filter arguments that modify the original filter context, and CALCULATE modifiers like USERRELATIONSHIP and ALL* which adjust filter and relationship settings after the context transition. **During execution**, CALCULATE works in a very specific sequential process. It first evaluates explicit filter arguments within the original context, after which it constructs a new filter context by copying and modifying the original, and then applies any modifiers that can alter the effects of the context transition [28].

Time intelligence functions in DAX are designed to execute calculations based on time and date. It is a best practice to always include a separate date table in almost every PBI data model, as these types of functions work much more effectively with them. These functions are typically used to generate tables that can be used as filters in CALCULATE statements, as they enable users to create dynamic, time-based calculations, which is the core of every BI tool.

DAX queries are executed using a **dual-engine architecture** that comprises a Formula Engine (FE) and a Storage Engine (SE). The architecture of the DAX engines is visible on Figure 3.2. The FE is responsible for interpreting DAX expressions and creating query plans. In a way, it is the organizer of query execution, sending parts of the query to the Storage Engine to retrieve data from there. This SE operates in two modes: VertiPaq and DirectQuery.

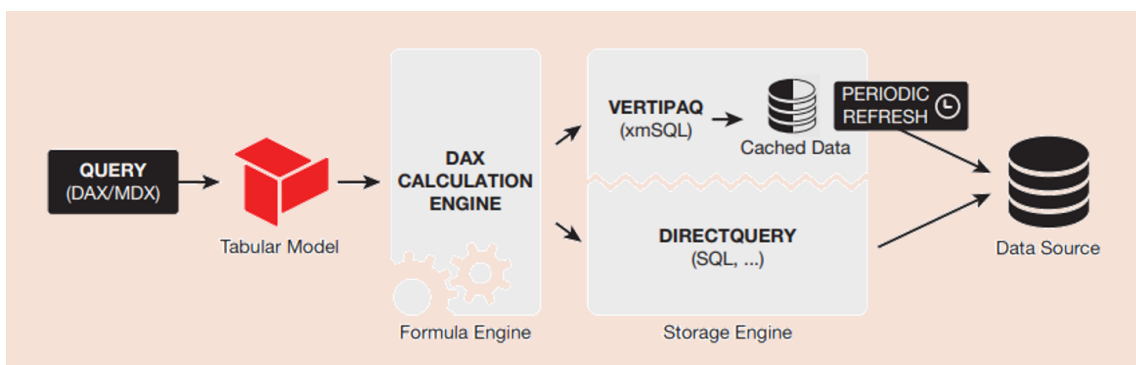


Figure 3.2: Architecture of a DAX query execution (adapted from [28]).

VertiPaq stores data in a compressed, columnar format in memory. It optimizes data retrieval and minimizes input/output operations to improve query performance. It uses techniques like value encoding, hash encoding and Run Length Encoding (RLE). The compression ratio is influenced by the following factors: the cardinality of the column,

the number of repetitions, the number of rows in the table, and the data type of the column [28].

DirectQuery mode bypasses in-memory caching, directly querying data from the source, which may slow performance due to its reliance on real-time data access. This is one of the causes of performance issues mentioned when discussing Power Query. The communication between these engines involves the FE requesting data and the SE responding with uncompressed data, which the Formula Engine then processes. In composite models, both VertiPaq and DirectQuery can be utilized within the same data model.

Optimizing DAX is a lengthy and difficult process. The same formula can perform very differently on different datasets, as many things effect the efficiency of DAX, like the cardinality of the column and the data model itself. The optimization begins by isolating a single expression that causes the performance issue and constructing a query that replicates that issue. The query goes into DAX Studio, where server timings are analysed, and the query plan is inspected. The focus should be on identifying bottlenecks within the SE or FE [28]. Based on these results, the DAX code or data model should be redesigned. After the modifications, the performance should be reevaluated. In the case study, a few calculated columns caused major performance bottlenecks and they had to be optimized. These are showcased in that section.

PBI offers many **build in and custom visuals** that can be chosen for a specific requirement. The custom visual can be accessed through the Power BI App Marketplace. The most commonly used programming language to build custom visuals is TypeScript. Microsoft has a detailed documentation on how to get started with building custom visuals and they are not discussed here as it is again, beyond the scope of this thesis. One also has the option to build their own custom visuals, which can later be reused by other users as .pbviz files or they can be deployed to the marketplace mentioned above. The other option to create visuals in PBI is by using Python and R scripts. They are also great options as these languages have a variety of visualization, especially for exploratory data analysis, options provided by their libraries such as matplotlib and seaborn. The supported Python and R visuals, along with their limitations compared to core visuals are discussed at the end of this chapter. Custom visuals can also come with advanced features. As an example, with the Deneb custom visual, users can create their own custom Vega and Vega-Lite visualizations, using declarative JSON syntax.

Python and R scripting in PBI is a very powerful feature, although it comes with many limitations. Scripting can be used in two main places in PBI: Power Query and cus-

tom visualizations. In Power Query, the two languages can be used for creating custom connectors to data sources that are not supported by PBI and for creating specific query steps. The latter is much more powerful, as there are only a few use cases where connecting to a data source with a Python script makes sense, and those cases are mostly for sample datasets.

To use a Python script **as a query step**, the data table returned always has to be in a Pandas DataFrame and the current data must be referenced with the “dataset” keyword. This option is showcased in the case study, where a recursive BOM algorithm is integrated into Power Query as a specific query step. R in Power Query requires the “mice” library to work.

Both scripting languages are also used to create visualizations in PBI. As of now, the two main Python data visualization libraries supported in PBI by are seaborn and matplotlib [36], while the R ones are ggplot2, lattice, plotly, leaflet, ggmap, ggvis, heatmaply, circlize, corrplot, diagrammeR and dygraphs [37].

The setup of the scripting languages requires the installation of Python/R on the same machine where PBI Desktop is, along with the libraries that are intended to be used and are supported by PBI. The supported version of R/Python, along with the version of their libraries can be found in the documentation of Microsoft.

The use of the scripting features comes with some heavy **limitations**. First, the data privacy level must be set to “Public”. To use scripting in PBI service, a Personal Gateway is required, and Enterprise Gateway is not yet supported, meaning that there needs to be a machine where the given scripting language and PBI Desktop are both installed. For smaller organizations, the use of a service account for this task is possible, although it is still not a perfect solution, even if it is deployed on a Windows virtual machine (VM). This is also showcased in the case study. With Python/R visuals, the data size is limited to 150,000 rows and the processing time to 5 minutes. Although visuals respond to other filters, they cannot be used as one, so cross filtering is not available [38]. These limitations only make Python and R in PBI useful in certain scenarios. This was considered when creating the framework.

Once the reports are prepared in PBI, they are ready to be published to Power BI Service or Power BI Report Server. PBI Report Server is only discussed briefly in the thesis. Before delving into reports and security in PBI service, Dataflows should be mentioned first.

Dataflows are an important part of the PBI architecture. They run independently of the

PBI datasets or solutions and use the Power Query engine in PBI Service. Thus, data transformations are performed in the cloud, separate from any specific dataset, which ensures that data is reusable across multiple reports and even across different PBI workspaces or applications. Technically, Dataflows offer two types of storage: Dataverse and Azure Data Lake, with the former being part of the Power Platform and the latter providing a data service hosted in the cloud. Dataflows support incremental refreshes, AI functions (only in Analytical Dataflows), and can connect to over 80 data source types [29]. From a practical standpoint, Dataflows can be used in PBI, Excel, and other services. They can be created within PBI and Power Apps, and the data transformed by them can be accessed from different Microsoft services. They can be put in two main categories: Standard and Analytical. Analytical Dataflows offer more advanced functionalities such as support for AI functions and storage in Azure Data Lake Storage. Standard dataflows are part of the PBI Pro license, which is the most appropriate choice for SMEs.

Shared datasets are a bit different compared to Dataflows. A shared dataset is a centralized data model that is used across multiple reports. The difference is that shared datasets have an underlying data model, which is not the case with Dataflows. Refreshing a shared dataset updates all associated reports. These reports use live connection and are often referred to as *thin reports* [29].

PBI Datamarts are one of the most important new features of PBI for SMEs. Datamarts are combine dataflows for ETL processes, an Azure SQL database for data warehousing, and PBI datasets for analytical modelling. They are all managed by a unified Web UI in PBI Service and as of now, they support over 150 data sources. Unlike traditional datamarts, which usually require separate tools for each layer (see Figure 3. In the literature review for details), PBI Datamarts have a unified development environment. They merge the data transformation to visualization pipeline in a single platform, using T-SQL Queries under the hood. It also supports **incremental refresh**, to set up a certain data load and does not load the entire data during every refresh. Licensing for Power BI Datamarts is accessible for SMEs, as they are included in the Premium Per User license (only one needed for a service account in the case of SMEs), without the need for additional Azure subscriptions or SQL database licenses [29].

In terms of scalability, there are certain limitations of datamarts in data processing speed that make this solution unfeasible in an enterprise level in many cases. They only support a data size up to 100GB and there are several computational restrictions compared to other Azure products such as the lack of performance tuning and the capacity SKUs

ranges [39]. In addition, they also lack the integration with other Azure services. However, they are a good solution in an enterprise level for building datamarts that are separate from the main DW. For most SMEs, this solution can work as a standalone DW in many cases, making it a crucial part of the framework for PBI that is showcased later.

The **best practice in PBI Service architecture** as proposed by [29] is to use a multi-layered architecture. In a multi-layer architecture, dataflows are the base for data sources. They abstract the data extraction and transformation process, while chained datasets build on this by applying additional transformations. This makes data models reusable. Layering datasets is the base for a structured approach to data modelling, where each layer represents a level of transformation. These can be updated, maintained or replaced independently without affecting the other layers in the PBI Service architecture.

In a broader sense, **PBI Service** is a web-based hosting environment for PBI Reports, datasets, dashboards, dataflows, deployment pipelines and datamarts, which is cloud-based [29]. **Workspaces** are basically collaborative environments. There, users can share and organize PBI content mentioned above. User authentication and management is managed through Azure Active Directory and the Microsoft 365 portal. This security can be configured at an object or at a workspace level. Workspaces can be created, configured and accessed with Pro/Premium Per Capacity/Premium Per User licenses. For this reason, it is important for SMEs to clearly outline the number of projected users to determine and forecast the licensing costs.

PBI Report Server is an **on-premises** server for hosting paginated reports. Its most remarkable feature is that it provides a secure environment for organizations that want to keep both their data and reporting internally managed. It offers a different version of PBI Desktop for report creation, which is similar to the regular Power BI Desktop experience, but some things are changed to be compatible with the Report Server's infrastructure. PBI Report Server is not detailed further here as it is not a suitable solution for SMEs in most cases.

PBI offers a range of **licensing options**, including the free version for individual use, Power BI Pro and Power BI Premium Per User for collaborative analytics, and Power BI Premium for larger-scale enterprise use with dedicated capacity and advanced features. A table summarizing licensing options is visible on Table 3.1 . Power BI Premium allows sharing of content with free users within an organization, with a tiered pricing model starting at \$5,000 per month for P1 nodes and offers additional benefits such as larger dataset storage and more frequent refreshes (it now includes Microsoft Fabric as

part of the package). Power BI Premium licenses are not discussed further as this thesis focuses on SMEs. User-based licensing is for shared capacity services while capacity-based licensing is specifically designed for dedicated resources. For on-premises reporting, Power BI Report Server is available through SQL Server Enterprise Edition with Software Assurance [29]. In the case of SMEs, the two licenses that we are considering are Power BI Pro at \$9.99 per user per month and Power BI Premium Per user at \$19.99 per user per month.

	Free	Pro	Premium Per User	Embedded (A SKU)	Premium Per Capacity (EM and P SKU)	SQL Server Enterprise with Software Assurance
Power BI Desktop	Yes	Yes	Yes	Yes	Yes	Yes
Power BI Mobile app	Yes	Yes	Yes	Yes	Yes	Yes
Dataflow		Yes	Yes	Yes	Yes	
Incremental Refresh			Yes	Yes	Yes	
Deployment Pipelines			Yes	Yes	Yes	
XMLA endpoint read/write			Yes	Yes	Yes	
Datamart			Yes	Yes	Yes	
Advanced AI (text analytics, image detection...)			Yes	Yes	Yes	
Secure Embed			Yes	Yes	Yes	
Autoscale					Yes	
Consume content without paid user license				Yes	Yes	
Embed in custom application				Yes	Yes	
Paginated Reports		Yes	Yes	Yes	Yes	Yes
Report Server						Yes
Maximum model size	1 GB	1 GB	100 GB	400 GB	400 GB	As per server spec
Refresh rate	8/day	8/day	48/day	48/day	48/day	unlimited
Maximum storage	10 GB / user	10 GB / user	100 TB	100 TB	100 TB	As per server spec

Table 3.1: Comparison of Power BI and SQL Server Enterprise features across different licensing tiers (adapted from [29]).

The last PBI feature mentioned in this section is **deployment pipelines**. It is a tool for the management of PBI content across multiple workspaces and it has different environments like development, testing, and production. It has features such as content compar-

ison, deployment, and connection configuration through environment-specific rules. All of these can be automated using the Power BI REST API [29]. Deployment pipelines are a premium feature, requiring either a workspace in a Premium capacity or Premium Per User (PPU) license. Thus, the option for SMEs in this case is PPU for leveraging this feature.

The last topic regarding PBI is **security**. PBI supports three types of data-level security: row-level, column-level, and object-level security. These configuration options let admins create tailored data access for different user groups. **Row-level security** (RLS) restricts data access at the row level in a dataset, so users can only view the data relevant to them. This is configured through DAX expressions and roles in PBI. **Column-level security** (CLS) restricts access to specific columns in a dataset, but its setup is different from RLS as it requires external tools like Tabular Editor for setup, as it is not natively supported in PBI Desktop at the time of writing. **Object-level security** (OLS) limits access to entire tables or objects and like CLS, it also requires the use of external tools for implementation. Although **page-level security** (PLS) is not directly supported in PBI there are specific workarounds for its implementation. They work by using different reports or customised visibility settings on visuals for different user groups. The last security type, **dynamic row-level security** makes the management of user-specific data views much easier [29]). It automatically applies filters based on user credentials, which significantly reduces the maintenance overhead compared to static RLS. Dynamic RLS is the option that is considered in the case of SME as it is easy to implement, and it is included in the PBI Pro license.

