INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA DE INTELIGÊNCIA ARTIFICIAL




Assistente Virtual Inteligente para Acesso a Dados de Negócio

GUILHERME LIMA FRANCO
outubro de 2025

P.PORTO

Intelligent Virtual Assistant for Business Data Access

Guilherme Lima Franco
Student No.: 1201437

Dissertation for obtaining the Degree of Master in Artificial Intelligence Engineering

Supervisor: Luís Manuel Silva Conceição

Evaluation Committee:
President:
Ana Maria Neves Almeida Baptista Figueiredo, Associate Professor, Institute of Engineering, Polytechnic of Porto

Vogals:
Francisco Supino Marcondes, Assistant Professor, School of Engineering, University of Minho
Luís Manuel Silva Conceição, Assistant Professor, Institute of Engineering, Polytechnic of Porto

Porto, October 2, 2025

Abstract

Modern organisations increasingly struggle to access and interpret enterprise data that is dispersed across isolated Business Information Systems (BIS). These silos hinder the ability to obtain a unified view of information, which is essential for timely and informed decision-making. Advances in Large Language Models (LLMs) offer the possibility of querying such data in natural language, thereby lowering the technical barrier for business users. However, the adoption of these models in corporate environments is constrained by concerns over data privacy, regulatory compliance, and the high operational costs of cloud-based solutions. These challenges underline the need for on-premises, resource-efficient approaches that preserve control over sensitive information.

This dissertation presents an intelligent virtual assistant that answers business questions by orchestrating Model Context Protocol (MCP) tools to inspect schemas, draft explicit-projection SQL, validate read-only execution, and ground responses in results from a local Microsoft SQL Server instance of AdventureWorksDW2022. No model fine-tuning is performed; instead, the approach combines runtime schema filtering, deny-list validation, and prompt scaffolding to minimise hallucinations and enforce governance.

A controlled evaluation over 52 representative prompts compares three configurations: a prompt-only baseline (B0), MCP with unfiltered schemas (B1), and a curated setup with filtering and explicit projections (S). The curated configuration yields substantially higher execution accuracy and fewer schema-error incidents than both baselines, demonstrating that governed tool use materially increases correctness without relaxing the privacy posture on a single on-premises workstation. Latency observations are reported descriptively and are attributable primarily to model generation rather than orchestration.

These findings support the feasibility of privacy-preserving, on-premises conversational analytics under the EU General Data Protection Regulation (GDPR) and the EU Artificial Intelligence Act (Regulation (EU) 2024/1689), and suggest practical next steps: broadening schema coverage, refining curation policies, and exploring lighter local models and decoding strategies to improve interactivity.

Keywords: Model Context Protocol, Local LLMs, Data Privacy, Virtual Assistant, Business Information Systems, SQL Server.

Resumo

As organizações modernas enfrentam dificuldades crescentes no acesso e na interpretação de dados empresariais dispersos por diferentes BIS. A fragmentação destes sistemas dificulta a obtenção de uma visão integrada da informação, essencial para apoiar decisões estratégicas e operacionais. Os avanços recentes em LLMs oferecem a possibilidade de interagir com esses dados em linguagem natural, reduzindo barreiras técnicas para os utilizadores de negócio. Contudo, a sua adoção em ambientes corporativos é limitada por preocupações relacionadas com a privacidade, a conformidade regulatória e os elevados custos de soluções baseadas na cloud. Estas restrições reforçam a relevância de abordagens locais (on-premises), concebidas para garantir eficiência de recursos e maior controlo sobre a informação sensível.

Esta dissertação propõe um assistente virtual inteligente que responde a questões de negócio através da orquestração de ferramentas, via Model Context Protocol (MCP), que: (i) inspecionam o esquema da base de dados, (ii) geram instruções SQL com projeções explícitas, (iii) validam a execução em modo apenas-leitura e (iv) apresentam respostas ancoradas nos resultados de uma instância local do Microsoft SQL Server, com a base AdventureWorksDW2022. Não é realizado qualquer *fine-tuning* do modelo; em alternativa, combina-se filtragem de esquema em tempo de execução, validação por lista de exclusão (*deny-list*) e estruturação de *prompts* (*prompt scaffolding*) para reduzir alucinações e reforçar a governação.

A avaliação controlada, baseada em 52 *prompts* representativos, compara três configurações: (B0) referência apenas com *prompt*; (B1) MCP com esquema não filtrado; e (S) configuração curada, com filtragem e projeções explícitas. A configuração curada demonstrou maior precisão de execução e menor ocorrência de erros de esquema, evidenciando que o uso governado de ferramentas melhora substancialmente a correção sem comprometer a privacidade, mesmo em hardware de baixo desempenho. As medições de latência foram analisadas de forma descritiva e atribuídas sobretudo ao tempo de geração do modelo, e não à orquestração.

Os resultados obtidos confirmam a viabilidade de uma abordagem de analítica conversacional em ambiente local, capaz de preservar a privacidade e assegurar conformidade com o GDPR e com o Regulamento Europeu da Inteligência Artificial (Regulamento (UE) 2024/1689). O trabalho aponta ainda perspetivas futuras, como o alargamento da cobertura de esquemas, o aperfeiçoamento de políticas de curadoria e a exploração de modelos locais mais leves e estratégias de decodificação que reforcem a interatividade.

Palavras-chave: Model Context Protocol, LLMs Locais, Privacidade dos Dados, Assistente Virtual, Sistemas de Informação Empresarial, SQL Server

Contents

List of Figures

List of Tables

List of Source Code

List of Acronyms

AHP Analytic Hierarchy Process.

AI Artificial Intelligence.

AI-FIS AI-based Fuzzy Inference System.

APIs Application Programming Interfaces.

ASR Automatic Speech Recognition.

BERT Bidirectional Encoder Representations from Transformers.

BIS Business Information Systems.

CNNs Convolutional Neural Networks.

CRM Customer Relationship Management.

DL Deep Learning.

DPIAs Data Protection Impact Assessments.

DW Data Warehouse.

ERP Enterprise Resource Planning.

FHIR Fast Healthcare Interoperability Resources.

FILLIS Factory Integrated Logic and Language Interface System.

FRIAs Fundamental Rights Impact Assessments.

GDPR General Data Protection Regulation.

IIoT Industrial Internet of Things.

IT Information Technology.

IVA Intelligent Virtual Assistant.

LDA Latent Dirichlet Allocation.

LLMs Large Language Models.

LSP Language Server Protocol.

LSTM Long Short-Term Memory.

MCP Model Context Protocol.

MES Manufacturing Execution Systems.

ML Machine Learning.

MLLMs Multimodal Large Language Models.

NLP Natural Language Processing.

OLAP Online Analytical Processing.

OLTP Online Transaction Processing.

RAG Retrieval-Augmented Generation.

RNNs Recurrent Neural Networks.

SLR Systematic Literature Review.

SMO Sequential Minimal Optimization.

TAM Technology Acceptance Model.

TF-IDF Term Frequency-Inverse Document Frequency.

VAD Voice Activity Detection.

Chapter 1

Introduction

This chapter provides the context for the work detailed in this thesis, outlining the problem statement and defining the objectives established to address it. Additionally, it presents an overview of the document's structure, offering a roadmap for the reader.

1.1 Context and Motivation

In today's rapidly evolving business landscape, organisations are increasingly reliant on sophisticated tools to manage and analyse critical data for decision making. Historically, before the advent of Business Information Systems (BIS), companies operated with manual record-keeping methods and relied heavily on the tacit knowledge of managers. These manual processes, while functional in simpler business contexts, often led to inefficiencies and errors, particularly as businesses expanded during the Industrial Revolution. Taylor (1911), often referred to as the father of scientific management, emphasised the importance of systematised processes to improve efficiency, a principle that underscored the transition from manual to mechanised systems.

With the emergence of mechanised systems in the 19th century, companies began adopting basic accounting and inventory management tools to better track their operations. This shift reflected what Drucker (1969), a pioneer in management theory, called the "information revolution" of its time, as organisations started recognising the value of structured data in driving productivity.

By the mid-20th century, advancements in computing technology introduced the possibility of automating business processes. Early computer systems, such as punch-card machines, were employed to handle payroll and inventory management, paving the way for the transformational role of computing in businesses. As noted by Neumann (1993), one of the architects of modern computing, the capacity for machines to process and store information efficiently marked a significant turning point for industries. These foundational systems laid the groundwork for the development of BIS, a broad category of software solutions designed to integrate, manage, and optimise business processes through centralised data management. Over time, different types of BIS have emerged to address specific organisational needs. This thesis focuses on three key types: Enterprise Resource Planning (ERP) systems, which integrate core business functions such as finance, supply chain, and human resources; Customer Relationship Management (CRM) systems, which enhance customer interactions, sales, and marketing processes; and Manufacturing Execution Systems (MES), which oversee and optimise production activities in manufacturing environments (Choudhury and Harrigan 2014; Kletti 2007; Wagner and Monk 2008). Collectively, these systems improve operational efficiency, ensure data consistency, and support informed decision-making across various business domains.