At the time of writing, Microsoft **documentation** for PBI is available in 44 languages, but in certain places it is lacking, and many elements are not explained that well. As an example, for Python scripting in PBI service, the only thing mentioned is that a Personal Gateway is required and there are no clear explanations on why Enterprise Gateways are not supported for such a crucial feature. On the other hand, PBI has a very active **community** within its forums and most problems are already explained clearly in Power BI Community. Moreover, it also has the Fabric Ideas page, where users can request new features.

In **conclusion**, although PBI is designed for business users that do not have a technical background, it has a wide range of features, most of which are not apparent to novice users and take some time to master. These are the features outlined in this section. They clearly demonstrate that with deep knowledge of PBI's toolkit, many solutions can be

implemented for SMEs that do not seem possible at first sight. The features outlined in this section serve as a foundation for creating the PBI centred framework in the following sections.

3.2 The History and Feature Set of Tableau

Tableau was founded in 2003 by Chris Stolte, Pat Hanrahan, and Christian Chabot, as a CS project at Stanford University. The company also designed and developed VizQL, a tool that translates drag-and-drop actions into data queries, which was the foundation of intuitive and easy data visualization that Tableau offers. In 2019, Tableau was acquired by Salesforce, which brought some significant changes to how the company views it as a product and how it evolved since the acquisition. These changes, which affected the popularity of the product, especially in business use cases are discussed later. Despite this, Tableau still maintained its position as a Leader in Gartner's 'Magic Quadrant for Analytics and Business Intelligence Platforms', although it has started falling behind Power BI in recent years of the report.

The first concept that is important to mention for Tableau is its **components**, as it is more complex than in the case of PBI where there are two desktop and two service components. Tableau offers a range of products aimed at different needs like data preparation and data visualization. These options vary primarily in connectivity, distribution, automation, and security. Tableau Desktop is available in Personal and Professional versions, and it provides a development environment where users can create visualizations; the Professional version supports a wider array of data sources and distribution channels including Tableau Server. Tableau Server and Tableau Online are designed for centralized data management and distribution, with features like scheduled data refreshes and the capability to edit workbooks in a cloud environment. Tableau Reader and Tableau Public is for users who need to view or interact with Tableau workbooks without full development capabilities, with Tableau Public offering additional abilities to develop workbooks that must be stored in Tableau's public cloud [30]. Tableau Prep Builder is a tool for connecting to data, querying it, and creating relationships between tables. It is basically an ETL tool that is similar to Power Query, with the only difference being that it required a separate tool.

In Tableau, there are also a high number of **data sources** users can connect to. These range from Excel and text files to databases and web data with customized connectors or

ODBC. In Tableau Desktop Personal, the connectivity is limited to basic file types like Excel, whereas in Tableau Desktop Professional, the options are expanded with 63 additional native data source connections [30]. Similarly to PBI, a lot of external vendors offer custom connectors to a variety of software. As an example, the OdataLink for Xero custom connector used for PBI can also be used in Tableau. Users can perform **data manipulations** such as joins or unions directly in the data preparation interface of Tableau. For more advanced ETL, the use of Tableau Prep Builder is recommended. Data operations in Tableau include the ability to rename fields, create calculated fields, group data, and apply filters. Tableau also supports live connections which work in a similar way as Direct Query does in PBI.

Tableau Prep in most cases is a paid ETL tool. In terms of data connectivity, it supports ODBC (Open Database Connectivity) and JDBC (Java Database Connectivity). Based on the approach of [40], data preparation can be broken down into three main steps: Input Step (E), Clean Step (T) and Output Step (L). This structure is similar to Power Query, although the UI differs, and the Clean Step is more limited compared to Power Query. In Tableau Prep Builder, modifying the structure of data sets involves four main steps: Pivot, Aggregate, Join, and Union. The Pivot step rearranges data from columns to rows or vice versa, Aggregate summarizes data, Join merges columns from different data sources based on common keys, and Union stacks rows from similarly structured tables [40]. The following join types are supported: Inner Join, Left Join, LeftOnly Join, Right Join, RightOnly Join, Full Join, and NotInner Join. Join conditions can also be added when configuring a given join.

Calculated Fields in Tableau Prep are similar to measures and calculated columns in DAX. They allow users to create or modify columns in a dataset by using functions like round(), ceiling(), or abs() directly to data field. The types of functions are the following: numerical calculations, string calculations, date calculations, conditional calculations with a Boolean output, logical calculations, type conversions and ranking calculations. **Table calculations** are also available in Prep Builder, and they are similar to calculated tables in PBI. There are a lot of functions that Calculated Fields offer, and they are not detailed in this thesis. One interesting function type that is not available in M or DAX but it is present in Calculated Fields is the regular expression, with functions like REGEXP_EXTRACT() and REGEXP_EXTRACT_NTH(). Despite these features, it is worth mentioning that Calculated Fields offer a more limited range of options when compared to DAX. Note that **Calculated Fields are used both in Tableau Prep Builder and**

in Tableau Desktop, although they work slightly differently in each tool.

When data preparation and cleaning is done, the data needs to be exported as Tableau Prep is just an ETL tool and is not a part of other Tableau tools. When **outputting data**, we can either save it to a file in formats like CSV, TDE, or Hyper, or publish it as a data source on Tableau Server. Each output option has configurations for file name, location, and type, or for server details, project placement, and descriptive metadata when publishing to Tableau Server [40]. When **writing data back to a database**, there are a couple of options. We can create a new table, add data to an existing table, or replace data within a table. To set up the output, the connection details must be configured, such as server address and authentication method, then the target database should be defined with the table name [40]. This is a powerful feature of Tableau Prep Builder that is not available in PBI. The lineage view of the ETL processes performed is also a lot cleaner and more intuitive compared to Power Query, even if the feature set is more limited.

Prep Builder also has the capability to include **scripts as query steps**. The Script step can integrate external **Python or R scripts** to help with data preparation capabilities beyond built-in functionalities mentioned above. These scripts can be used to establish API connections or handle complex data transformations such as the recursive BOM algorithm that is mentioned in the case study. This requires setting up a server connection through Tableau Python Server [41] or Rserve [40]. These connections are set up with Tableau Server (Tableau Cloud is not yet supported), meaning that scripts can run in the service this way, which is an advantage that Tableau has over PBI, which only allows the use of scripts in the service with a Personal Gateway. As scripting works in a similar way in Tableau Desktop as it does in Tableau Prep Builder, it is not mentioned again when discussing Tableau Desktop.

Refreshing data in Tableau Prep can be set up to run automatically using schedules in Prep Conductor. Similarly to PBI, there are two refresh options. A full refresh replaces the entire dataset with new data from the source. It is ideal for making sure that data is consistent, but it requires careful attention as data loss can occur if historical data is deleted from the source. Incremental refreshes, on the other hand, append only new or updated records to the existing dataset, which is a good feature for maintaining the integrity of historical data.

It is important to consider the **instances where Tableau Prep Builder needed**. In simpler use cases where only basic joins or unions are required, Tableau Desktop might be enough without needing to switch to Tableau Prep Builder. However, for more complex

data manipulation tasks involving multiple joins, advanced aggregations, or repeated pivots or configuring advanced update options, Prep Builder is required and recommended [40]. As we explored the capabilities of Tableau Prep Builder, the ETL capabilities of Tableau Desktop are not discussed further in this thesis. Calculated Fields and Calculated Tables are also not discussed further, as they work similarly to the ones mentioned in Tableau Prep Builder. The remaining part that is discussed regarding Tableau Desktop are its data visualization capabilities.

Tableau clearly beats PBI in the **data visualization** aspect. It has a lot more chart option, including built in and custom ones and they are also a lot more customizable. Tableau's visualization options include highlight tables, heat maps, dual-axis combination charts, scatter plots, tree maps, sparklines, small multiples, bullet graphs, stacked area charts, histograms, box-and-whisker plots, symbol maps with Mapbox integration, filled maps, dual-axis maps, sequential path maps, custom polygon maps, Gantt charts, waterfall charts, dual-axis slope graphs, donut charts, funnel charts, pace charts, Pareto charts, control charts, dynamic dual-axis bump charts, dumbbell charts, and customizable jitter plots. The use of the charts mentioned above, along with best practices for their use, is discussed further in [30], along with best practices for their use.

To leverage the **more advanced visualization features** of Tableau, detailed explanations of the following visualization types and techniques are provided in [31]: Highlight Tables, Dual-Axis Combination Charts, Scatter Plots, Slope Graphs, Maps, Geographic Maps, Trellis/Tile/Small Multiple Maps, Timelines, Diverging Bar Charts, Marginal Histograms, Unit Histograms/Wilkinson Dot Plots, Dual-Axis Waterfall Charts, Performance Indicator Titles, Minimalist Dot Plots, Gauges, Ranged Dot Plots, Leapfrog Charts, Bump Charts, Donut Charts, Funnel Charts, Pace Charts, Pareto Charts, Control Charts, Dual-Axis Bump Charts, Sparklines, Small Multiples, Bullet Graphs, Stacked Area Charts, Histograms, Box-and-Whisker Plots, Symbol Maps, Filled Maps, Dual-Axis Maps, and Custom Polygon Maps. Data visualization in Tableau is more complicated and advanced compared to PBI and it takes a much longer time to master.

Tableau Server/Online is Tableau's platform for sharing, distributing, and collaborating on Tableau Desktop-created content in an online environment. Tableau Server is self-hosted and it requires manual maintenance. On the other hand, Tableau Online is entirely cloud-hosted by Tableau and it uses the Amazon Web Services (AWS) infrastructure. The detailed architecture of Tableau Online is visible on Figure 3.3.

Tableau Online supports a wide range of data sources like SQL Server, PostgreSQL,

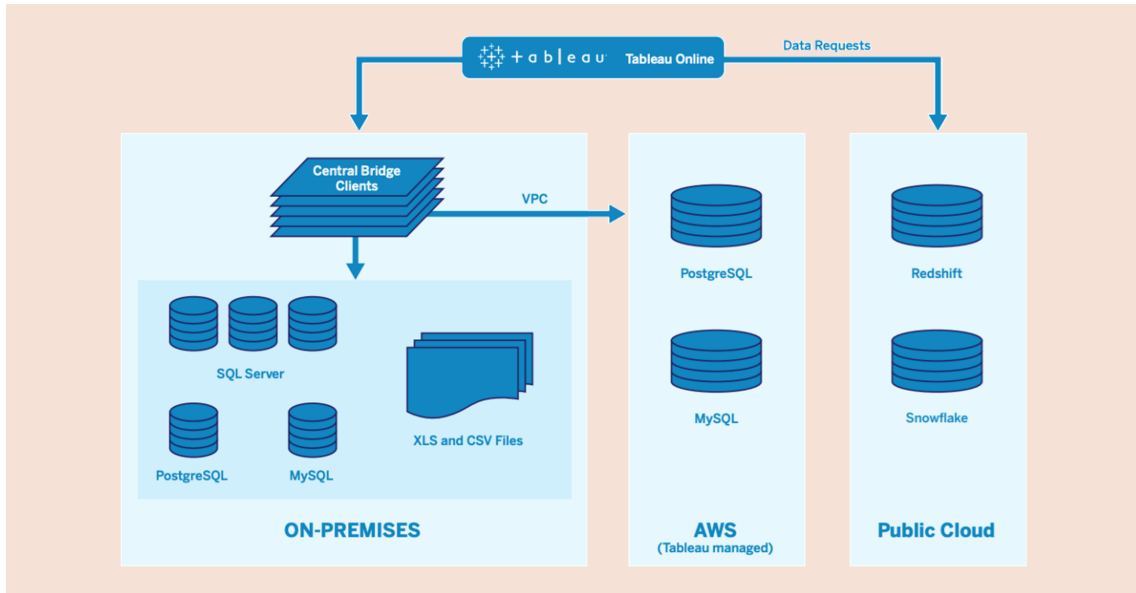


Figure 3.3: Tableau Online architecture (adapted from [42]).

MySQL, Redshift, Snowflake, Google Big Query, and various flat files. It connects directly to public cloud data sources. For private databases, whether on-premises or in a Virtual Private Cloud (VPC), Tableau uses a group of Tableau Bridge services to manage live queries and scheduled refreshes [42].

When **sharing** Tableau reports, they should usually be published to Tableau Server. There, users can interact with them using a less expensive Tableau Server viewing license instead of the full Tableau Desktop developer license [43]. This is what is used in the framework for SMEs. Tableau Server uses several file types, which is different from PBI, and it is essential for users to understand. These include Tableau Data Source (.tds) files, which define data connections and other properties, and Tableau Packaged Data Source (.tdsx) files, which contain both metadata and data extracts for sharing. Additionally, Tableau Workbook (.twb) files contain visualizations and their layouts, while Tableau Packaged Workbook (.twbx) files include visualizations along with the data needed to display them [43]. Figure 3.4. Showcases this exact Tableau Service architecture to simplify what is said above.

There are several **best practices in Tableau Service architecture**. Here, all the common approaches are listed, which are visible on Figure 3.5, based on the grouping created in [43].