Among these tools, BIS have become indispensable, providing centralized platforms to integrate and manage various business operations, such as supply chain management, finance, and human resources. Researchers such as Davenport and Short (1990) have highlighted the role of integrated systems in enabling "process innovation", which allows organisations to reimagine and optimise workflows for competitive advantage. As businesses continue to grapple with increasing complexity, BIS not only streamline operations but also support data-driven strategies, empowering organisations to make informed decisions in real time.

1.2 Problem Statement

Despite the advancements in BIS and the transformative potential of Artificial Intelligence (AI), significant challenges remain in enabling seamless interaction between end-users and these systems. Current BIS are often complex and not user-friendly, which limits their accessibility to non-technical professionals. For example, research by Davis (1989) on the Technology Acceptance Model (TAM) highlights that perceived ease of use significantly affects user adoption of information systems. This complexity forces organisations to rely on Information Technology (IT) specialists for tasks like data extraction and analysis, creating operational bottlenecks and delaying critical decision-making processes.

The increasing volume and complexity of business data further exacerbate these issues, placing a high cognitive load on users and reducing overall efficiency. Companies like General Electric have reported difficulties in managing their vast and diverse data assets, which led to delays in generating actionable insights (Budagov 2020).

AI-driven solutions, such as virtual assistants leveraging Natural Language Processing and Machine Learning, offer the potential to bridge this gap. For example, IBM's Watson Assistant has been successfully integrated into several industries, including banking and healthcare, to provide users with conversational interfaces for complex tasks (Amendaño-Murrillo et al. 2020). These solutions allow users to interact with ERP systems more intuitively, without requiring extensive technical expertise. However, barriers to adoption persist, including difficulties in data integration from diverse sources, the need for scalable and adaptable systems, and concerns about trust and accuracy in AI-generated insights.

The advent of AI offers transformative solutions to these challenges. Key AI technologies, such as Natural Language Processing (NLP), Machine Learning (ML), and Deep Learning (DL) models, have demonstrated significant potential in simplifying complex data systems.

- NLP enables systems to interpret and respond to user queries in natural language, reducing the technical barrier for end-users. A notable example is Salesforce's Einstein AI, which leverages NLP to provide personalised insights and automate routine tasks for sales and marketing teams. This has improved user adoption by making data systems more accessible to non-technical users (Salesforce 2024).
- Machine learning models, when integrated with ERP systems, can autonomously analyse data, identify patterns, and provide predictive insights. For instance, SAP's AI-enabled ERP platform uses ML to forecast inventory needs, helping businesses like H&M reduce overstock and understock issues (SAP 2023).

- Virtual assistants, such as Microsoft's Dynamics 365 Copilot, serve as intermediaries between users and ERP systems, offering conversational interfaces that simplify complex queries. These assistants can dynamically adapt to user needs, reducing the learning curve for employees (Microsoft 2024b).

While the promise of AI-driven virtual assistants is substantial, implementing such solutions in business contexts involves several challenges:

- **Data Integration:** ERP systems often pull data from diverse sources, each with unique formats and standards. Consolidating this data into a coherent model requires robust data integration techniques. Amazon faced significant challenges integrating data across its global operations, which required the development of a custom AI-powered data lake for seamless consolidation (Forbes 2020).
- **customisation and Adaptability:** Every business has unique requirements and workflows. Virtual assistants must adapt to these specific needs dynamically. For example, Coca-Cola customised its AI assistant to address specific supply chain needs, enabling faster resolution of operational issues and enhancing productivity (Doering 2024).
- **Scalability:** As businesses grow, their data volumes and complexity increase. AI systems must scale accordingly to maintain performance and reliability. Google Cloud's BigQuery ML has been widely adopted by companies like Spotify to handle massive data volumes, demonstrating the scalability of AI in dynamic business environments (Müller et al. 2018).
- **Trust and Accuracy:** Business decisions often hinge on the information provided by these systems. Ensuring the accuracy and reliability of AI-generated insights is paramount to user adoption. Tesla's integration of AI in its supply chain faced initial skepticism due to errors in early predictive models, highlighting the critical need for robust validation and continuous learning mechanisms (TechCrunch 2021).

AI-driven solutions have the potential to transform the way organisations interact with their data and make decisions. By addressing challenges such as data integration, customisation, scalability, and trust, businesses can fully harness AI to optimise their operations. Successful implementations, like those by Salesforce, SAP, and Microsoft, serve as powerful examples of how AI can bridge the gap between complex BIS and end-users, empowering organisations to achieve greater efficiency and innovation.

1.3 Project Proposal

The development of an artificial intelligence-driven project focused on acquiring and analysing business information from repositories such as ERP, CRM, MES systems aims to enable managers to intuitively query and understand operational data. Every stage of the project will be constructed based on AI methodologies, encompassing data extraction, transformation, and loading, as well as the construction of a semantic model and query interpretation. The integration of various business repositories is crucial for consolidating information from multiple sources. By leveraging AI techniques, the system automatically extracts, transforms, and harmonises data, ensuring accessibility and structural efficiency. Semantic modelling, also AI-based, facilitates the understanding of business data logic by identifying patterns and relationships across different datasets. Advanced machine learning algorithms and natural language processing techniques ensure that data semantics are preserved and accurately interpreted by the system. The project further enables intelligent interactions between managers and the database, converting natural language inquiries into structured queries. The application of AI-driven natural language processing allows the system to grasp user intent and generate aggregated, contextually relevant responses. The presentation of responses extends beyond raw data, offering strategic insights that support decision-making. AI models analyse patterns, forecast trends, and detect anomalies to provide actionable and relevant information. Finally, the implementation of the project necessitates a secure and reliable environment, ensuring data governance and compliance. Through this structured, AI-based approach, the project revolutionizes business data analysis and enhances decision-making processes with efficiency and precision.

1.4 Objectives

To address the research problem and support the overall aim of this dissertation, a set of objectives has been defined. These objectives serve as a roadmap for guiding the design, implementation, and evaluation of the proposed intelligent virtual assistant, ensuring that the solution is both technically feasible and aligned with business and regulatory requirements.

The main objectives of this dissertation are:

1. To design an intelligent virtual assistant for business data access that enables users to interact with enterprise systems through natural language, facilitating more intuitive and efficient information retrieval.
2. To ensure privacy and regulatory compliance by adopting an on-premises and resource-efficient approach, thereby reducing dependency on external services and aligning with legal frameworks such as the GDPR and the AI Act.
3. To enhance the accuracy and relevance of responses by incorporating advanced natural language processing and information retrieval techniques, ensuring that users receive meaningful insights from complex business data.
4. To develop a methodological framework that achieves a balance between accuracy, efficiency, and scalability, making intelligent assistants viable for real-world business environments.
5. To demonstrate the feasibility of lightweight and efficient AI solutions that maintain strong performance while operating under practical resource constraints.
6. To identify and critically evaluate the challenges and opportunities in deploying intelligent assistants for business data access, including issues of integration, scalability, explainability, and user trust.

Taken together, these objectives frame the methodological choices and research questions developed in the subsequent chapters, ensuring coherence between the problem statement, the technological approach, and the intended contributions.

Having established what the dissertation seeks to achieve, the next section clarifies the ethical and legal constraints that bound these objectives and shape the design and evaluation.

1.5 AI governance in business: ethical, legal, and privacy challenges

The accelerated integration of AI into business operations presents profound ethical and legal challenges, particularly in terms of privacy, accountability, and regulatory compliance. The European legal framework, primarily defined by the General Data Protection Regulation (GDPR) and the AI Act, establishes clear obligations for businesses employing AI. The GDPR enforces strict guidelines on data handling and privacy, while the AI Act introduces a risk-based approach to AI regulation. Given that data is a vital corporate asset influencing decision-making and market competitiveness, ensuring compliance with these regulations is not only a legal necessity but also a strategic imperative. Mismanagement of data can result in substantial financial penalties, reputational harm, and operational disruptions (Voigt and Bussche 2017). The objectives defined in Section 1.4 must be executed within clear ethical and legal constraints.

This section examines the ethical dimensions of AI, the regulatory constraints imposed by GDPR, and the compliance requirements set forth by the AI Act, focusing on their implications for business enterprises.

1.5.1 Ethical considerations in AI development

Ensuring fairness, accountability, and transparency in AI development is essential for businesses seeking to leverage intelligent systems responsibly. Bias and discrimination within AI models, particularly in recruitment and financial services, pose substantial risks that can undermine corporate integrity and violate anti-discrimination laws (Barocas et al. 2023). For example, AI-based hiring platforms have in some documented cases systematically downgraded CVs containing gendered language, reflecting historical biases in training data, thereby contravening equal opportunity principles and exposing organisations to reputational and legal consequences.

AI systems must also operate with a high degree of explainability to facilitate trust, regulatory compliance, and meaningful oversight. However, many machine learning models function as opaque “black boxes,” making it difficult to trace or rationalise their decision-making processes (Doshi-Velez and Kim 2017). This opacity becomes particularly problematic in high-stakes domains such as credit scoring or insurance claim approvals, where the absence of clear audit trails has led to legal disputes and regulatory scrutiny.