TWB-centric architecture (avoid): Publishing .twb files to Tableau Server, where each workbook accesses an external data source directly. Suffers from maintenance and performance issues due to network latency and slow DB servers. Advantages include a

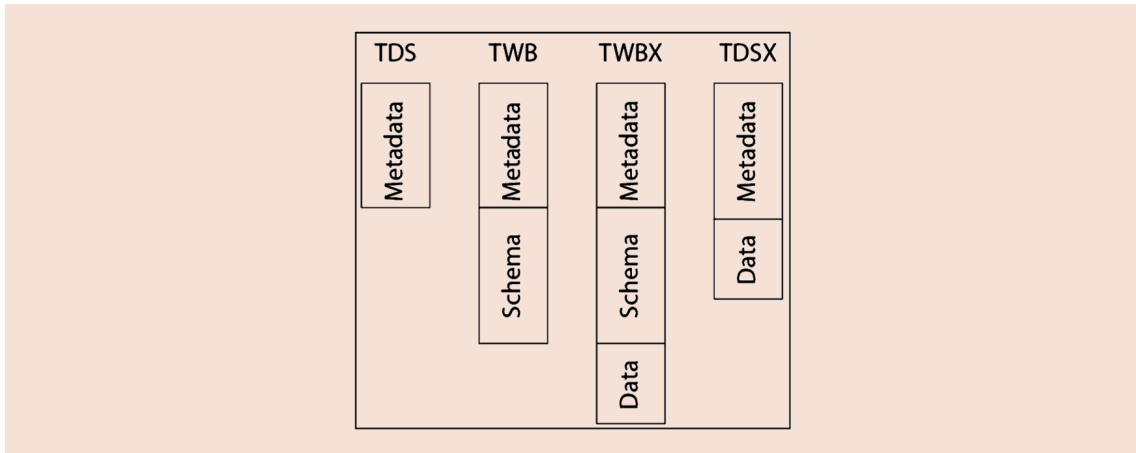


Figure 3.4: Tableau Server file types and architecture (adapted from [43]).

small footprint and easy revision history.

TWBX-centric architecture (avoid): The .twbx files that include data extracts are published to Tableau Server. No need for external data calls and faster performance. The main disadvantage is that large file sizes that can be difficult to maintain and are not updated in real-time. The main advantage is performance because of the localized data.

TDS-centric architecture (adopt): It uses .tds files as the main data source on Tableau Server, which are referenced by .twb files. Easy maintenance of data across multiple workbooks and small file sizes that lead to faster downloads and updates. The main disadvantage is poor performance from some data dependencies.

TDSX-centric architecture (adopt): Similar to TDS-centric but uses .tdsx files, which include both the metadata and data extracts. Scheduled updates that keep data up to date without real-time processing. This leads to quicker performance and easier maintenance as it handles large datasets better. It lacks real-time data updating, which can only be solved with frequent refreshes.

The next important topic regarding Tableau is **licensing**. Tableau Server offers four major licensing models, including role-based and core-based licenses. **Role-based licenses** are assigned per user and assign a range of capabilities available to each user. As an example, viewing capabilities are under Viewer licenses and full data management and creation is under Creator licenses. **Core-based licenses** are based on the server's capacity and not number of users. It specifies the maximum number of computer cores on which Tableau Server can operate. Tableau also has **Updatable Subscription Licenses** (USLs), which simplify license management by collecting every feature and every role into a single key. Lastly, Tableau's **Embedded Analytics licensing** can be used to em-

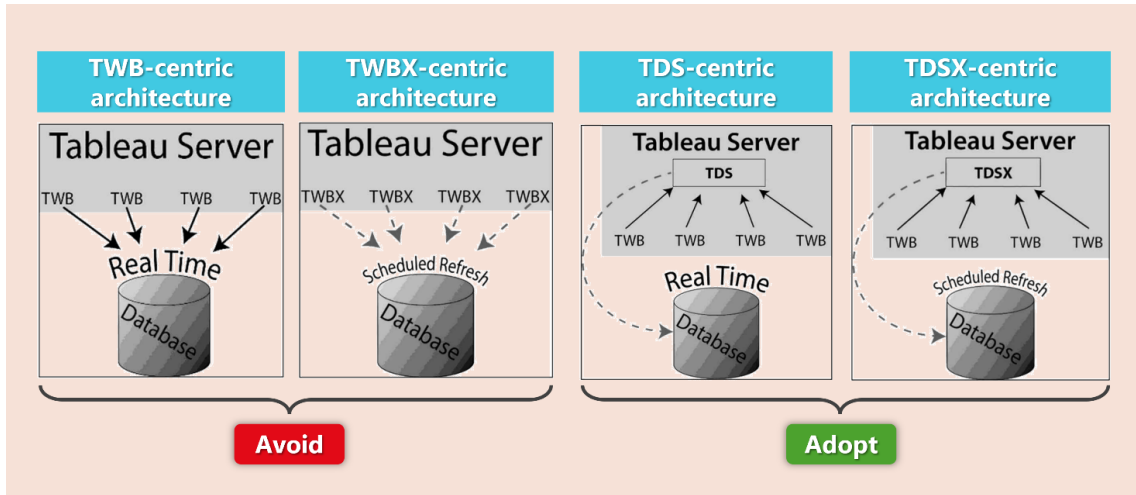


Figure 3.5: Tableau Server architecture approaches (adapted from [43]).

bed Tableau reports in external applications. Tableau Server prices are generally cheaper than Tableau Online as the company has to host its own servers [44]. Although in Tableau Public publishing report is free, the published report is available for anyone to see. Please note that Tableau's licensing structure is much more complicated than PBI's and there are a lot of things that are omitted in this thesis.

Tableau Public is also an online **community**, where users can publish their dashboards for free. This is one of the most powerful features of Tableau as the data visualization community thanks to this feature is much more active than in PBI. However, the community forums are less active for other technical questions besides data visualization, which makes it quite difficult to search for issues that come up during development. Tableau compensates this by having a much more detailed and descriptive **documentation** than PBI, so it is better to search for answers in the documentation than in community forums. As mentioned before, for PBI the situation is the exact opposite.

To **summarize**, Tableau is a tool that has a steeper learning curve, and it is not as intuitive as PBI. It has more powerful data visualization components, but navigating some of the technical aspects is much more difficult and it is also more limited. It has two main deployment options for organizations, the on-premises Tableau Server and Tableau Online in the cloud. For SMEs, the choice between PBI and Tableau depends on the requirement of the given company. These requirements are discussed under each tool's framework.

Chapter 4

Constructing a BI Centred Framework

The goal of this chapter is to provide BI-centred frameworks for BPA, DSS and mini-ERP implementations. As these BI tools are designed to have a user-friendly UI, with easy drag-and-drop features, using these tools for more complex solutions is not as straightforward, even for users who come from a technical background. Thus, having a BI-centred framework provides a practical guide to a variety of professionals from business analysts to BI practitioners, enabling SMEs to build solutions that are cost-effective, flexible, and even scalable.

In this chapter, three frameworks are presented: one for PBI, one for Tableau and a general one that can be used when assessing the capability of a BI tool for the implementation of more complex tasks. At the beginning of each subsection, the framework is introduced, and its components are explained below it, along with the limitations that should be considered at each of its components. In the case of the frameworks for Tableau and PBI, if at any point something is not described in enough detail, a more detailed description can be found in the “The Development and The Current Feature Set of BI Tools” section. As a final remark, the feature set of the BI tools described here can change in the future and that might alter the structure of the frameworks. If this thesis is read long after it was originally written, the accuracy and the components of the framework should be reassessed with further research on new feature releases from both Tableau and PBI. Finally, it is important to note that this framework is tailored to SMEs, where the amount of data that is processed is limited. For example, if there are more than 20 data sources, each with at least one billion rows, using BI tools for the tasks that are described in this thesis is not feasible, and in those cases, using BI tools is solely recommended for data visualization and report, which is their intended purpose.

4.1 The Framework for Power BI

When creating the framework for PBI, the broader architecture that is described in earlier chapters was considered and tailored to the specific needs of SMEs. The main goal is to have a framework that is cost-effective and is flexible enough to push PBI to its limits. As mentioned before in the introduction, for technical details regarding each step, consult the “The History and Feature Set of Power BI” section. The framework for PBI, that is specifically built for SMEs is visible on Figure 4.1. Besides the recommended architecture, the figure contains recommended external tools for PBI, along with licensing costs.

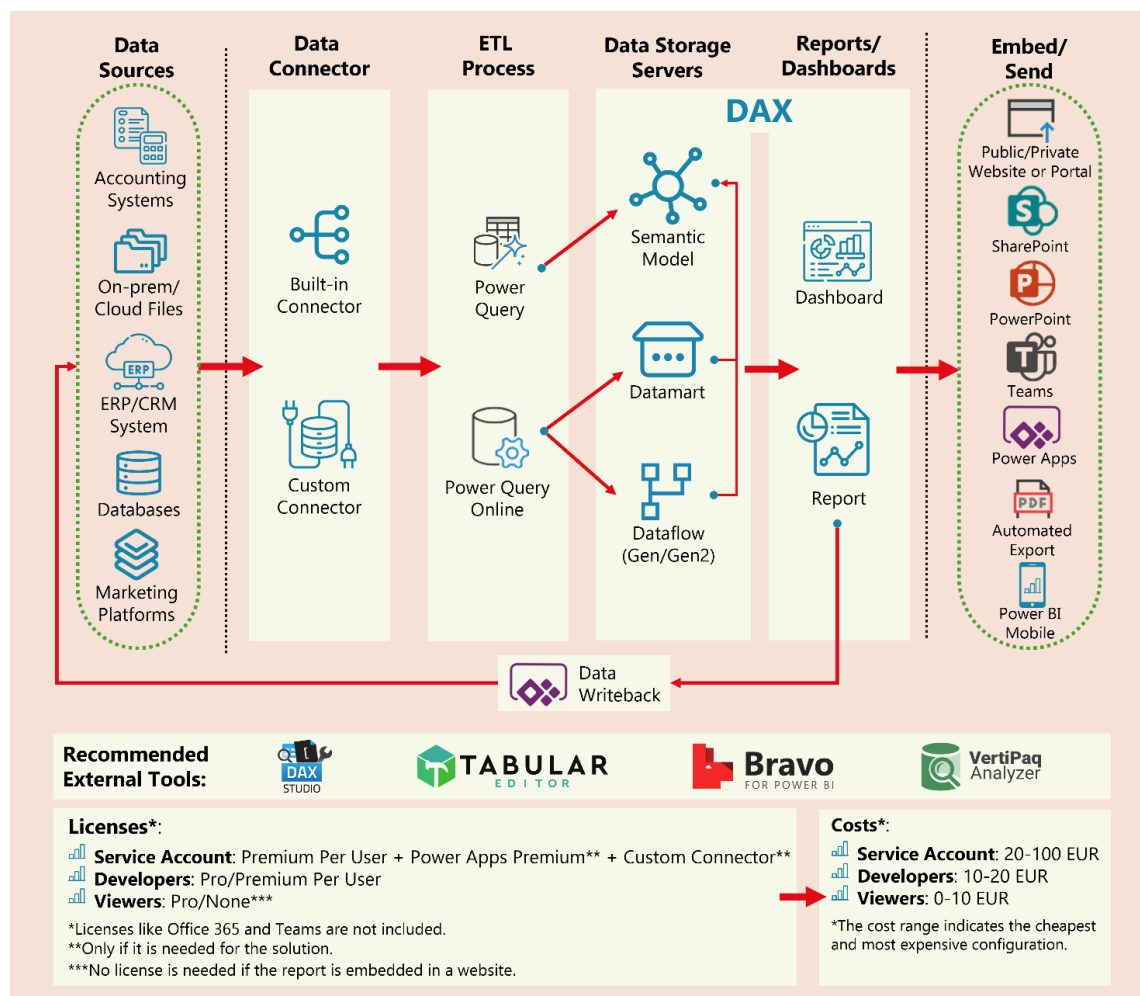


Figure 4.1: BI centered framework for Power BI tailored to SMEs.

The framework consists of six main sections, the first one of which is **Data Sources**. For SMEs, I considered five main data source types. Data extracted from accounting systems is one of the most important pieces of data for controlling purposes. An example for an accounting system that fits SMEs and is used in the case study is Xero. On-premises

files can be accessed via a Personal/Enterprise Gateway. Using Enterprise Gateways is the preferred method for connecting to on-premises files. SharePoint files for example are stored in a cloud environment and they can also be accessed by PBI. PBI offers connectors for most ERP/CRM systems, and if they are not available, there is most likely an external vendor offering a plugin option. When connecting to these systems, depending on the DB they use, connecting to them in Direct Query is preferred for real-time updates. The same guideline is applicable for databases. The last type of data source that is a staple for SMEs is a marketing platform. PBI offers built-in connectors for Google Analytics, Salesforce, and many others.

As mentioned above, the two ways to **connect to data in PBI** is through its built-in connectors and custom ones offered by plugin providers. The number of available built-in connectors in Power Query online is less than the available ones in the desktop version. Custom connectors are mostly built via REST API, and they are offered by external vendors. These only support Import mode and usually have a licensing structure, where one plugin solution is between 10 and 30 euros per month.

In the case of semantic models, Power Query in PBI desktop is used as the main **ETL tool**, and in the case of Dataflows and Datamarts, it is the online version of Power Query. It is important to mention once more that transformations that can be done in Power Query, should be done there as from a performance and storage perspective, M is more efficient than DAX. Python and R can only be used in the desktop version, and it requires a Personal Gateway, which I recommend being set up with a service account. The data size limitations should also be considered before using scripting in M.

In PBI service, we have three **data storage options**. For more information on each of these options, consult the “The History and Feature Set of Power BI” section. A Dataflow driven architecture is recommended as it scales better, it is easier to maintain, it is reusable, refresh times can be separated, and incremental refresh can be set up for the Dataflows where this is a better option. More information on this can be found in the section mentioned above.

PBI visualizations can be published as **reports or dashboards** (paginated reports are excluded as they are not a viable option for SMEs). **Dashboards** are designed to be a high-level, single-page overview with real-time data updates. They work best for quick insight sharing, with minimal interactivity, like ad hoc reporting. In contrast, **reports** are designed for detailed, multi-page analysis with interactive features like drill-through and filtering and they tend to focus on historical data.

Using **Power Apps**, it is possible to create a workaround to be able to **input data in PBI**. This solution works best with Direct Query data source options as the updates that are written back to the source are reflected almost immediately in the report. For Import mode data source options, this solution is only viable with short refresh times, otherwise inputting data becomes a very tedious task. For this reason, for smaller datasets using Import mode I recommend the use of SharePoint Online, with an embedded link to the spreadsheet in the report, as this does not require a separate Power Apps license and mapping is possible if there is a row ID in the table.

Once the reports/dashboards are published, they have a variety of **embedding options**. It is important to note that a lot of these options are from the Microsoft ecosystem, which is the main reason why PBI is the best choice for those companies, who are already in that ecosystem. These embedding options include Teams, SharePoint Online, PowerPoint, and Power Apps. In all these cases, reports retain their full functionality if they are embedded, although the viewers are required to have at least a Pro license, which makes this a weaker option for SMEs. The same thing is true for the “Website or portal” embedding option. The “Publish to web” option enables users to embed data in a website, and viewers can access and interact with that data without needing a license for doing so. However, this comes with a few security concerns. This option should not be used with confidential data as even if this report is embedded in a private website, as it is possible for an unauthorized user to get the link for the report and view the content. There are also performance drawbacks with this option. Reports and dashboards can be exported to a PDF file, and this export can be automated for given time intervals with Power Apps. As an example, a dashboard containing the sales of the previous day can be sent the next day at 8 am to executives via email. Finally, for mobile BI, PBI has a dedicated application that is available on both App Store and Google Play. Reports can be configured to have a separate mobile view in PBI desktop, where all the interactivity is retained. This feature only requires a Pro license for users, so for members of the Pro/PPU workspaces, it does not come with any additional cost.

In the **recommended external tools** section, I listed the tools I regularly use in PBI development. I only describe them briefly here as a detailed explanation is beyond the scope of this thesis. **DAX Studio** is a tool for analysing and optimizing DAX queries, and when working with larger tables, it is a staple in a PBI developer’s toolkit as it helps to identify performance bottlenecks in DAX code, which can slow down visual load times (measures) and report load and refresh times (calculated columns). **Tabular**

Editor 3 (TE3) offers syntax highlighting, semantic checking, auto-indentation and formatting, auto-completion, definition peeking, improved search and dependency viewing, diagram views for editing table relationships, preview tables with sorting and filtering, model browsing with Pivot Grids, DAX query execution, background refresh operations monitoring, model statistics via VertiPaq Analyzer, integrated DAX debugging, logical DAX call tree analysis, evaluation context switching, row-by-row context iteration, expression watching, and many others. While TE2 is free and open source, TE3 has a separate licensing structure [45], **Bravo** is a tool for more novice users. It has three remarkable features: DAX code formatting, model analysis, and separate date table creation.

VertiPaq Analyzer gathers data from Dynamic Management Views (DMVs) in Analysis Services, which PBI uses to store data. Thus, model objects, server operations, connections, and active sessions can be analysed, which are accessible through tools like SQL Server Management Studio or Azure Data Studio. The tool is not used as much on its own since it is included in DAX Studio.

In terms of **licensing**, I recommend the following setup. One service account has access to every data source needed and configures the custom plugins needed. Depending on the requirements, it has either a PPU, or a Pro license, along with a Power Apps one (if needed), and every dataset is configured through it in the service. As there are a small number of BI developers in SMEs, only they have access to this account. PBI developers have their own PPU, or a Pro license and other stakeholders have a Pro license to be able to access the workspace. Users accessing reports published with the “Publish to web” option, do not require a license. This exact licensing structure is demonstrated in the case study. Using this structure, it is easy to calculate the exact licensing costs for a given number of BI developers, other stakeholders, public viewers and one service account.

To summarize the **PBI framework**, its limitations should be mentioned first. As the exact limitations for the components of the framework are mentioned throughout this thesis, they are not discussed further here. Before implementing a solution, its given limitations should be checked first. As an example, when using the “Publish to web” function, data privacy, data size, model structure and performance limitations are all available in the documentation done by Microsoft, although sometimes not in enough detail. As a final remark, I want to reiterate once again that **this framework is tailored to SMEs**. Using the same structure in an enterprise setting almost always leads to a disaster in terms of performance, scalability, and maintenance.

4.2 The Framework for Tableau

As it is stated in the previous section, the broader architecture for Tableau that is described in earlier chapters was considered and tailored to the specific needs of SMEs. For more details on each element, the “The History and Feature Set of Tableau” section should be consulted. The framework for Tableau, that is specifically built for SMEs is visible on Figure 4.2. Besides the recommended architecture, the figure contains recommended external tools for Tableau, along with licensing costs. The licensing cost structure is not as detailed as in the PBI framework as Tableau has a more complex licensing structure.

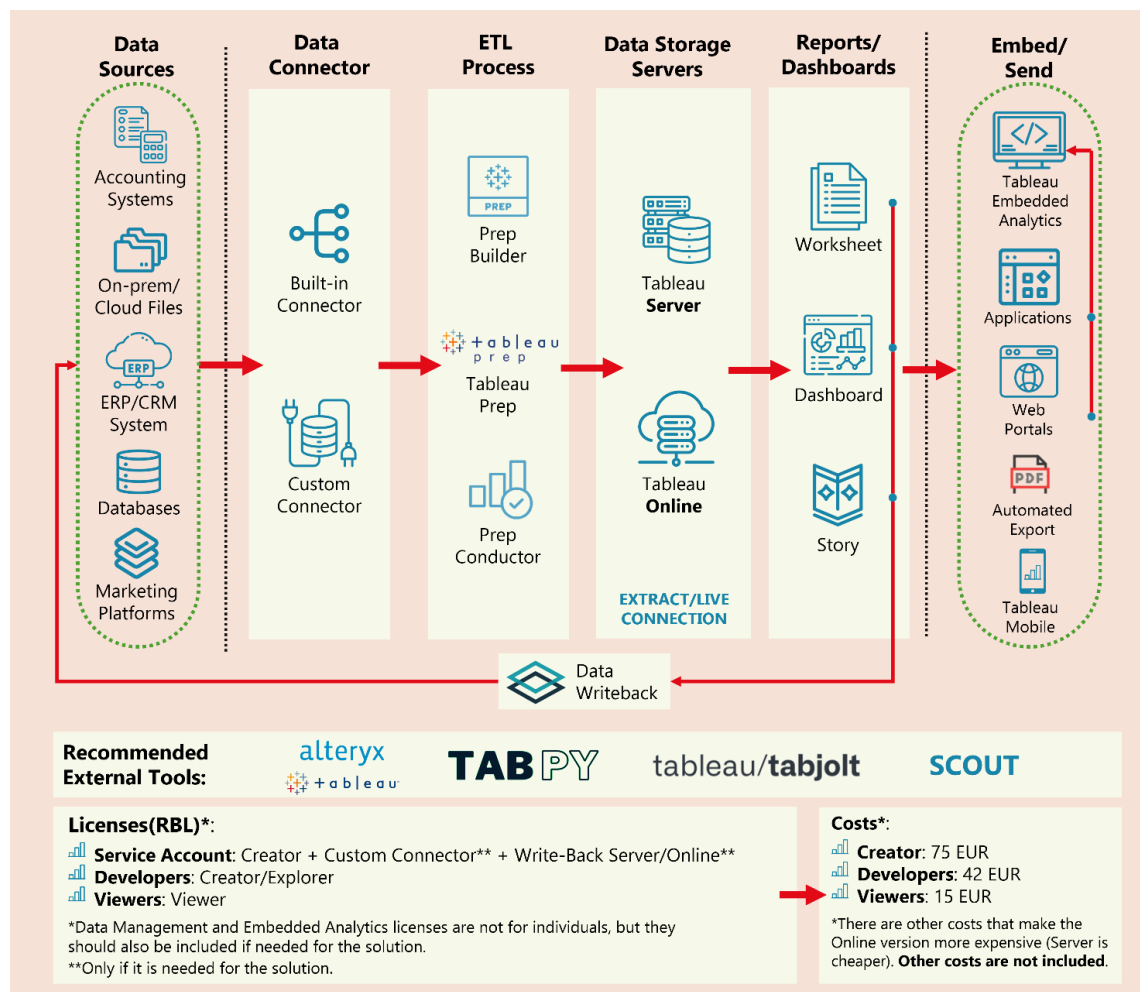


Figure 4.2: BI centered framework for Tableau tailored to SMEs.

Similarly to the PBI framework, this one also consists of six main sections, beginning with **Data Sources**. The same main data sources are considered for Tableau as both tools have a similar coverage in terms of system connectivity. As an example, the OdataLink connector for Xero can be used both for PBI and Tableau. However, some systems offer better connectivity with either PBI or Tableau, and this should always be checked before-

hand. Listing all of these differences is a lengthy process, and it is beyond the scope of this thesis.

There are also two main ways to **connect to data** in Tableau, and everything said about custom connectors for PBI either in this section or the previous ones also applies to Tableau, including the general range of licensing costs.

Where Tableau differs from PBI is the **ETL process**. Tableau has a separate application for ETL processes, which is included in the “Creator” license. A detailed description of Tableau Prep can be found in the “The History and Feature Set of Tableau” section. It consists of two tools, Prep Builder and Prep Conductor. Prep Builder is similar to Power Query, as it is a dedicated ETL tool. Prep Conductor, which was not mentioned yet is a component of the Tableau Data Management Add-on that enables scheduling and monitoring of automated flow executions in Tableau Server. It allows users to manage, view, and edit the details of their data flows, including handling errors and viewing run histories. Using TabPy, it is possible to run Python scripts as query step, with the same applying to R with Rserve. The same best practice of doing as much of the work upstream as possible, also applies to Tableau.

While in PBI we only had one **storage option** that is suitable for SMEs, in Tableau we have both Tableau Server and Tableau Online. Tableau Server is an on-premises option, while Tableau Online is a cloud-hosted version of Tableau Server. This means that SMEs that want to leverage Tableau Server must have the IT infrastructure that supports it. When this is not possible, the only option is to use Tableau Online, which has higher cost. This trade-off should be evaluated before choosing the licensing structure.

In Tableau, there are three primary types of **visualization deployments**: worksheets, dashboards, and stories. **Worksheets** are the base building blocks where individual visualizations are created; **dashboards** are comprised of multiple visualizations from worksheets into an interactive display; and **stories** are a sequence of worksheets or dashboards that narratively guide users through a story build from the other two components [46]. The naming convention itself suggests that Tableau puts more emphasis on storytelling with data than PBI.

Tableau requires the use of an external tool for **data writeback**. *Write-Back* offers an option for both Server and Online options and it works much easier than the Power Apps workaround with PBI. However, its main drawback is that the cheapest option (up to ten users) currently costs 165 USD monthly, so this is only a viable option to medium-sized companies [47]. A single Power Apps license is a much cheaper option to achieve the

same results in PBI.

Tableau also differs from PBI in the **embedding options**, as it offers a separate service for embedding. Tableau Embedded Analytics allows for the integration of interactive Tableau visualizations and dashboards into external applications and websites. It uses APIs like the Embedding API v3 and REST API, alongside features like Connected Apps and External Authorization Servers. Tableau also has automated export options and a mobile application like PBI.

I listed four **recommended external tools** that can be useful for BI development in SMEs. **Alteryx** is a more advanced ETL tool that should be used in cases where Tableau Prep Builder does not satisfy the requirements. **TabPy**, or Tableau Python Server, enhances Tableau's functionality by enabling the execution of Python scripts and stored functions through Tableau's table calculations. It can also be configured to work on Tableau Server so scheduled refresh can be enabled with Python scripts [41]. **TabJolt** is a load and performance testing tool designed for Tableau Server, used to establish performance baselines, and help in scalability under a given workload and hardware availability. It helps optimize server configurations and prepare deployments for production by simulating user activities and measuring impact on server performance [48]. Finally, **Scout** is an exploratory tool that monitors performance metrics for workbooks on both Tableau Desktop and Tableau Server, as it identifies and diagnoses slow-performing workbooks. It is also used to validate performance changes following server upgrades, configuration adjustments, or data source modifications.

The **licensing structure** in Tableau is much more complicated compared to PBI as there are a lot of additional costs and some hidden fees that need to be taken into account. A calculator is provided in [49] that approximates the licensing cost based on the number of Creator, Explorer, and Viewer licenses, as well as cloud and server-specific add-ons and other options like Embedded Analytics and Data Management. The service account should have a Creator license, while BI developers should use Creator or Explorer licenses depending on their role. The rest of the stakeholders should only use Viewer.

To conclude the **Tableau framework**, one important takeaway should be mentioned. Although some of the features of Tableau are better suited for SMEs, the licensing costs with everything factored in are much higher than PBI. This makes Tableau a less obvious choice for SMEs due to its cost-effectiveness when compared to PBI, especially if the company already uses services such as Microsoft 365. For this reason, my recommendation for SMEs is to go with PBI in most cases. As mentioned before, **this framework is**

tailored to SMEs and in an enterprise setting, the architecture would look different. As in the case of PBI, before implementing a solution in Tableau, its given limitations should always be investigated first.

4.3 The General Framework

This framework generalizes the two frameworks mentioned above. It does not apply to every BI tool, instead it aims to collect the feature set of the market leaders. Those tools have a lower barrier of entry in general and fit most business scenarios the best. It is visible on Figure 4.3.