Beyond fairness and interpretability, the handling of sensitive data raises critical ethical considerations. The unauthorised collection and processing of personal or business-sensitive information not only threaten individual privacy rights but also expose organisations to severe penalties under regulations such as the GDPR (Wachter et al. 2018). This risk is magnified in AI-driven CRM systems and targeted marketing platforms that profile consumer behaviour. Notably, the substantial fine imposed on H&M in 2020 for excessive monitoring of employee data illustrates the tangible consequences of overstepping data protection boundaries.

Moreover, AI applications deployed in business environments frequently process highly confidential corporate information, including proprietary workflows, trade secrets, supplier agreements, and strategic plans. Virtual assistants integrated with ERP or supply chain systems may inadvertently expose custom pricing models or sensitive logistics arrangements. Such disclosures could erode competitive advantage and breach contractual non-disclosure obligations, underscoring the need for robust data governance measures and clear internal access controls.

The phenomenon of unintended memorisation further complicates these ethical challenges. Large language models and retrieval-augmented systems, even when deployed on-premise, can unintentionally encode fragments of sensitive prior interactions, risking inadvertent disclosure in subsequent responses (Bender et al. 2021; Yao et al. 2024). For instance, an AI assistant trained on historical procurement data might inadvertently reveal details of an unannounced supplier contract when queried about unrelated operations. This highlights the imperative for technical safeguards, such as differential privacy techniques and rigorous audit mechanisms, to ensure that sensitive data remains secure and insulated from unintended propagation.

Autonomous decision-making in AI applications—particularly those influencing credit approvals, hiring, insurance assessments, and customer segmentation—therefore necessitates carefully designed human oversight frameworks. Documented intervention points and override capabilities are essential to mitigate operational and financial risks, as well as to uphold accountability in line with emerging regulatory expectations, such as those codified in the AI Act (European Parliament 2024). Without such controls, businesses risk deploying systems whose failures could result not only in financial losses but also in significant erosion of stakeholder trust.

1.5.2 GDPR and AI: data protection and compliance for businesses

The GDPR (Regulation (EU) 2016/679) mandates stringent data protection measures for businesses utilising AI. It requires that AI-driven data processing be lawful, fair, and transparent, ensuring that users are informed about how their data is collected and used (Kuner et al. 2020). Businesses must define clear, legitimate purposes for data processing and adhere to data minimisation principles by collecting only the information essential for operational needs. Individuals possess extensive rights under the GDPR, including access to their personal data, rectification of inaccuracies, and the ability to object to automated decision-making processes that significantly impact them (Zarsky 2017). Compliance strategies must include conducting Data Protection Impact Assessments (DPIAs) to evaluate potential risks and integrating privacy-by-design principles into AI systems (Tene and Polonetsky 2014). Transparency and explainability must be embedded into AI models to enable regulators, customers, and corporate stakeholders to understand their functionality. Companies must also implement clear consent mechanisms, ensuring that individuals can manage their data preferences effectively.

1.5.3 The AI Act: regulating AI risks in business applications

The AI Act, introduced by the European Commission and entering into force in August 2024, establishes a comprehensive risk-based regulatory framework for artificial intelligence within the European Union (European Parliament 2024). It categorises AI systems into distinct risk tiers—minimal, limited, high, and unacceptable—imposing graduated legal obligations based on potential societal and economic impacts. AI systems deemed to present an unacceptable risk, such as those involving unconsented real-time biometric identification in workplaces, are strictly prohibited. Conversely, high-risk AI systems, which encompass applications in financial services, recruitment, advanced employee analytics, and critical enterprise decision-making, are subject to rigorous compliance requirements. These obligations include robust data governance mechanisms to ensure data quality and bias mitigation, transparency measures that facilitate meaningful human oversight, and stringent security protocols to safeguard sensitive personal and business data. Organisations must also maintain comprehensive technical documentation that details training methodologies, datasets used, and system risk assessments, thereby enabling full traceability and facilitating regulatory accountability.

Legal obligations now in force under the EU AI Act include: (i) an expanded definition of *high-risk* AI to cover sophisticated employee monitoring beyond biometrics and strategic business forecasting; (ii) enhanced transparency and accountability (e.g., model cards, dataset disclosures, and provenance/watermarking for generative models); (iii) a *mandatory EU registry* of high-risk AI; (iv) reinforced human-oversight provisions with a presumption of non-compliance if no human-in-/on-the-loop is documented; (v) coupling with the EU Cyber Resilience Act for systems processing sensitive data; (vi) standardised bias audits and Fundamental Rights Impact Assessments (FRIAs); and (vii) increased administrative sanctions up to 7% of worldwide turnover for severe infringements (European Parliament 2024).

Enterprises operating in the EU must determine whether deployments fall within the expanded high-risk categories, register such systems in the EU database, maintain complete technical documentation (including bias-mitigation protocols), and, where generative/foundational models are used, implement watermarking/provenance and cybersecurity safeguards consonant with the AI Act and the Cyber Resilience Act. Non-compliance carries significant financial and reputational risks, making rigorous governance an operational imperative.

1.5.4 Best practices for ethical AI deployment in business

To align with both ethical imperatives and evolving regulatory frameworks such as the GDPR and the AI Act, businesses should embed comprehensive governance structures throughout the AI lifecycle. Effective corporate AI strategies prioritise ethical AI design, ensuring that models operate within clearly delineated legal parameters and do not perpetuate discriminatory biases (Binns 2018). Multi-stakeholder governance (regulators, legal counsel, DPOs, and senior leadership) should evaluate privacy, data security, and fundamental-rights impacts. Continuous monitoring and auditing help maintain fairness, accuracy, and legality through robust audit trails and performance reviews (McAfee and Brynjolfsson 2017). Organisational culture matters: targeted training, clear policies on personal and business-sensitive data, and documented controls strengthen trust and credibility. Finally, explainability features that provide human-interpretable rationales support transparency duties under the AI Act and improve defendability in high-stakes contexts (Lipton 2018).

Regulatory implications for the dissertation. The legal and ethical framework outlined above directly constrains the research design and scope developed in later chapters. Concretely, the dissertation will: (i) prioritise data minimisation and purpose limitation, restricting access to only the attributes strictly necessary for the demonstrations; (ii) favour on-premises processing and strong access controls to reduce data-transfer risk and support accountability; (iii) avoid high-risk contexts under the EU AI Act (e.g., recruitment, credit-worthiness, or biometric identification) and exclude automated decision-making that produces legal or comparably significant effects on individuals; (iv) embed human oversight checkpoints and full audit logging to enable traceability and post-hoc review; (v) prepare the documentation expected by GDPR and the AI Act (records of processing, risk analysis, and technical/organisational measures), aligning with the phased application of the Act;

and (vi) plan for transparency to end-users (clear notices when AI assistance is involved and provenance/watermarking where synthetic content may arise). These constraints guide architectural choices, experimental scope, and evaluation criteria presented later, ensuring that the technical artefact is designed and assessed within a compliant, privacy-preserving envelope.

In summary, artificial intelligence presents substantial opportunities but also complex ethical and legal challenges that must be navigated carefully. Compliance with GDPR and the AI Act is indispensable for safeguarding data privacy and ensuring that AI systems align with legal and ethical standards. By embedding transparency, accountability, and regulatory adherence into AI development and deployment, businesses can harness AI's potential responsibly while mitigating legal and reputational risks. The evolution of AI governance frameworks will continue to shape enterprise AI, necessitating continuous adaptation to emerging standards (Calo 2018).

1.6 Document Structure

This dissertation is divided into five chapters that progress from foundational context to detailed technical implementations and concluding remarks.

The current chapter, Chapter 1, outlines the context, motivation, problem statement, and objectives of the dissertation, and situates the work within the ethical, legal, and privacy governance landscape for enterprise AI—framed by the GDPR and the EU AI Act. It also presents the overall structure of the document.

Chapter 2 reviews the main concepts and technologies underpinning the study, including artificial intelligence methods, business information systems, and large language models. It also explores issues of data privacy, introduces Ollama as an example of privacy-preserving deployment, and analyses the Model Context Protocol (MCP) as a tool integration pattern. The chapter concludes with a systematic literature review that identifies trends, gaps, and challenges in current research.

Chapter 3 consolidates the data foundation and the methodological framework. It first describes the datasets considered in this study—focusing on enterprise-style samples such as Microsoft AdventureWorks—together with the criteria for selection, comparison of available options, and the preprocessing required to support subsequent experimentation. Building on that foundation, it then presents the proposed methodology for constructing the intelligent virtual assistant, including system architecture and tool orchestration, model and runtime choices, query understanding, retrieval-augmented and SQL-grounded generation, and the trade-offs between performance, scalability, and efficiency for on-premises deployment.

Chapter 4 reports the experimentation conducted to evaluate the proposed solution. It defines the evaluation setup, including hardware and software configurations, and introduces a stratified task suite of representative business queries over AdventureWorksDW2022. The chapter then presents the metrics, baselines, and experimental protocol, followed by results concerning accuracy, responsiveness, safety, and governance. Threats to validity are considered, and qualitative case studies illustrate the system's behaviour under both successful and failure scenarios. The findings collectively assess the feasibility of deploying an on-premises, privacy-preserving assistant under real-world constraints.

Chapter 5 summarises the main findings of the research, evaluates how the objectives and research questions have been addressed, and discusses the contributions and limitations of the work. It also identifies directions for future research and provides final remarks.