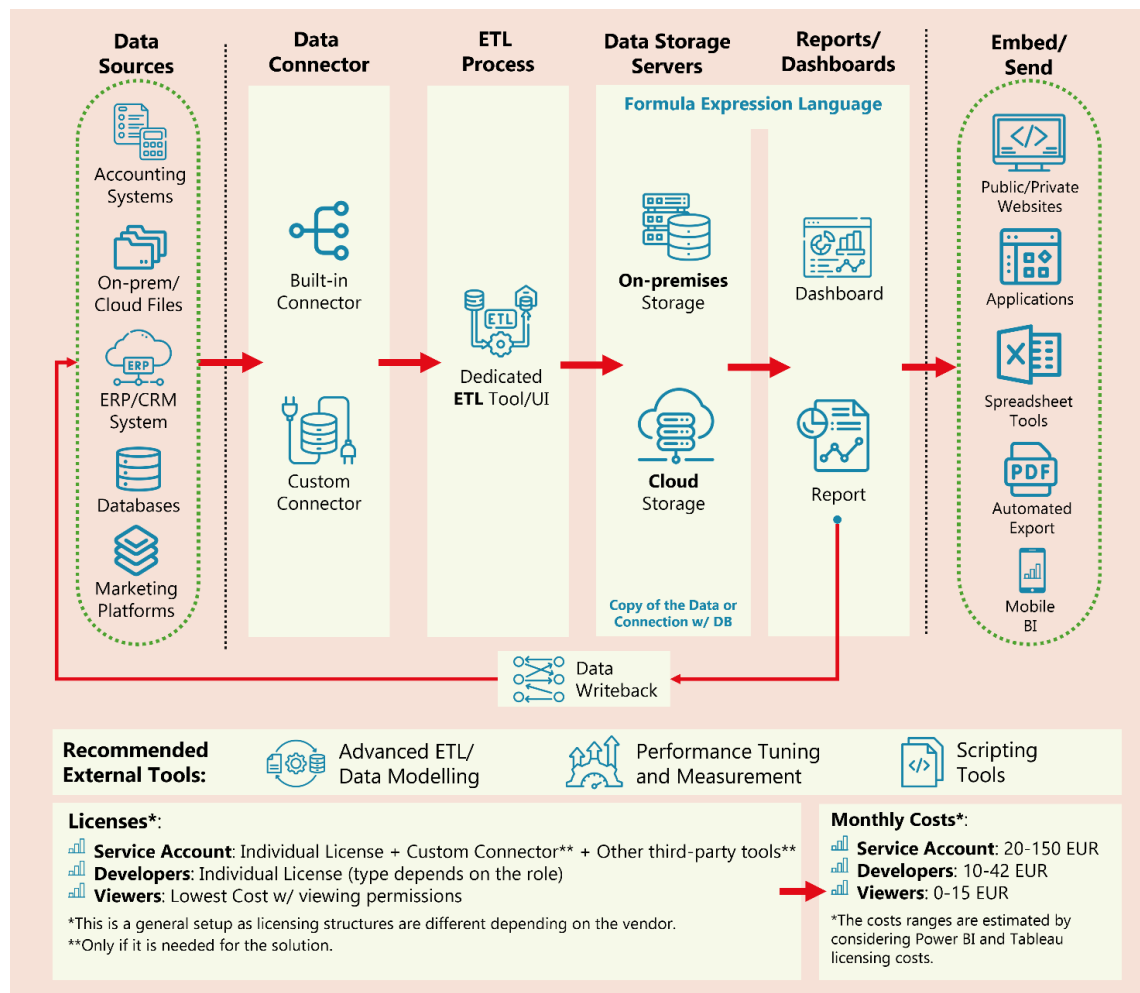


Figure 4.3: General BI centered framework tailored to SMEs.

As mentioned before, the framework is tailored to SMEs. The data source options are described in detail in the previous two sections, along with use and purpose custom and build-in connectors. Generally, BI tools offer a separate tool for ETL, which is either

built-in the main service or desktop application or it comes as a separate one. Data storage servers can be either on-premises or cloud solutions. In general, cloud solutions are easier to maintain and implement as they do not require a separate IT infrastructure. There are two main connection types. BI tools can either store a copy of the data or establish a live connection with the DB. When live connection is available, it is the preferred method, even if it comes with performance drawbacks. In general, BI tools offer their own formula expression language that is used for data manipulation after the ETL process is completed. The two main deployment types are dashboards and reports, though naming conventions may differ between BI tools. The difference between them is explained in the PBI framework. In terms of embedding options, reports and dashboards built by most BI tools can be integrated into public or private websites, applications, and mobile BI apps. They can also be exported into other spreadsheets tools like Microsoft Excel or into PDF files. Both export options can be automated and scheduled in most cases. The recommended external tools for BI development fall into three categories. Advanced data modelling and ETL tools are a must have when the dedicated tools offered by the BI vendor is not advanced enough to fit the data modelling or ETL needs of the SME. Performance tuning and measurement tools are needed for BI tools that have certain components that do not scale well, like DAX in PBI. Scripting tools can be integrated in the BI tool's ETL processes, they can create custom data visualizations or even offer an alternative to the formula expression language used by the tool. Finally, in terms of licensing, I recommend that the highest possible individual license that fits the requirements of the company should be assigned to a service account. Moreover, custom connectors and other add-ons that can be purchased as individual licenses should also be assigned to that account. For developers, individual licenses should be assigned, with the quality depending on their role. Lastly, other stakeholders should have the cheapest license that gives them permissions to view the reports and dashboards in a secure environment.

To conclude this chapter, I want to reiterate that the three frameworks showcased here are tailored to the needs of SMEs only, and in an enterprise use case the framework would be different. This is because in an enterprise environment, the goal is to use the tools solely for their intended purpose, which is data analysis and visualization. The solutions that are intended to be built with this framework extend beyond that. As such, these solutions are not scalable for enterprise data. However, they are extremely efficient in offering an alternative to SMEs if they want to use BI tools for BPA, DSS, or even mini-ERP systems. This exact capability is demonstrated with a case study in the following chapter.

Chapter 5

The Case Study of NGI Systems

NGI Systems is a technology consultancy company that specializes in integrating hardware and software. The company works on developing innovative products and solutions, particularly for early-stage initiatives, and they provide technical guidance and resource optimization. NGI Systems' goal is to solve technical challenges using the least amount of time and resources to achieve a viable solution. They offer a range of services, including requirements elicitation and refinement, architecture and design, UX/UI design, data and process modelling, agile application development, builds and deployments, infrastructure setup, automation testing, and project management.

The case study showcases two years of work, while I was working in the position of BI Developer at the company. At the start of my work, only the broad requirements for certain projects were outlined and the company had not yet adopted PBI. During my two years at the company, the solutions I built encompass over 100 tables and 100 individual report pages. All of the solutions were built from scratch, and they are used in production in a business-critical capacity.

The case study consists of seven solutions in total, which can be grouped into four categories: production management, HR compliance and management, financial reporting, and external billing reporting. The solutions link together multiple accounting systems, an issue tracker, a time logging system, and several other structured data sources for HR information and all of them use near real-time updates. The business requirements and the initial business logic are outlined before presenting every solution, along with the setup necessary to perform the given requirements. These steps are important in defining the infrastructure needed to build similar solutions. These setup requirements are put into the PBI centred framework discussed in the previous chapter. Optimization is also discussed

in a few cases. A recurring theme during this chapter is that with some exceptions, the difficulty of creating these solutions mostly relies in designing the infrastructure for them rather than implementing the given requirements.

This chapter is concluded by an interview with the CEO and co-founder of NGI Systems, Tibor Barna, who has been coordinating these projects and is also my direct supervisor. He offers valuable insights about the process of creating, building, and maintaining these solutions as well as their impact on the firm. If we think back on the literature review, where I discussed success factors in BI implementation, he clearly is our Project Champion.

5.1 The Power BI framework of NGI Systems

In this section, I demonstrate the setup needed to implement the solutions that I have developed in PBI. This includes licensing, PBI desktop setup, PBI service architecture, custom connector configuration and every other additional step needed. To display the effectiveness of the PBI framework from the previous chapter, I highlighted and expanded every component from it that we use at the company. Those components that we do not use from it are displayed in grey. **The PBI framework of NGI Systems** is visible on Figure 5.1.

The company uses three main **data sources** for PBI. Xero is the accounting system used for its United Kingdom entity and Saga is the one used for the Romanian one. Both are brought in PBI for reporting and to build a consolidated profit and loss statement (P&L). Supporting files that are used for mapping such as a Payouts Tracker for the HR Expenses report are stored on SharePoint. There, a supporting file structure is already built out with the appropriate access control. Lastly, the company uses a project management software from Atlassian. Each collaborator at the company must log their activity in Tempo, which is a part of this project management software. When logging, the collaborator must specify the Issue Key, Date, Duration, Start time, End time, Description and Type. The Billing Rate and Account fields are optional, and they are only used for edge cases. These logs are used for detailed external billing reports, and they are the foundation for the payment system for collaborators. Moreover, they are also used for HR compliance reports and in metrics reports where historical data is analysed. These two use cases were built in PBI with the appropriate supporting tables being stored on SharePoint. In Jira, issues are part of a hierarchy. Each issue has a parent issue, and each parent issue

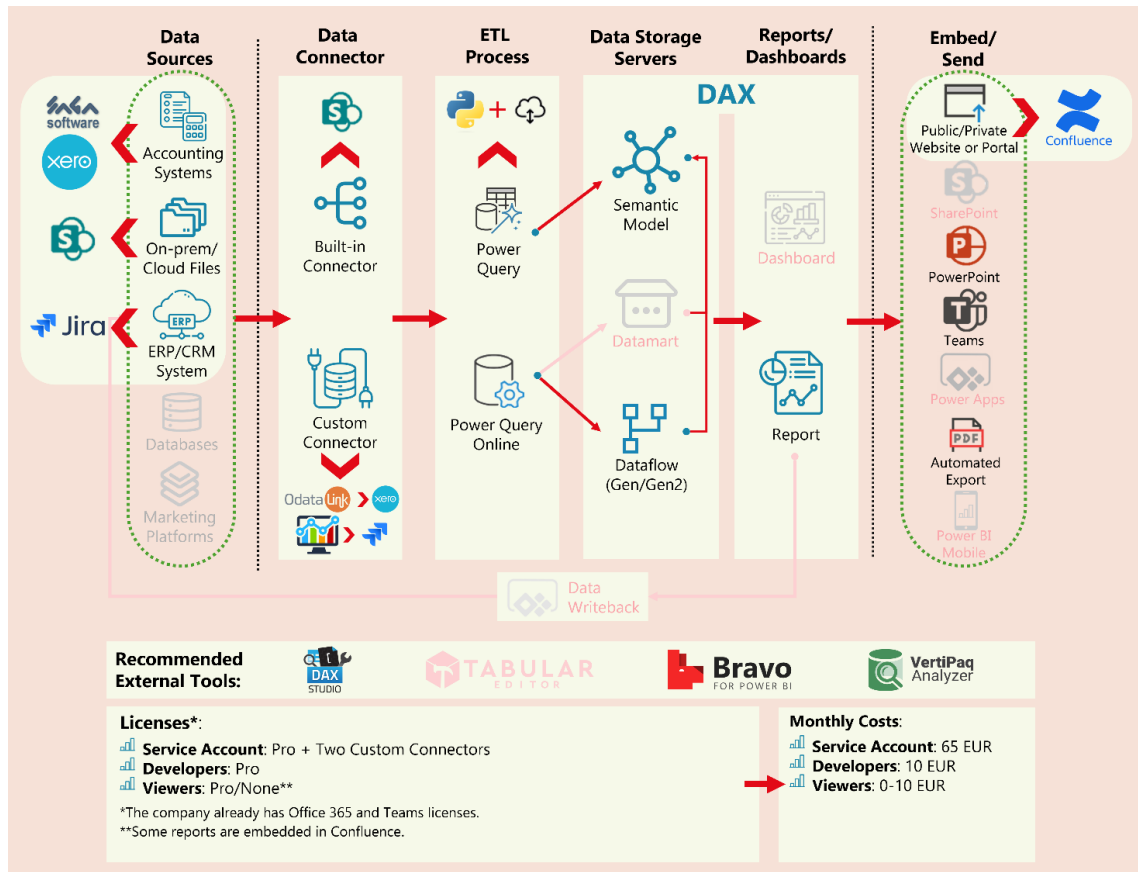


Figure 5.1: The Power BI framework of NGI Systems.

is part of a larger project. This is extremely important for both internal and external use. As an example, in the billing reports I built, our clients can see their cost breakdown in a hierarchy, with the bottom of this hierarchy being the individual worklogs. These are showcased in detail in the following sections.

NGI Systems uses a built-in and two custom **connectors**. The built-in connector brings data from SharePoint in import mode. The Power BI Connector for Jira is configured to bring in the issue hierarchy, the worklogs and other custom fields and tables that are needed for reporting and handling edge cases. It has a user-based licensing structure. OdataLink is a custom connector for Xero accounting data, with an additional layer of fields for constructing P&L, Balance Sheet, and Cash Flow reports. It has a data file-based licensing.

Both the desktop and the online version of PBI are used for **ETL**. In Power Query desktop, the supported version of Python and its libraries are installed to be able to do Python scripting as query steps. An on-premises data gateway (personal mode) is also installed with the credentials of the service account to enable reports containing scripting to be refreshed in PBI service.

Power Query online is used to configure the dataflows, which is the first type of **data storage** that the company uses. There are two main dataflows: a separate calendar table generated by Bravo and one for the data brought in by the Jira connector. Every other data source used for financial and HR data is stored in a large semantic model. The reports connect to that model via live connection.

There are 15 **reports** in the main workspace, and they contain over 100 pages in total. The internal HR compliance reports are **embedded** in separate private Confluence pages, so viewers of those reports do not require a license to interact with the report. Reports are also often embedded in PowerPoint, shared in Teams and exported as PDF files or as spreadsheets, especially the billing reports.

Bravo is the main **external tool** used for code formatting, model size analysis, and separate calendar table creation. DAX Studio is used for the performance tuning of measures and calculated columns.

In terms of **licensing**, the service account is used for configuring the two custom plugins and it has a Pro license. BI developers and viewers of the main workspace also have a Pro license. It is important to note that the company has Teams, Office 365, and SharePoint licenses, which are not included in the cost breakdown.