This structure ensures a logical progression from conceptual foundations to methodological development and experimental validation.

Chapter 2

State of the art

This chapter provides an overview of the key technologies and concepts underpinning this research. It first explores the role of BIS and the advancements in AI, including ML, DL, and NLP. The transformative impact of Large Language Models (LLMs) and Multimodal Large Language Models (MLLMs) is also discussed, with particular attention to the data privacy challenges these models present in business contexts, along with the synergies between AI and BIS.

Additionally, the chapter presents a Systematic Literature Review (SLR), which examines existing research on AI-driven solutions in BIS. The review follows a structured methodology to identify relevant studies, analyse current trends, and highlight challenges and opportunities in integrating AI with enterprise systems. This analysis provides a foundation for the subsequent research in this thesis.

2.1 Fundamental Concepts and Technologies

This section provides an overview of the key concepts and technologies that form the theoretical and technological background of this thesis. It explores the role of BIS in managing enterprise data, the advancements in AI that enhance data retrieval and decision-making, and the impact of emerging technologies such as ML, DL, NLP, and LLMs. By examining these foundational elements, this section establishes the basis for understanding the integration of AI-driven virtual assistants in business environments.

2.1.1 Business Information Systems (BIS)

BIS represent a critical framework within modern organisations, integrating information technology with core business processes to improve decision-making, operational efficiency, and strategic competitiveness (Davenport 2018). As explored by Wagner and Monk (2008), BIS encompasses a range of applications that support the management of business functions through the consolidation and analysis of data. Recent studies, such as those by Choudhury and Harrigan (2014), emphasise the transformative potential of BIS in creating value through enhanced information flow and process optimisation.

BIS are integral to organisational operations, providing platforms for managing and analysing business processes. Despite the wide range of applications that BIS encompass, this thesis focuses on three core components: ERP, CRM, and MES.

- **ERP Systems**: ERP platforms integrate core business functions, offering a unified view of processes such as finance, supply chain, and human resources. Studies highlight the benefits of ERP systems in improving operational efficiency and decision-making (Wagner and Monk 2008).

- **CRM Systems**: CRM tools enhance customer engagement by managing interactions, analysing customer behaviour, and providing actionable insights. Advances in AI have enabled predictive analytics in CRM, transforming customer service and sales strategies (Choudhury and Harrigan 2014).
- **MES Systems**: MES platforms oversee manufacturing processes, facilitating real-time monitoring and optimisation of production workflows. These systems play a crucial role in ensuring product quality and operational efficiency (Kletti 2007).

2.1.2 Machine Learning and Deep Learning

ML and DL are pivotal technologies underpinning modern advancements in AI. ML encompasses algorithms that enable systems to learn patterns and make predictions or decisions without explicit programming. Its methods are broadly classified into supervised learning, unsupervised learning, and reinforcement learning (Murphy 2013). Supervised learning focuses on labelled data to train predictive models, while unsupervised learning uncovers hidden structures within unlabelled datasets. Reinforcement learning optimises sequential decision-making by rewarding desirable outcomes (Sutton and Barto 1998).

DL, a subset of ML, has gained prominence due to its ability to model complex representations through multi-layered neural networks (LeCun et al. 2015). Architectures such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have demonstrated efficacy in processing image and sequence data, respectively. Transformers, introduced by Vaswani et al. (2017), have revolutionised DL by enabling parallel processing of sequence data, significantly improving scalability and performance in applications such as NLP.

2.1.3 Natural Language Processing (NLP)

NLP bridges the interaction between humans and machines, enabling systems to process, understand, and generate human language. Early NLP systems relied on rule-based approaches, but modern techniques leverage statistical and neural models (Jurafsky and Martin 2007). The development of word embeddings, such as Word2Vec and GloVe, marked a significant leap in representing semantic relationships (Mikolov et al. 2013; Pennington et al. 2014).

Transformer-based models, such as BERT (Devlin et al. 2019) and GPT (Radford et al. 2019), further advanced NLP by introducing attention mechanisms to capture contextual nuances. These models have been employed across diverse applications, including sentiment analysis, machine translation, and conversational agents, showcasing the versatility of modern NLP frameworks (Brown et al. 2020).

2.1.4 Large Language Models (LLMs) and Multimodal Large Language Models (MLLMs)

LLMs represent a significant advancement in AI, enabling generation of coherent, contextually relevant text across diverse domains. Examples include OpenAI's GPT series and Google's T5 (Raffel et al. 2020), which are trained on vast corpora encompassing diverse linguistic patterns and knowledge. These models leverage a two-phase paradigm of pre-training followed by fine-tuning: pre-training builds general-purpose linguistic understanding, while fine-tuning adapts the model for domain-specific tasks (Bommasani et al. 2022).

MLLMs extend this ability by processing multiple data modalities, such as text, images, and audio. For instance, GPT-4 Vision (OpenAI) and Google's PaLM-E incorporate multimodal inputs to facilitate reasoning across heterogeneous sources (Tsimpoukelli et al. 2021). These enhancements hold promise for transformational applications in settings such as healthcare diagnostics and intelligent enterprise systems.

Data Privacy Considerations in Large Language Models and the Case for Small LLMs

While LLMs have demonstrated remarkable capabilities in natural language understanding and generation, a critical concern in their deployment is the use of user data for continuous retraining. Many commercial LLMs, such as OpenAI's GPT series, Google's Gemini, and Meta's LLaMA, rely on iterative training cycles that incorporate user interactions to refine model accuracy and responsiveness (Bommasani et al. 2022). This process, known as continual learning, enables these models to adapt to evolving language patterns, improve contextual reasoning, and enhance performance on real-world queries. However, this approach also raises significant concerns regarding data privacy, security, and proprietary information leakage (Bender et al. 2021).

Retraining LLMs using user data can inadvertently lead to unintended memorisation of sensitive or proprietary information, exposing businesses and individuals to potential risks of data breaches or regulatory non-compliance (Yao et al. 2024). Additionally, reliance on large-scale data aggregation poses ethical challenges, as user-generated content may be used without explicit consent, leading to concerns about intellectual property violations and data sovereignty.

To mitigate these risks, this thesis explores the implementation of small LLMs that do not retain or utilise user data for retraining. Unlike foundational models that rely on cloud-based infrastructure for continual updates, small LLMs can be deployed on-premises or in privacy-preserving environments, ensuring that user interactions remain strictly within organisational boundaries. For instance, Zhang et al. (2024) proposed a framework called CoGenesis, which integrates large and small models to address privacy concerns by deploying small models on local devices, thereby safeguarding user data.

The key advantages of small LLMs include:

- **Enhanced Data Privacy** – On-device or self-hosted deployment ensures that sensitive business data is not transmitted to external servers, reducing risks associated with third-party data access.
- **Regulatory Compliance** – Organisations can maintain compliance with data protection frameworks such as GDPR (General Data Protection Regulation) and CCPA (California Consumer Privacy Act) by ensuring that user interactions are not stored or reused for model training.
- **Optimised Resource Usage** – Small LLMs require significantly fewer computational resources compared to their larger counterparts, making them suitable for edge computing and enterprise applications with constrained processing capabilities.

- **Customisation & Domain Adaptation** – Unlike general-purpose LLMs, small models can be fine-tuned with domain-specific data without exposing proprietary information to external training pipelines.

For instance, in business intelligence applications, a self-hosted small LLM can be integrated within an enterprise ERP (Enterprise Resource Planning) system to facilitate natural language queries without transmitting corporate data outside the organisation. Similarly, in customer service automation, a fine-tuned model can assist in context-aware response generation without retaining user conversations for future training.

Ollama as an Example of Privacy-Preserving LLM Deployment

A practical example of this approach is Ollama, an open-source orchestration framework that allows LLMs to be executed entirely on local hardware, supporting macOS, Linux, and Windows environments. Ollama enables the running of a variety of open-source models offline, ensuring that all user prompts, retrieved context, and generated outputs remain within the organisation's infrastructure. This offline-first design directly addresses privacy and compliance concerns, particularly for sectors such as healthcare and finance, where GDPR and other data-protection regulations prohibit uncontrolled cross-border data flows.

Ollama integrates with developer workflows via "Modelfiles", supports GPU acceleration, and provides observability features such as "thinking mode" for secure inspection of reasoning processes. It is increasingly being used in hybrid architectures where sensitive processing occurs locally while optional calls to cloud-based frontier models are mediated through secure enclaves (Ollama 2025a).

However, as with any software framework, security considerations remain essential. Recent vulnerability disclosures in earlier Ollama versions (e.g., file enumeration and model-poisoning risks) underline the importance of regular updates, rigorous access control, and deployment within hardened environments.

Ollama's approach aligns with the growing research focus on on-premises privacy-preserving AI. Frameworks such as OnPrem.LLM similarly provide infrastructure for local LLM inference, supporting backends like llama.cpp, vLLM, Hugging Face Transformers, and Ollama itself (Maiya 2025). These tools facilitate the integration of Retrieval-Augmented Generation (RAG) pipelines, quantisation, and GPU acceleration into secure, no-code or low-code environments, reinforcing the feasibility of deploying LLM-driven business assistants without sacrificing data governance.

By leveraging privacy-preserving LLM toolchains such as Ollama, enterprises can realise the advantages of intelligent assistants while maintaining full control over sensitive information, satisfying both operational and regulatory requirements.

LM Studio as a Privacy-Preserving Desktop LLM Runner

LM Studio is a desktop-first runtime that enables local execution of open-source language models across macOS, Windows, and Linux, with optional GPU acceleration through platform-native backends. By downloading models to the workstation and executing inference locally, LM Studio confines prompts, intermediate context, and generated outputs to the organisation's infrastructure, an arrangement that aligns with privacy and data-sovereignty requirements frequently encountered in regulated domains. The tool emphasises ease of adoption: installation is lightweight, model selection is curated through an integrated catalogue, and defaults are sensible for non-specialist users seeking to evaluate on-premises NLP capabilities without standing up server-grade tooling (Element Labs 2025).

Beyond interactive chat, LM Studio can expose a local HTTP endpoint that emulates common inference APIs, enabling rapid integration with existing applications, test harnesses, and orchestration layers. Practically, this permits drop-in experimentation with prompt templates, system instructions, and temperature controls while maintaining operational separation from external services. Developers can profile latency and throughput under different quantisation levels, adjust context-window usage, and observe the impact of batching on responsiveness, all within a contained environment suitable for enterprise pilots. Where available, GPU support leverages vendor runtimes to reduce token latency; CPU-only modes remain viable for small models or constrained hardware (Element Labs 2025).

As with any local-execution framework, secure deployment hygiene remains essential. Administrators should restrict file-system permissions, control model provenance, and apply routine updates to mitigate risks such as supply-chain tampering or inadvertent exposure through misconfigured local endpoints. Logging should be scoped to operational telemetry rather than sensitive prompt content, and network access should be explicitly disabled where offline operation is a requirement. These controls mirror standard enterprise hardening practices and are compatible with change-managed roll-outs (Element Labs 2025).

In sum, LM Studio offers a pragmatic path to privacy-preserving LLM experimentation and integration on a single machine. Its offline-first posture, API compatibility, and support for quantised open-source models make it a credible complement to server-oriented stacks, enabling organisations to prototype conversational and analytical assistants without exporting data or relinquishing governance.

2.1.5 Model Context Protocol (MCP) as an Integration Pattern for AI Assistants in BIS

The Model Context Protocol (MCP), proposed by Anthropic in late 2024, is an open standard designed to enable structured communication between large language models (LLMs) and external tools, services, and data sources (Anthropic 2024). MCP provides a lightweight but formalised integration framework that allows LLMs to dynamically discover, negotiate, and invoke external capabilities during a conversational session.

Its emergence can be contextualised within the broader trajectory of tool-augmented LLMs. Earlier approaches such as OpenAI's function calling API or LangChain's tool abstractions were proprietary, model-specific, and lacked interoperability (O'Leary 2024a). MCP instead positions itself as an analogue to the Language Server Protocol (LSP) in software development: a standardised contract that mediates communication between clients (LLMs) and servers (tool providers). Similar precedents exist in computational research, such as Jupyter's messaging protocol, which decoupled kernel execution from notebook interfaces. MCP thus addresses the fragmentation of AI tool integration by offering a vendor-neutral, open standard.

The motivation for MCP is particularly strong in BIS, where LLM-based assistants must interact with heterogeneous data repositories (ERP, CRM, MES) under strict compliance constraints. Without a standard protocol, organisations face ad-hoc connectors, duplicative integrations, and potential vendor lock-in.

Architecture & Protocol Semantics

The MCP architecture formalises the contract between conversational LLM clients and external capabilities by specifying roles, discovery, lifecycle control, invocation semantics, validation, and failure handling. This section focuses on how these elements compose to deliver interoperable, governable, and auditable tool use in enterprise contexts. MCP distinguishes three principal roles:

- **Client** — the AI assistant or LLM runtime that initiates discovery, opens sessions, and invokes tools.
- **Server** — the endpoint that exposes a catalogue of tools and resources to the client.
- **Tool/Resource Providers** — domain capabilities surfaced by the server (e.g., read-only SQL execution; document retrieval).

The protocol is distinguished by several features that shape how these actors interact. A discovery phase ensures that, upon connection, clients are able to query servers for a complete registry of available tools together with their schemas and usage constraints. Each interaction unfolds within a session lifecycle, where permissions are explicitly scoped and a clear termination sequence is enforced. Communication itself is structured through JSON-RPC 2.0 transported over WebSockets or streams, which allows for bidirectional messaging, asynchronous invocation, and systematic error handling. In order to guarantee reliable interoperability, tools describe their inputs and outputs using JSON Schema, enabling clients to validate requests and responses consistently. Finally, error semantics are explicitly defined: failures are categorised, whether they arise from schema violations, execution errors, or permission denials, and appropriate retry semantics ensure that transient errors can be recovered gracefully.

Security & Privacy Model

MCP establishes explicit trust boundaries between LLMs and enterprise systems by sandboxing tool execution and enforcing the principle of least privilege. Tools can be restricted to read-only or parameterised operations, ensuring that sensitive systems are never directly exposed. From a GDPR perspective, MCP aligns with several core principles. Lawfulness and purpose limitation are ensured through the explicit declaration of scope and intent that each tool must provide. Data minimisation is achieved by transmitting only the strictly required structured outputs rather than wholesale datasets. Finally, integrity and confidentiality are supported through encrypted communication channels and authenticated endpoints.

These features also resonate with the obligations of the EU AI Act. In high-risk domains, such as financial forecasting or HR analytics, MCP provides the possibility of maintaining auditable records of all tool invocations, thereby facilitating traceability and regulatory oversight (European Parliament 2024).

The potential threats can be systematically analysed using the STRIDE model, which highlights risks such as spoofing, tampering, repudiation, information disclosure, denial of service, and elevation of privilege. Mitigations include authenticated sessions, JSON schema validation, signed responses, column-level whitelisting, back-pressure mechanisms, and sandboxed execution environments.

Governance & Observability

Beyond protocol semantics, effective enterprise integration requires a governance framework, and MCP provides several mechanisms in this regard. Policy enforcement is supported through the integration of role-based or attribute-based access controls at the level of each tool. Access is further constrained through session-based credentials or scoped tokens that determine what the client can invoke. To guarantee accountability and regulatory compliance, all requests and responses are serialised into persistent logs that support audit and replay. Additionally, MCP offers evaluation hooks that allow organisations to inspect or filter model outputs prior to their release, providing safeguards against bias, personally identifiable information leakage, or hallucination.

Performance & Reliability

The effectiveness of the MCP in enterprise contexts depends not only on its architectural semantics but also on its operational characteristics. Performance and reliability determine whether tool-augmented assistants can deliver responses within interactive timescales while maintaining robustness under varying workloads. These aspects are critical in BIS settings, where latency, fault tolerance, and deterministic behaviour directly affect usability, governance, and user trust. Performance considerations include:

- **Latency**: Each MCP call incurs JSON serialisation overhead; local tools achieve sub-10ms latency, but remote calls add network delays.
- **Back-Pressure**: Servers may signal overload, prompting clients to throttle requests.
- **Idempotency & Retry**: MCP defines request IDs to ensure deterministic retries.
- **Failure Isolation**: Tool crashes are contained; the LLM remains operational.

For on-premises BIS deployments, MCP's **local server mode** allows SQL queries, embeddings lookups, or file access without crossing organisational boundaries, aligning with low-latency and data-sovereignty constraints.

Interoperability & Extensibility

MCP's plug-and-play design allows new tools to be attached without retraining models. As shown in Table 2.1, MCP provides distinct advantages over alternative integration frameworks, including its open standard, capability discovery, JSON schema validation, and cross-vendor interoperability. In contrast, solutions such as OpenAI's Assistants API, LangChain, and LlamaIndex exhibit strong ecosystems but lack a unified standard, while custom gRPC/REST implementations provide control at the cost of high engineering overhead. This comparative analysis highlights MCP's position as a promising architectural standard for tool-augmented LLMs in enterprise contexts.

Framework	Strengths	Weaknesses
MCP	Open standard; capability discovery; JSON schema validation; cross-vendor interoperability	Immature ecosystem; limited production deployments
OpenAI Assistants API	Tight integration with GPT-4; hosted evaluation	Vendor lock-in; cloud-only
LangChain	Extensive ecosystem of tools; strong Python support	No unified standard; added integration overhead
LlamaIndex	Effective document ingestion; RAG-oriented workflows	Proprietary abstractions; weak governance
Custom gRPC/REST	Full control; optimised performance	High engineering cost; no common standardisation

Table 2.1: Comparison of MCP and alternative integration frameworks for tool-augmented LLMs.

MCP represents a promising architectural standard for integrating LLMs with BIS, especially in on-premises, privacy-sensitive contexts. Its formalism (schemas, capability discovery, session semantics) reduces integration fragility and supports compliance with GDPR and the AI Act.

However, MCP is not a universal solution. For simple, tightly scoped assistants that only query a single database, direct ODBC/SQL adapters or in-process function calls may be more efficient. MCP becomes compelling when multiple heterogeneous tools must be orchestrated, when auditability is paramount, or when vendor-neutral extensibility is required.