5.2 HR Compliance and Management Reports

The first project carried out in PBI was the **automation of the payment calculation processes**. As the company works with many freelancers and it has two legal entities, it is quite difficult to track the correct number of payments, which are not only based on hourly rate, but a variety of other factors such as a multi-currency logic and project-based contracts. These are not discussed here in detail as it would require at least ten pages to explain the business logic behind it. There are also retainers for collaborators with a fixed collaboration type (i.e. fixed monthly hourly rate) to handle the case when someone has more/less logged hours compared to the requirement and other items such as clawback, one-time bonuses, referral compensations and meal tickets. On top of the retainer and payment calculation process, for every collaborator, the monthly **effective hourly rate** can be calculated. This is later used in another report to determine the cost of each worklog. To put it in perspective, the company currently has over 260,000 worklogs, which is an extreme level of granularity for cost calculation. The business logic and the DAX code of the calculation of the effective hourly rate is the only piece of business

logic showcased from the report in this thesis. The flowchart and the DAX code of the calculated column that computes the effective hourly rate is visible on Figure 5.2. This column takes data from SharePoint, Jira and Tempo to calculate the final result.

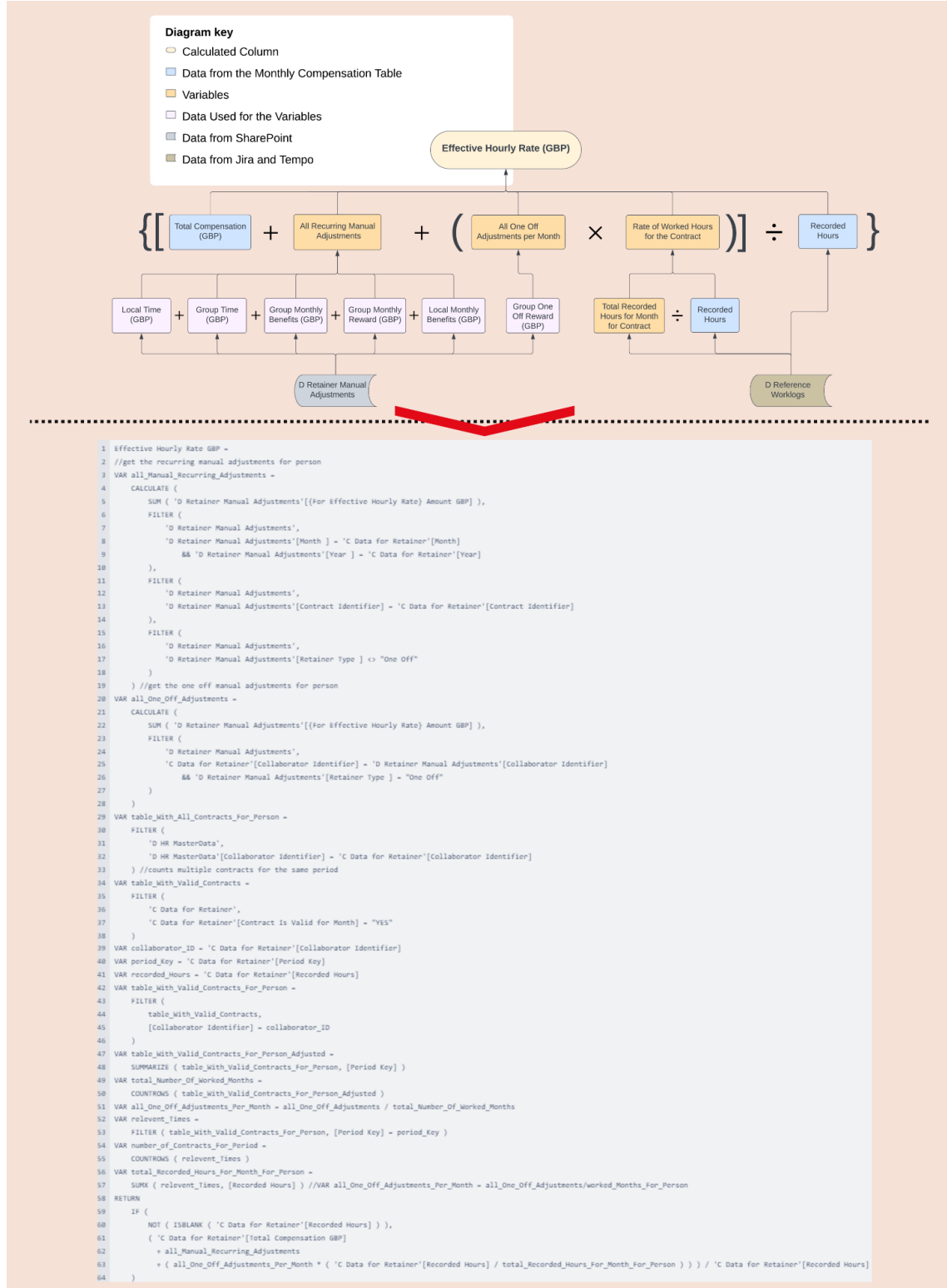


Figure 5.2: Effective Hourly Rate business logic and calculated column.

The formula visible on Figure 5.2 reflects the cost value of one hour of work for a given collaborator, for a given month and for a given contract (as one collaborator can have multiple contracts for the same period) in GBP. There are over 20 calculated columns in this project that are like this in complexity.

HR Compliance reports are used for tracking the activity of collaborators based on data from Jira and Tempo. There are three main reports that fall into this category: the Timesheet Compliance Report, the Time Concentration Report, and the Outsized Worklogs Report.

The **Timesheet Compliance Report** is used to track if collaborators log their work in a timely manner as per the company policy. This way, the performance of collaborators is available both for management and for the collaborators themselves at a glance. The **Time Concentration Report** summarizes the logged hours for the last three months on each issue as there should be a justification for going over 50 hours on a single issue according to company policy. The **Outsized Worklogs Report** collects every worklog from the past three months where an individual log is above two hours. It is sorted in ascending order. All three of these reports are embedded on a separate Confluence page and display data near real time. A portion of each report is visible on Figure 5.3.

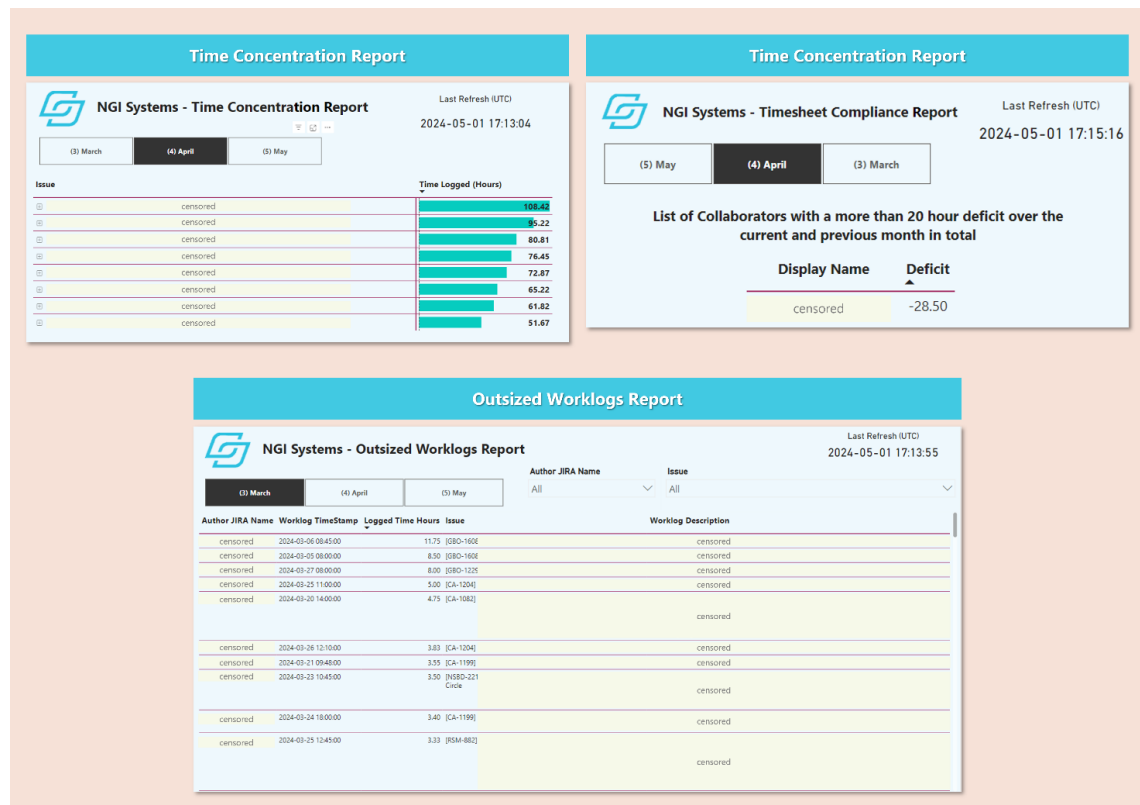


Figure 5.3: HR compliance reports on Confluence.

5.3 Production Management Reports

The most complex report in terms of BI architecture and capabilities is the ERP report that I have developed for production management. The solution encompasses the management of outbound and inbound invoices related to the manufacturing of IoT devices, a separate breakdown for research and development (R&D) costs, a production simulator, and exploded BOM (bill of materials), graph visualizations of component assemblies, a multi-currency logic, and an inventory explorer based on a first in, first out (FIFO) logic. The exploded BOM works with a Components Processing Mappings table, where based on a stock keeping unit (SKU) code, every intermediary step in manufacturing is mapped from the unboxing of a product to the assembly of the final IoT device from subcomponents. Every part of this project is implemented with Power Query and DAX, besides the recursive BOM algorithm. The complex Power Query and DAX solutions, such as the FIFO logic is not showcased in this thesis. On the other hand, the integration of the recursive BOM algorithm is discussed in more detail.

The initial setup needed to enable Python scripting in PBI service is discussed earlier in this chapter. There are three recursive Python algorithms in this ERP report: one to compute the exploded BOM and two others to make the data structure both for the directed graph and the tree graph. Only the first one is discussed here. The recursive BOM algorithm computes the amount of every bottom level input needed for a given top level output. Top level outputs fall into three categories: Cable or Accessory, Internal Component and Primary Finished Component. There were many challenges in creating an accurate output, such as separating multi-output processes with only bottom level inputs and tracing the byproducts back to the process where they can be accounted for accurately. The overview of the algorithm, along with its detailed steps is visible on Figure 5.4.

The rest is not discussed further as it is beyond the scope of this thesis. During the rest of this section, some of the report tabs are discussed. The Production Capacity Overview shows how many of each top-level output can be manufactured based on the current stock of both bottom level inputs and intermediaries. The Production Simulator displays the quantity needed to manufacture the given top-level input, the quantity in stock and the unit production capacity. Thus, inventory bottlenecks can be identified before preparing for production. Finally, the Inventory Explorer tab showcases the current amount in stock of every item, along with the adjustments that happened to them historically. When hovering over a line item, the tooltip shows the exact FIFO breakdown of the product based on the

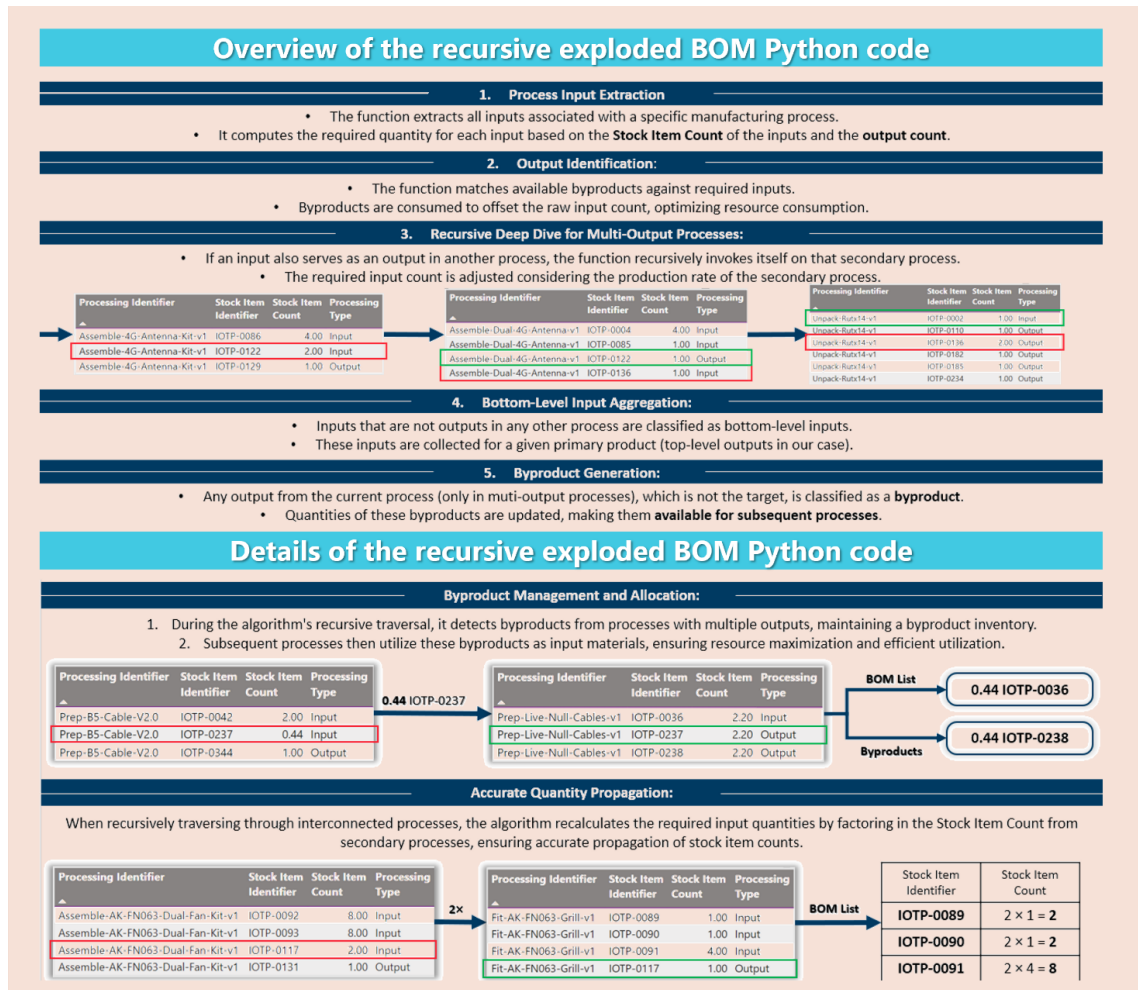


Figure 5.4: Recursive exploded BOM Python code.

orders and sales. Before concluding this section, it is worth noting that this solution really pushes the limit of what PBI is capable of and this ERP report is only feasible for SMEs. However, this ERP report is working at business-critical capacity, and it streamlined the manufacturing, billing and the preparation phases for *NGI Systems*. Some of the report tabs from the ERP report are visible on Figure 5.5.