In this dissertation's applied context—an intelligent virtual assistant accessing SQL Server and document corpora on-premises—MCP offers a balanced pathway between interoperability, governance, and extensibility, albeit with the caveat of ecosystem immaturity.

2.1.6 Synergies Between BIS and AI

Multimodal models further extend these synergies by enabling holistic analyses across textual, visual, and numerical data within BIS, facilitating tasks such as intelligent inventory management and predictive maintenance (Tsimpoukelli et al. 2021). Despite these advancements, challenges persist, including data privacy concerns, integration complexities, and the need for domain-specific customisation.

The integration of AI with BIS has transformed how organisations manage data, automate processes, and enhance decision-making. AI-driven solutions leverage ML, DL, and NLP to improve efficiency, accuracy, and user interaction across various BIS applications (Davenport and Short 1990).

One significant area of AI-BIS synergy is ERP systems, where AI enhances forecasting, automation, and decision support. For instance, SAP's AI-powered ERP system integrates ML algorithms to predict supply chain disruptions, helping companies like Siemens optimise inventory management and reduce operational costs (SAP 2023). AI-driven chatbots, such as Microsoft's Dynamics 365 Copilot, allow employees to interact with ERP systems using natural language, simplifying data retrieval and report generation (Microsoft 2024b).

In CRM systems, AI personalises customer interactions and improves sales forecasting. Salesforce Einstein AI, for example, employs NLP and predictive analytics to analyse customer behaviour, recommend sales actions, and automate lead prioritisation (Salesforce 2024). Companies like Coca-Cola use AI-powered CRM tools to enhance customer engagement, optimise marketing campaigns, and predict purchasing trends (Doering 2024).

The MES has also benefited from AI integration, particularly in predictive maintenance and quality control. AI-powered anomaly detection in MES systems helps manufacturers like General Motors identify potential equipment failures before they occur, reducing downtime and maintenance costs (Motors 2023). Siemens MindSphere, an industrial IoT platform, combines AI with MES data to optimise production processes and improve efficiency (Siemens 2024).

Multimodal AI models further enhance BIS by integrating text, images, and numerical data for intelligent decision-making. For example, Amazon's AI-driven demand forecasting system combines historical sales data, weather patterns, and consumer behaviour analysis to predict demand fluctuations and optimise supply chain logistics (Li 2024; Tsimpoukelli et al. 2021)

Despite these advancements, challenges remain, including data privacy concerns, system interoperability, and the need for customisation to fit specific business requirements (Yao et al. 2024). As AI continues to evolve, its role in BIS will expand, driving greater efficiency, automation, and strategic insights across industries.

2.2 Systematic Literature Review

This section outlines the methodology used for the SLR, detailing the systematic steps taken to achieve the thesis objectives. It begins by presenting the research questions and defining the search strategy, including the identification of sources, terms, and procedures for study selection and data extraction. Section 2.2.2 presents the findings for each research question, accompanied by a detailed analysis. Finally, Section 2.2.3 discusses the implications of the results within the broader research context.

2.2.1 Methodology

According to Mark and Roberts (2006) and Grant and Booth (2009), an SLR is an established methodology for identifying, evaluating, and synthesizing all relevant research concerning a particular question, topic, or phenomenon of interest. Despite the growing body of literature on artificial intelligence, there remains limited comprehensive review focused on their application in business information systems. Recognizing this gap, a systematic review of existing research on the use of AI technology in certain BIS, was conducted.

This review adheres to the methodology described by Moher et al. (2009), which includes the development of a detailed review protocol that specifies research questions and search strategies. Following the execution of the systematic review, the collected data was critically analysed and synthesized to provide insight and discuss the results in depth.

Research questions

In this SLR, it was defined the main research question as follows: "What is the current state-of-the-art in artificial intelligence techniques for integrating and modelling data from

BIS?" To address this question, the most recent literature is reviewed according to four sub-questions (see Table 2.2).

In the first question, the primary AI technologies employed for the integration and modelling of data from diverse BIS are analysed. This helps to identify the specific tools and techniques, such as Machine Learning, Deep Learning, or other AI-driven methods, that are most frequently utilised in this domain. Additionally, the diversity of these technologies is explored to understand their adaptability and effectiveness in handling various BIS contexts.

In the second question, the focus shifts to examining the most commonly used methods in NLP for interpreting and generating natural language responses within enterprise systems. This includes an assessment of different NLP approaches, such as transformers, large language models, or rule-based methods, and their applications in real-world business scenarios, such as customer service automation or intelligent data query systems.

The third question investigates how Machine Learning and Deep Learning techniques have been applied to the integration and modelling of data from BIS. The goal is to understand the extent to which these techniques enable seamless data consolidation, predictive analytics, and decision-making support in business environments. Moreover, it assesses the relevance of these approaches to specific BIS challenges, such as data heterogeneity and scalability.

In the fourth question, the primary challenges in developing intelligent virtual assistants capable of consolidating business data from multiple sources are analysed. This includes identifying obstacles such as data interoperability, maintaining data privacy, ensuring system reliability, and the scalability of virtual assistants in complex business environments. Additionally, the benefits of overcoming these challenges, such as improved user experience and enhanced decision-making, are explored to guide future research and development efforts in this area.

	Research question
RQ1	What artificial intelligence technologies have been used for the integration and modelling of data from different BIS?
RQ2	What are the most commonly used methods in the application of NLP for interpreting and responding in natural language in enterprise systems?
RQ3	How have ML and DL techniques been used for the integration and modelling of data from BIS?
RQ4	What are the main challenges faced in developing intelligent virtual assistants that consolidate business data from multiple sources?

Table 2.2: Research questions

Definition of search strategy

The search conducted in this review focused on works available in the literature regarding current scientific knowledge about intelligent virtual assistants and chatbots applied to business information retrieval and decision support. The search strategy is divided into three steps: definition of search sources, selection of search terms, and the process of study selection and data extraction.

Definition of search sources

The initial step of the search strategy involved identifying and defining the sources to be used for conducting the systematic literature review. For this study, the search was carried out across three electronic databases (refer to Table 2.3) and included works published as journal articles, book chapters, conference proceedings, and books.

Identifier	Database	URL
ED1	IEEE Xplore	`https://ieeexplore.ieee.org/`
ED2	Web Of Science	`https://webofscience.com/`
ED3	Science Direct	`https://www.sciencedirect.com/`

Table 2.3: Electronic databases

Definition of search terms

The second step of the search strategy involved creating a set of search strings aligned with the defined research questions. Three distinct search strings were formulated, each corresponding to a specific area of study. These search strings are detailed in Table 2.4.

Scope	String
Virtual Assistant	(Virtual Assistant OR Chatbot OR Chat OR Bot) **AND**
AI Technologies	(Natural Language Processing OR NLP OR Large Language Models OR LLM OR Deep Learning OR DL OR Artificial Intelligence OR AI OR Model Context Protocol OR MCP) **AND**
Business Information Systems	(Enterprise Resource Planning OR ERP OR Customer Relationship Management OR CRM OR Manufactoring Execution Systems OR MES)

Table 2.4: Search strings

Study selection and data extraction process

The study selection and data extraction process involved establishing specific criteria for inclusion and exclusion to refine the results and focus on the most relevant studies that could address our SLR research questions. Clear conditions for both inclusion and exclusion were established, as detailed in Tables 2.5 and 2.6. Studies meeting at least one inclusion criterion were retained, whereas those matching any exclusion criterion were omitted.

	Inclusion Criteria
IC1	The source focuses on virtual assistants, chatbots, or AI systems designed for interaction with ERP or similar business systems.
IC2	The source studies utilise NLP, ML, or Data Integration techniques specifically for business management applications.
IC3	Peer-reviewed journal articles, conference papers, or reputable industry white papers.

Table 2.5: Inclusion Criteria

	Exclusion Criteria
EC1	Exclude studies that focus on AI applications outside business contexts (e.g., healthcare or education)
EC2	Non-peer-reviewed articles, blogs, or studies lacking rigorous methodology.
EC3	Sources not written in English.
EC4	Sources published before 2020.

Table 2.6: Exclusion Criteria

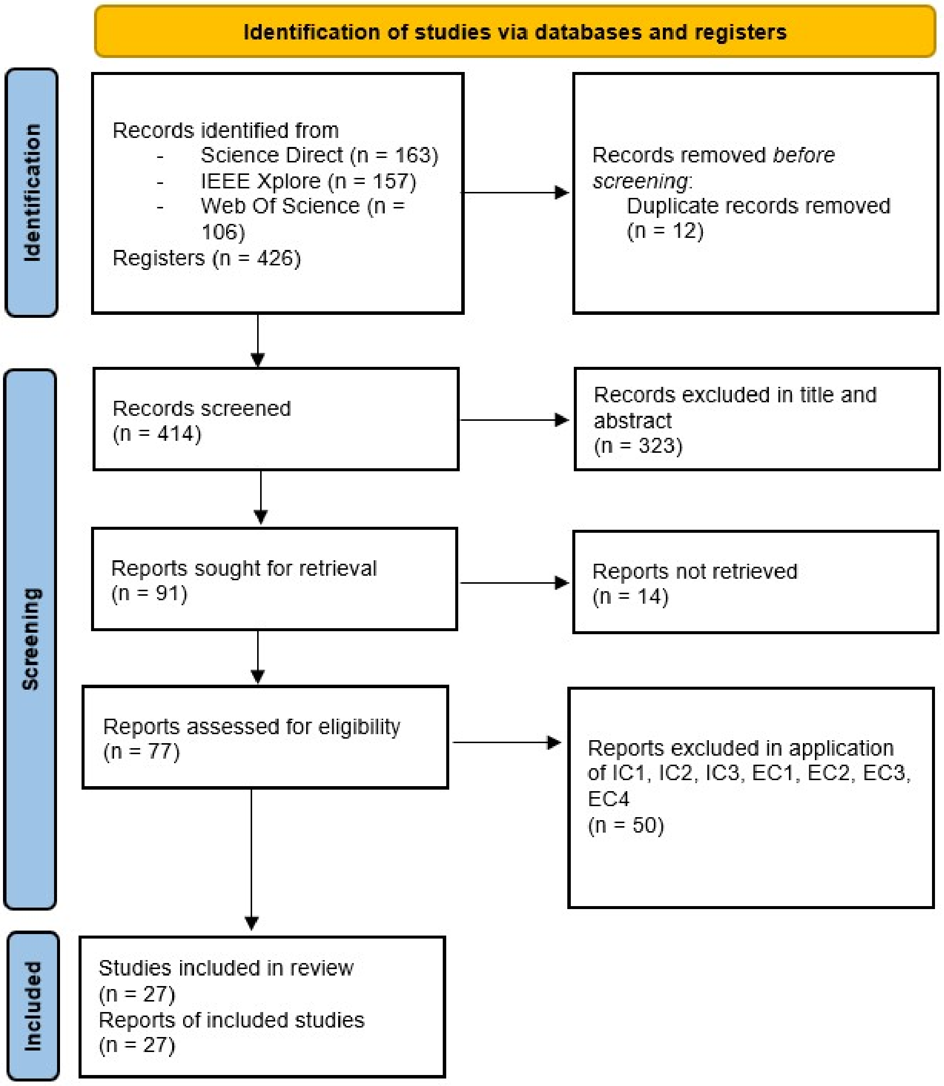


Figure 2.1: PRISMA flow diagram

Following this, the PRISMA Page et al. (2021) guidelines were considered as a foundation for the research methodology. However, since the research was carried out by a single researcher, the full implementation of PRISMA was not feasible. Instead, the research process was adapted and structured based on the principles of PRISMA, encompassing three stages: Identification, Screening, and Inclusion. As shown in Figure 2.1, during the Identification phase, 426 studies were gathered from various databases: Science Direct (163), IEEE Xplore

(157), Web of Science (106). After removing 12 duplicate records, 414 unique studies were screened. In the Screening phase, 323 studies were excluded based on title and abstract review, leaving 91 studies for further assessment. In the Eligibility phase, a full-text analysis was done on the remaining records according to the Inclusion and Exclusion criteria that were defned. As a result of this evaluation, 50 studies were deemed ineligible, leaving a total of 27 studies that met the criteria and were included in the review.

Table 2.7 presents a structured overview of the studies examined in this research, categorised by their relevance to each research question. This categorisation enables a clearer understanding of the current research landscape and identifies gaps for future exploration.

Study	**RQ1**	**RQ2**	**RQ3**	**RQ4**
W. Jin et al. (2025)	X		X	X
Ehtesham et al. (2025)	X		X	X
Przegalinska et al. (2025)	X	X	X	
Yue et al. (2024)		X	X	X
O'Leary (2024a)		X	X	X
O'Leary (2024b)	X		X	
Kern et al. (2024)	X	X	X	
Gupta et al. (2024)		X		
Todoric et al. (2024)	X	X	X	
S. Sharma et al. (2024)				X
Garcia et al. (2024)	X			X
Ozay et al. (2023)	X			
Garg and D. Sharma (2023)			X	
R. Sharma et al. (2023)	X	X	X	X
Ahaneku et al. (2023)	X		X	
Asan and Tecim (2023)	X	X	X	X
Unni et al. (2023)			X	
Pereira et al. (2022)	X	X	X	X
Mantravadi et al. (2022)	X			
Krishnareddy et al. (2022)	X	X	X	X
Sai et al. (2022)		X	X	
Hüsson, Holland, Fathi, et al. (2021)				X
Youn and S. V. Jin (2021)				X
Doshi (2021)		X		
Makarius et al. (2020)	X	X		X
Huang (2020)			X	
Hüsson, Holland, and Arteaga Sanchez (2020)	X	X	X	X

Table 2.7: Studies by research question

2.2.2 Results

This section presents the results of the SLR, detailing how each research question was addressed based on the information provided in the selected studies.

Research question 1: What artificial intelligence technologies have been used for the integration and modelling of data from different BIS?

Artificial intelligence technologies have profoundly reshaped how data is integrated and modeled within BIS. Among the most impactful innovations are Generative AI and Retrieval-Augmented Generation (RAG). These technologies leverage LLMs like ChatGPT and Bard, enabling enterprises to harness the power of AI to access and synthesize structured and unstructured data. By combining generative capabilities with retrieval systems, organisations achieve more accurate, contextually relevant responses to complex queries, enhancing decision-making processes (O'Leary 2024b). Furthermore, Human-AI collaboration tools have emerged as pivotal in enabling seamless data integration. These tools optimise the interaction between human expertise and machine intelligence, fostering efficient task management and robust decision support across diverse organisational landscapes (Przegalinska et al. 2025). Recent studies further emphasize the use of Modular Chatbot Frameworks in production environments. These frameworks, which utilise modern NLP and machine learning technologies, enable efficient data retrieval and integration tailored to specific organisational needs. By implementing modular designs, enterprises can adapt AI systems to diverse use cases, improving internal processes and operational efficiency (Kern et al. 2024). Additionally, the concept of Socio-Technical Systems highlights the alignment of AI technologies with social structures within organisations, enhancing both technical performance and collaborative efficacy (Makarius et al. 2020). Pereira et al. (2022) propose the use of Voice Activity Detection (VAD), Automatic Speech Recognition (ASR), and Ontological Components. These technologies facilitate the semantic mapping of ERP system functionalities, enabling seamless integration and standardisation of data across multiple systems. The study from Mantravadi et al. (2022) demonstrated, the integration of data in Business Information Systems, specifically MES and ERP systems within smart factories, relies heavily on Industrial Internet of Things (IIoT) platforms and standardised frameworks like ISA 95. Key technologies include interoperability-focused architectures, modular designs, and open-source systems such as Odoo. The study also emphasizes the use of cryptographic protocols for secure data transmission, ensuring robust and reliable integration. As Garcia et al. (2024) demonstrate, LLMs offer transformative potential in integrating and modelling data across business systems, particularly in manufacturing environments. Their framework highlights the use of LLMs to consolidate and process diverse datasets from sources like ERP systems, IoT devices, and operational logs. The application of embeddings and vector databases plays a pivotal role in enabling efficient data retrieval by mapping text data into numerical representations that allow similarity-based searches. This ensures seamless and relevant information retrieval, significantly improving decision-making processes. The study also introduces Factory Integrated Logic and Language Interface System (FILLIS), an LLMs-based virtual assistant designed to interact with manufacturing systems, including ERP systems. FILLIS automates tasks such as querying technical documentation and synthesising data for operational guidance. By integrating long-term memory (e.g., historical datasets like equipment manuals) and short-term memory (e.g., conversational history), FILLIS ensures contextually accurate and user-specific responses. Furthermore, the system demonstrates the potential for integrating external agents, such as mathematical models, to extend functionality. This framework reinforces the role of advanced AI technologies, particularly LLMs, in facilitating data integration, cross-departmental collaboration, and operational efficiency within BIS-related systems. These insights add depth to the understanding of AI technologies that enable seamless data modelling and integration, underscoring their impact on enhancing business intelligence and decision-making capabilities. The article by Ozay et al. (2023) discusses the use of Latent Dirichlet Allocation (LDA) as a method for topic modelling, enabling the identification of hidden patterns and themes within large datasets. This technique enhances the integration of customer data within CRM systems by providing actionable insights into customer behavior. Krishnareddy et al. (2022) explores the use of the AI-based Fuzzy Inference System (AI-FIS), which integrates fuzzy logic and machine learning to model and analyse customer data in CRM systems. AI-FIS facilitates data preprocessing through normalization and tokenization, ensuring consistent and relevant data integration. By analysing patterns and customer sentiment, AI-FIS enables a comprehensive approach to customer data modelling, improving decision-making and data-driven strategies in CRM, which aligns with ERP-like systems. The article introduces an intelligent ERP (i-ERP) system that integrates AI techniques, ML, and NLP to automate and optimise ERP functionalities. The i-ERP model leverages technologies such as cloud computing for managing extensive datasets and predictive analytics to enhance decision-making. The proposed system uses advanced classification algorithms, such as Sequential Minimal Optimization (SMO), to transform unstructured user support requests into actionable solutions, improving ERP data integration and process modelling (Asan and Tecim 2023). Advanced AI technologies, such as autonomous learning analytics and customised deep learning algorithms, have been deployed to integrate and model data from BIS, enabling the identification of latent patterns, novel correlations, and actionable insights from unstructured data, such as enterprise chat logs, email transcripts, and CRM databases (R. Sharma et al. 2023). Ahaneku et al. (2023) discusses the use of Google Dialogflow for NLP-based chatbot interactions and Bidirectional Encoder Representations from Transformers (BERT)-based deep learning models for log data analysis in manufacturing plants. Additionally, it introduces a self-learning module that continuously scans and updates error detection models using MES databases, which could be relevant to BIS integration in industrial settings. Todoric et al. (2024) article discusses AI-powered CRM features, such as Salesforce Einstein, predictive analytics, and machine learning-driven customer segmentation, highlighting how AI enhances data integration and business intelligence within CRM. Hüsson, Holland, and Arteaga Sanchez (2020) explores AI-powered speech-interaction features in ERP, such as search functions, data retrieval, and process automation, demonstrating how AI enhances ERP usability and business intelligence. Recent work has highlighted the MCP as a promising approach for orchestrating heterogeneous AI workflows across structured and unstructured data sources. W. Jin et al. (2025) demonstrated how an MCP-driven framework could coordinate multiple multimodal LLMs and OCR components to transform unstructured clinical data into structured records, showing the potential for non-intrusive integration of disparate systems. Similarly, Ehtesham et al. (2025) applied MCP to healthcare data via the Fast Healthcare Interoperability Resources (FHIR) standard, enabling declarative and dynamic retrieval of electronic records and natural language summarisation tailored to different user personas. Although situated in medical contexts, these studies illustrate the transferability of MCP-based integration to enterprise environments such as ERP and CRM, where the consolidation of diverse business datasets is a persistent challenge. The findings emphasise MCP's role as a unifying layer that supports interoperability, modularity, and scalability in BIS integration.