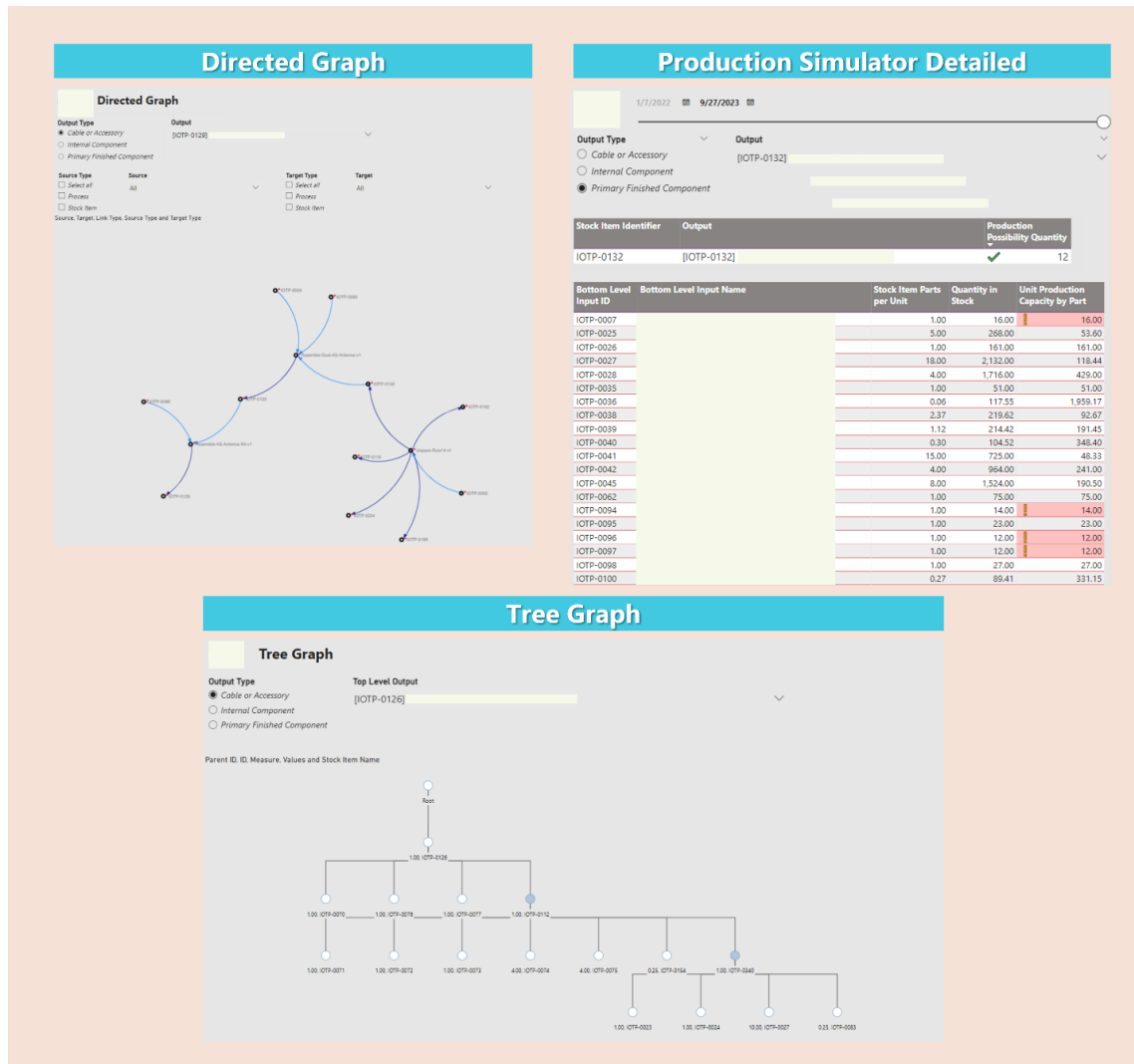


Figure 5.5: Sample pages from the ERP report.

5.4 Financial and External Billing Reports

The final category of the reports in this case study falls into two separate subcategories. The **financial reporting** category is one which is still in progress. The final goal is to create a consolidated report from the two entities, which has an issue level granularity. This means that based on the effective hourly rates, the billing report, the data from Xero and Saga, and finally with the Jira data, a full consolidated P&L will be built out in PBI, which is why most of the data flows into a Master Time Cost Dataset.

Billing reports are built for the company's clients to create a level of transparency and visibility of the billed work done. There is a separate input table in SharePoint that defines the billed hourly rate of every collaborator for the given billing report and date range. Every billing report has a similar structure, and it consists of eight report tabs.

A summary table is present on the first tab, where the worked ours and the total cost is broken down into issues and parent issues. The next tab displays this in a more detailed matrix visual with a drill-down option. On the lowest level, clients can see the amount of work done by day, by issue and by collaborator in each month. The third tab contains a burndown chart, with the rolling credit balance of the client. The next four tabs offer different visualizations of the worklogs with a decomposition tree, funnel, pie chart and stacked are chart. Each of these pages have several filers and drill-down options. The last tab contains the service credits purchased by the client. Management from the client company is given access to only this report with a separate Pro license. On request, exports from the first three tabs can be given as separate PDF files or spreadsheets as well.

These reports create a level of transparency and granularity in billing that is rarely seen in the case of SMEs. The report tabs from these reports cannot be showcased as they contain sensitive data.

5.5 Performance Tuning, Error Handling and Documentation

To address the limitations that I have come across during development, three main areas had to be addressed: optimization, error handling and documentation. All three of these are recommended to any SME building BI solutions to ensure scalability, transparency, and proper maintenance.

Performance tuning was necessary as some of the calculated columns, where a given granularity was required had poor performance due to the cardinality of the columns which were used in the formulas. The optimization processes are performed in DAX Studio based on the best practices and guidelines outlined in [28]. One example of this is the allocation of the Effective Hourly Rate mentioned before back to the worklogs to determine the cost value of each log. The table has over 260,000 rows with most of its columns required in the formula having a high cardinality. This caused the formula to run for more than 24 minutes to compute the result. As this is a calculated column, it means that the opening time of the report in PBI desktop and the refresh time in PBI service were both extended with 24 minutes. As the worklogs and the number of contracts grow, this number would grow with it. The comparison of query plan and the server timings of the old and new formula is visible on Figure 5.6. The optimization process in general is de-

tailed in the “The History and Feature Set of Power BI” section. The “EncodeCallback” in the figure refers to the even when the SE is not able to perform an operation and it sends a request to the FE, which is single threaded and has a poor performance compared to the SE. The old formula was written during the initial phase of BI development at the company, and it used functions DAX that scanned entire tables where it was not necessary and a CROSSJOIN which both negatively impacted performance. The new formula uses more variables, which are reused throughout the code and CALCULATETABLE and table filters in earlier to only retrieve the necessary data as early as possible in the evaluation phase and thus puts more workload on the SE as visible on the figure mentioned above. It is important to note that there is not a one size fits all approach to DAX code optimization and it can be a very tedious process even when following best practices, as the performance not only depends on the formula itself, but also on the underlying data structure. The new formula has a server timing of less than two seconds, which is orders of magnitude lower than the old formula. This sped up operation is PBI desktop and refresh times in PBI service, while also retaining the original business logic.

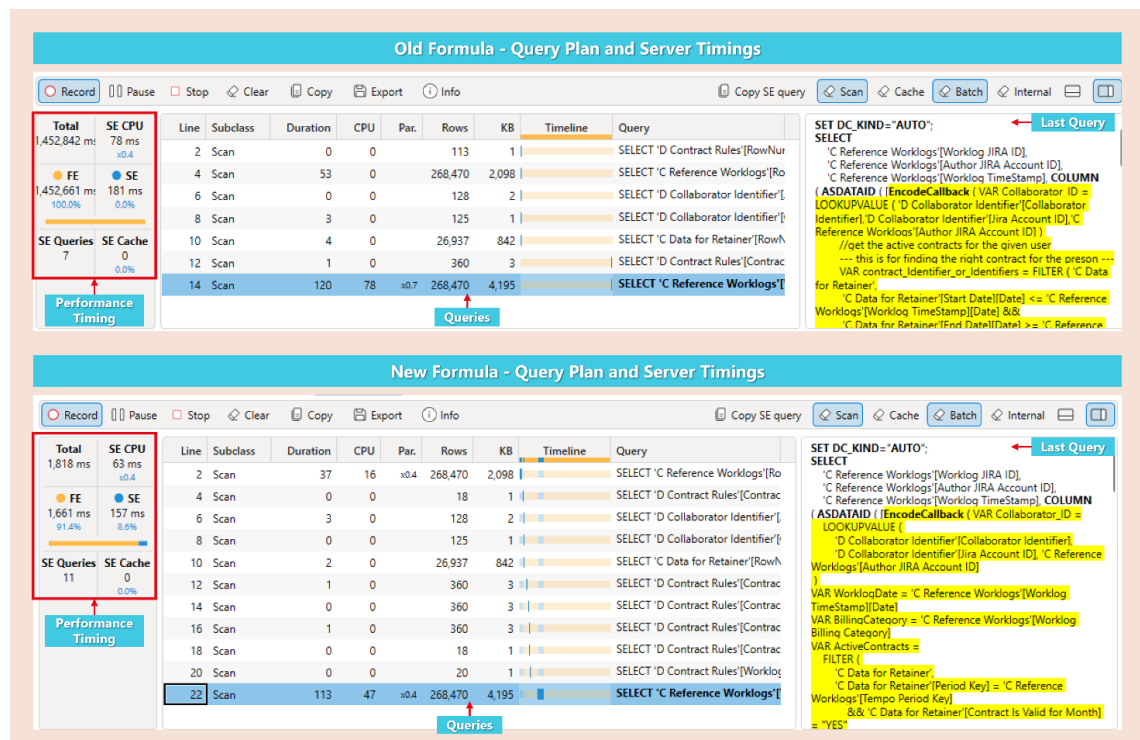


Figure 5.6: Query plans and server timing of the old and the new optimized DAX code.

Error handling measures are extremely important in every report at the company since they are used in business-critical scenarios. One example of this is showcasing the corrupted worklogs in the billing reports, where an exact billing value cannot be

determined for a given log. A lot of other error handling functions are related to those SharePoint files where manual data entry is required, such as the report containing the automation of the payment calculation processes where over a dozen of such functions are in place. There are also separate pages for worklogs that are not assigned to a given contract for a given month as they are not taken into account when calculating payments.

Lastly, the work done in every report is **documented** on Confluence pages, which are updated at regular intervals. This document is not only made for BI developers, but also for other collaborator who regularly use the reports to understand the exact business logic behind them. This is crucial when performing operations such as inputting data in SharePoint for the payment calculation processes. Flowcharts are embedded into the documentation to provide guidance on more complex formulas for collaborators that do not have a technical background. The documentation also includes the credentials for the service account, PBI resources and workarounds such as unlocking and locking historical periods in Jira using private APIs.

5.6 Interview with the CEO of NGI Systems

As mentioned before, the case study section is concluded by conducting an interview with Tibor Barna, the CEO of NGI systems. The presence of a project champion has been identified as a significant factor in BI adoption for SMEs [25]. As stated in the literature review, a project champion is a person with leadership skills, deep knowledge of business processes and understanding of technological innovations, preferably in an executive position. All these requirements are satisfied by the interviewee given his position at the company and his background.

The interview was conducted via email, where I asked six main questions, which are related to the solutions mentioned above. These questions are visible on Figure 5.7. The rest of this section is grouped into six part, where the main idea from the interview is highlighted.

The **first question** focused on measuring the qualitative effects of the billing reports which are showcased earlier in this chapter. Tibor Barna concluded that “The external billing reports definitely help to automate the reporting process and ensure that the clients had an effectively real-time view of what was happening. It brings lots of benefits, but at the same time, it does generate its own challenges overall. I feel strongly that this is a net positive, and it is something that simply required processes to be adjusted around it to

Interview Questions
1. In what ways did the external billing reports contribute to business performance and transparency with the company's clients?
2. How did the HR Compliance and Management reports streamline HR and other internal processes?
3. In what ways did the ERP report contribute to the production management of IoT devices?
4. What are the biggest limitations of the Power BI reports in general?
5. What are the future goals for the internal use of Power BI?
6. What do you think are the main challenges related to successful BI implementation in SMEs?

Figure 5.7: Interview question for the CEO of NGI Systems.

accommodate this new more automated and more fluid way of sharing reports transparently.”

The **second question** is about the HR Compliance and Management reports, which were the first BI based automation project at the company that I carried out. He remarked the following about the effectiveness of these reports: “The HR Compliance and Management reports are effectively a piece of a bigger puzzle... These reports helped us monitor any kind of delays, and we were able to set thresholds and really highlight anyone coming over a threshold of noncompliance with the reporting timeframe and have that as a one-stop shop embedded into an internal website that we could look at and at one glance and immediately see who is falling behind by how much and allow us to take immediate action on that.”

The **third question** is about my most challenging project, which really pushes PBI to its limit as it is effectively an ERP report for production management. He noted that “Two distinctions that need to be made. One is a scenario where everything happens under one roof. Let's say it's a small company and has an accounting system, and everything runs through that, all invoices in and out or components, etc. You can then leverage certain functionalities in that accounting system to do production management, bill of materials, composition of components into sub-assemblies, and then into finished products. This is obviously manageable, typically in an SME space, not at the granularity that you want, and then having to jump to a much heavier ERP solution, which requires a huge amount of time to set up has extensive licensing costs, has functionality overhead, etc. **This, we**

felt, is a very good middle ground for us to bridge this space without the additional specific complications that we faced in our setup.”

The **fourth question** focused on exploring the limitations of PBI. I explain these limitation in detail in the previous chapter. He concluded that “The biggest limitation of everything in Power BI at the moment, to take it to the next level of usability is the capability to input and modify certain bits of data in a controlled manner. If this problem is solved, then the usability and the overall value ad of the platform and the solutions that can be built on top of the platform will significantly increase.” I mentioned the PowerApps workaround multiple times throughout this thesis, but this option is not viable to the company since we use a master dataset, where most data sources are in input mode.

The **fifth question** was about the final goal for the use of BI for internal processes. As mentioned before, the process is still ongoing. He noted the following: “As I have touched on earlier, we are on a journey to have the full real-time P&L visibility on the ongoing activities that we are undertaking. The goal is to see in real time when certain indicators and metrics start to deteriorate beyond levels that we are not happy with, and it allows us to take corrective action, a lot sooner than otherwise.

The **last question** focused on the company culture necessary at SMEs to successfully implement BI solutions. He commented that “Power BI is, I would say, generally accessible in terms of cost and complexity to SMEs. One of the biggest limitations is honourably, the mindset of the managers and a lack of skills and experience in having the correct data governance approaches, which just does not come naturally to an SME organization. The other one is the actual plumbing to make sure that the relevant data sources are captured and streamlined into the reports so that everything can be automated. In the mid-term, what makes the most sense is to understand what real key indicators are and get those automated so they can be viewed in real time.” This aligns with the success factors highlighted in the literature review [5, 12].

Chapter 6

Conclusions

The thesis aims to demonstrate that with the evolving feature set of BI tools, it is possible for SMEs to use these tools for building DSS, small scale ERP systems and use them for BPA. Moreover, the main goal is to build a BI centred framework that supports these solutions in the planning phase.

The main research questions are the following:

1. *In what ways do BI tools contribute to the decision-making and BPA processes within SMEs, and how can these contributions be streamlined through a BI-centred framework?*
2. *What are the key factors that determine if a BI-centred approach can be used successfully for a given set requirements and what are the limitations of these solutions?*

The two tools explored in detail are Power BI and Tableau. Their feature set was discussed to give a transparent overview of the exact scenarios where they can be deployed. This includes data source options, ETL capacity, storage options, visualizations, deployment, security, embedding and licensing among others. For SMEs, the clear recommendation is Power BI, mainly due to its cost-effectiveness.

The thesis clearly demonstrates that BI tools can contribute to the decision-making processes of SMEs, and having a framework that outlines the main elements based on nine aspects can help SMEs navigate the architecture of these systems. This streamlines scalability, cost-effectiveness and it demonstrates what these BI tools are capable of.

The two main limitations of the investigated BI tools are the lack of a native feature to input data and the scalability of the solutions depending on the data landscape of the given

company. However, the latter in the case of SMEs can be overcome with optimization processes and following best practices in BI architecture. Thus, the main factors determining if a BI-centred approach can be used are data source types, data volume, data maturity and the desired feature set.

The thesis uses a case study-based approach of the SME *NGI Systems* to demonstrate the applicability of the framework and to showcase solutions that are beyond data visualization. The four main domains of the solutions are production management, HR compliance and management, financial reporting, and billing reporting. The case study is concluded by an interview with the CEO of the company to determine the impact of the solutions from a management standpoint.

This thesis provides a practical guide to BI implementation for SMEs, which was found to be the main gap in the current body of research. In addition, the feature set of the two market leaders were never explored holistically, which is addressed in this thesis. This guide explores every major feature of Tableau and PBI, which BI practitioners, executives, and business managers can follow during both the planning and development phases. It also offers a framework for determining the correct architecture, licensing, and external tool usage. Neither of these two aspects were found in other research papers.

The main limitation of this thesis is that it only explores two BI tools in-depth, and only one company is presented in the case study. Further research is needed to explore the feature set of other BI tools, besides the two market leaders. The case study should also be expanded to a larger sample size, both to showcase complex BI solutions for SMEs and to further demonstrate the applicability of the BI-centred frameworks.

Bibliography

- [1] M. R. Llave. “Business intelligence and analytics in small and medium-sized enterprises: A systematic literature review”. In: *Procedia Computer Science* 121 (2017), pp. 194–205.
- [2] D. Y. Lin et al. *Beyond financials: Helping small and medium-size enterprises thrive*. McKinsey & Company, 2022.
- [3] S. Ransbotham, D. Kiron, and P. K. Prentice. “Beyond the hype: the hard work behind analytics success”. In: *MIT Sloan Management Review* 57.3 (2016).
- [4] M. Amini and A. M. Abukari. “ERP systems architecture for the modern age: A review of the state of the art technologies”. In: *Journal of Applied Intelligent Systems and Information Sciences* 1.2 (2020), pp. 70–90.
- [5] H. Chen, R. H. Chiang, and V. C. Storey. “Business intelligence and analytics: From big data to big impact”. In: *MIS Quarterly* (2012), pp. 1165–1188.
- [6] S. Chaudhuri, U. Dayal, and V. Narasayya. “An overview of business intelligence technology”. In: *Communications of the ACM* 54.8 (2011), pp. 88–98.
- [7] K. Schlegel et al. *Magic quadrant for analytics and business intelligence platforms*. ID G00768632. Gartner, 2023.
- [8] G. Srivastava et al. “A review of the state of the art in business intelligence software”. In: *Enterprise Information Systems* 16.1 (2022), pp. 1–28.
- [9] D. Edge, J. Larson, and C. White. “Bringing AI to BI: enabling visual analytics of unstructured data in a modern Business Intelligence platform”. In: *Extended abstracts of the 2018 CHI conference on human factors in computing systems*. 2018, pp. 1–9.
- [10] R. K. Yin. *Case study research: Design and methods*. Vol. 5. Sage, 2009.

- [11] P. Baxter and S. Jack. “Qualitative case study methodology: Study design and implementation for novice researchers”. In: *The Qualitative Report* 13.4 (2008), pp. 544–559.
- [12] T. Davenport. “Analytics 3.0: In the new era, big data will power consumers products and services”. In: *Harvard Business Review* 91 (2013), pp. 64–72.
- [13] H. P. Luhn. “A business intelligence system”. In: *IBM Journal of Research and Development* 2.4 (1958), pp. 314–319.
- [14] J. C. Nwokeji and R. Matovu. “A systematic literature review on big data extraction, transformation and loading (etl)”. In: *Intelligent Computing: Proceedings of the 2021 Computing Conference, Volume 2*. Springer International Publishing, 2021, pp. 308–324.
- [15] A. McAfee et al. “Big data: the management revolution”. In: *Harvard Business Review* 90.10 (2012), pp. 60–68.
- [16] J. Dean and S. Ghemawat. “MapReduce: simplified data processing on large clusters”. In: *Communications of the ACM* 51.1 (2008), pp. 107–113.
- [17] M. F. Tutunea and R. V. Rus. “Business intelligence solutions for SME’s”. In: *Procedia Economics and Finance* 3 (2012), pp. 865–870.
- [18] D. C. Yen, D. C. Chou, and J. Chang. “A synergic analysis for Web-based enterprise resources planning systems”. In: *Computer Standards & Interfaces* 24.4 (2002), pp. 337–346.
- [19] T. Davenport. “Putting the Enterprise into the Enterprise System”. In: *Harvard Business Review* 76 (1998), pp. 121–131.
- [20] D. C. Chou, H. Bindu Tripuramallu, and A. Y. Chou. “BI and ERP integration”. In: *Information Management & Computer Security* 13.5 (2005), pp. 340–349.
- [21] C. Caserio. “Relationships between ERP and business intelligence: an empirical research on two different upgrade approaches”. In: *Information Technology and Innovation Trends in Organizations: ItAIS: The Italian Association for Information Systems*. Heidelberg: Physica-Verlag HD, 2011, pp. 363–370.
- [22] R. Russman, L. F. Seymour, and J. P. Van Belle. “Integrating BI Information into ERP Processes-Describing Enablers”. In: *International Conference on Enterprise Information Systems*. Vol. 2. SCITEPRESS. 2017, pp. 241–248.

- [23] T. H. Davenport et al. “Data scientist: the sexiest job of the 21st century”. In: *Harvard Business Review* 90.10 (2012), pp. 70–6, 128.
- [24] N. A. El-Adaileh and S. Foster. “Successful business intelligence implementation: a systematic literature review”. In: *Journal of Work-Applied Management* 11.2 (2019), pp. 121–132.
- [25] B. Puklavec, T. Oliveira, and A. Popovič. “Understanding the determinants of business intelligence system adoption stages: An empirical study of SMEs”. In: *Industrial Management & Data Systems* 118.1 (2018), pp. 236–261.
- [26] Y. Talaoui and M. Kohtamäki. “35 years of research on business intelligence process: a synthesis of a fragmented literature”. In: *Management Research Review* 44.5 (2020), pp. 677–717.
- [27] M. Russo and A. Ferrari. *Introducing Microsoft Power BI*. Redmond: Microsoft Press, 2016.
- [28] M. Russo and A. Ferrari. *The Definitive Guide to DAX: Business intelligence with Microsoft Power BI, SQL Server Analysis Services, and Excel*. Redmond: Microsoft Press, 2019.
- [29] R. Rad. *Pro Power BI Architecture: Sharing, Security, and Deployment Options for Microsoft Power BI Solutions*. New York: APress, 2023.
- [30] R. Sleeper. *Practical Tableau: 100 Tips, Tutorials, and Strategies from a Tableau Zen Master*. Sebastopol: O’Reilly Media, 2018.
- [31] R. Sleeper. *Innovative Tableau: 100 More Tips, Tutorials, and Strategies*. Sebastopol: O’Reilly Media, 2020.
- [32] T. Magoma, S. Khumalo, and T. Du Plessis. “Affordability of IBM Cognos business intelligence tool features suitable for small-and medium-sized enterprises’ decision-making”. In: *South African Journal of Information Management* 23.1 (2021), pp. 1–9.
- [33] V. Vashisht et al. “An Effective Approach for Integrating Microsoft Power BI Application with Python for Predictive Analytics”. In: *Micro-Electronics and Telecommunication Engineering: Proceedings of 4th ICMETE 2020*. Singapore: Springer Singapore, 2021, pp. 469–477.

- [34] W. Needham. *The story of Houston Electrics - Intro to Microsoft Fabric*. <https://www.skool.com/microsoft-fabric/classroom/5758d141?md=3c63e0d5cd934971b8ad2aa46b907bdd>. Retrieved April 16, 2024. 2023.
- [35] K. Puls and M. Escobar. *Master Your Data with Power Query in Excel and Power BI: Leveraging Power Query to Get & Transform Your Task Flow*. Chicago: Holy Macro! Books, 2019.
- [36] Otarb. *Learn which Python packages are supported - Power BI*. <https://learn.microsoft.com/en-us/power-bi/connect-data/service-python-packages-support>. Retrieved April 20, 2024. Jan. 2024.
- [37] Davidiseminger. *Learn which R packages are supported - Power BI*. <https://learn.microsoft.com/en-us/power-bi/connect-data/service-r-packages-support>. Retrieved April 13, 2024. Nov. 2023.
- [38] Otarb. *Create Power BI visuals using Python in Power BI Desktop - Power BI*. <https://learn.microsoft.com/en-us/power-bi/connect-data/desktop-python-visuals#known-limitations>. Retrieved April 9, 2024. Jan. 2023.
- [39] J. Serra. *Power BI as a enterprise data warehouse solution?* <https://www.jamesserra.com/archive/2022/06/power-bi-as-a-enterprise-solution/>. Retrieved April 24, 2024. July 2022.
- [40] C. Allchin. *Tableau Prep: Up & Running*. O'Reilly Media, 2020.
- [41] *tableau/TabPy*. <https://github.com/tableau/TabPy>. Retrieved April 19, 2024. n.d.
- [42] *Tableau Platform Architecture*. https://help.tableau.com/current/blueprint/en-us/bp_server_architecture.htm. Retrieved April 29, 2024. 2023.
- [43] M. Meier and D. Baldwin. *Mastering Tableau 2021: Implement advanced business intelligence techniques and analytics with Tableau*. Packt Publishing, 2021.
- [44] *Understanding license models and product keys*. https://help.tableau.com/current/server/en-us/license_product_keys.htm. Retrieved April 29, 2024. n.d.
- [45] *Tabular Editor 3*. <https://tabulareditor.com/>. Retrieved April 10, 2024. n.d.

- [46] *Workbooks and sheets*. https://help.tableau.com/current/pro/desktop/en-us/envIRON_workbooksandsheets.htm. Retrieved April 5, 2024. n.d.
- [47] WayNext. *Write-Back Cloud for Tableau*. <https://writeback4t.com/products/write-back-cloud-tableau/#pricing>. Retrieved April 24, 2024. Mar. 2024.
- [48] *tableau/tabjolt*. <https://github.com/tableau/tabjolt>. Retrieved April 19, 2024. n.d.
- [49] A. Bramel. *Tableau Server vs. Tableau Cloud (Online): What's Best for Your Business?* <https://www.phdata.io/blog/tableau-server-vs-tableau-cloud/>. Retrieved April 20, 2024. Oct. 2022.