Research question 2: What are the most commonly used methods in the application of NLP for interpreting and responding in natural language in enterprise systems?

NLP methods have been instrumental in transforming enterprise systems, particularly in their ability to interpret and respond in natural language. Prompt Engineering is a cornerstone of this transformation. By employing techniques like chain-of-thought reasoning, few-shot learning, and domain-specific contextual embedding, enterprises ensure that LLMs generate precise and relevant responses. This practice not only enhances interpretative accuracy but also aligns AI outputs with organisational objectives (O'Leary 2024a; Przegalinska et al. 2025). Another significant application is in AI-driven chatbots and virtual assistants, which are increasingly used to streamline internal communication and customer interactions. These tools utilise sentiment analysis and contextual understanding to provide meaningful and timely responses, significantly improving operational efficiency (Yue et al. 2024). Furthermore, studies highlight the adaptability of LLMs like GPT and BERT within modular chatbot frameworks. These frameworks employ fine-tuned NLP techniques to handle specialized organisational tasks, such as retrieving critical data or automating repetitive queries. The integration of such frameworks within enterprise systems not only enhances precision but also allows for continuous improvement based on user interactions and feedback (Kern et al. 2024). Pereira et al. (2022) study demonstrates the integration of open-source speech-to-text engines like Mozilla DeepSpeech and Vosk to interpret user commands. These NLP tools, combined with ontological databases, allow the conversion of natural language inputs into actionable system instructions. AI-FIS leverages NLP techniques, such as sentiment analysis and pattern recognition, to interpret customer reviews and interactions. These methods enhance the ability of CRM systems to understand customer emotions and preferences, providing personalised responses. (Doshi 2021) also details NLP techniques such as intent recognition, sentiment analysis, and conversational AI, demonstrating how these methods enhance chatbot performance in CRM. Tokenization and normalization preprocess textual data, enabling accurate classification and sentiment analysis, making NLP integral to improving natural language interactions within CRM systems (Krishnareddy et al. 2022). NLP techniques are central to the study, enabling the system to interpret user-entered support requests written in natural language. These requests undergo preprocessing steps, including tokenization, stemming, and stop word removal, to prepare the data for classification. The system uses NLP-based models to parse and classify text into predefined categories, ensuring accurate understanding and effective responses. This implementation facilitates a conversational interface, allowing ERP users to communicate naturally with the support system (Asan and Tecim 2023). AI-driven knowledge repositories and dynamic NLP-enhanced digital assistants are widely used in enterprise systems to provide seamless natural language interpretation and context-aware responses, offering real-time query resolution and intelligent recommendations by leveraging domain-specific ontologies and adaptive learning frameworks (R. Sharma et al. 2023). Sai et al. (2022) highlights how NLP is used in chatbots for supplier communication and procurement inquiries within ERP systems. These chatbots employ tokenization, lemmatization, Term Frequency-Inverse Document Frequency (TF-IDF) vectorization, and cosine similarity to analyse procurement-related text and provide contextualized responses. Todoric et al. (2024) highlights NLP applications, particularly Einstein GPT, for automated customer interaction, natural language understanding, and chatbot functionalities, demonstrating AI's role in improving enterprise communication. Hüsson, Holland, and Arteaga Sanchez (2020) discusses NLP-based voice commands, speech synthesis, and explanation modes for automating ERP workflows and assisting users with system navigation.

Research question 3: How have Machine Learning and Deep Learning techniques been used for the integration and modelling of data from BIS?

ML and DL techniques have become integral to the integration and modelling of data in BIS. These technologies are central to Data Augmentation and Knowledge Management, where advanced algorithms process vast datasets to identify patterns, extract insights, and enhance decision-making capabilities (O'Leary 2024b). In addition, the deployment of Enterprise Knowledge Bases underpinned by deep learning models facilitates robust data integration. Techniques like retrieval-augmented LLMs employ sophisticated neural architectures to synthesize large datasets, thereby driving business intelligence and operational efficiency (Przegalinska et al. 2025). Deep learning approaches are also highlighted in modular chatbot frameworks, which leverage LLMs for automating complex organisational processes. These systems utilise predictive modelling to streamline data integration tasks, ensuring that relevant insights are readily accessible for decision-makers. The emphasis on continuous learning within these frameworks ensures that AI systems remain adaptive and effective in dynamic business environments (Kern et al. 2024). Pereira et al. (2022) adds that the use of Recurrent Neural Networks (RNNs) in ASR systems underscores the role of deep learning in transforming unstructured audio data into structured text, which can then be incorporated into BIS workflows for enhanced decision-making and process optimisation. Krishnareddy et al. (2022) highlights the use of machine learning algorithms within AI-FIS for clustering and classifying customer data. Fuzzy logic is combined with machine learning to handle uncertainty in customer behavior data, enabling precise segmentation and predictive modelling. These techniques enhance CRM systems' ability to integrate and process large datasets effectively, improving customer retention and satisfaction through targeted interventions. ML and DL techniques are employed to classify and model support requests within the ERP system. Algorithms such as SMO, Random Forest, and Long Short-Term Memory (LSTM) are utilised to classify and predict user queries effectively. The model uses TF-IDF for data weighting, enabling the extraction of meaningful patterns from text data. These approaches enhance the system's ability to handle large-scale data integration and provide predictive insights for ERP functionalities (Asan and Tecim 2023). Specific DL techniques, such as contextual semantic analysis and unsupervised neural embeddings, have been applied to large-scale datasets across BIS to uncover unknown relationships between operational variables, allowing businesses to create predictive models and derive domain-specific insights (R. Sharma et al. 2023). In Garg and D. Sharma (2023) study, DL models are applied to analyse historical test data and software logs to identify patterns and generate predictive insights, ensuring comprehensive test coverage for ERP systems. These models dynamically prioritize high-risk areas during testing, optimizing resource allocation and improving the robustness of BIS integration. Huang (2020) article partially addresses this question by discussing ML models that analyse historical sales engagement data to estimate deal success rates and customer response likelihood. These models enhance data-driven decision-making by identifying sales patterns, optimizing deal-closing strategies, and improving customer interaction outcomes. Sai et al. (2022) article partially addresses this question. It mentions the use of Analytic Hierarchy Process (AHP), a structured decision-making approach that applies machine learning principles to rank suppliers based on multiple weighted criteria, such as reliability, cost, and service quality. In Ahaneku et al. (2023) study, BERT-based neural network is used to match user queries with historical log data for troubleshooting and the self-learning module continuously improves the AI model by extracting patterns from MES logs and generating synthetic training data, allowing adaptation to evolving error patterns in industrial systems. Unni et al. (2023) highlights fraud detection as a key application, using predictive modelling and neural networks to identify anomalies in transactions. Unsupervised learning helps detect suspicious patterns, while reinforcement learning adapts fraud prevention strategies. Todoric et al. (2024) study identifies Salesforce Einstein, an AI-powered analytics engine, as a key ML tool used for automating data analysis, improving lead scoring, and generating predictive insights for sales and customer management. Hüsson, Holland, and Arteaga Sanchez (2020) AI-powered speech-interaction chatbot uses machine learning for data clustering, voice-enabled analytics, and predictive assistance, enhancing ERP data integration and decision support. Both W. Jin et al. (2025) and Ehtesham et al. (2025) highlight how deep learning models and LLMs can be operationalised for data integration and modelling within structured systems. W. Jin et al. (2025) employed transformer-based multimodal LLMs (e.g., Qwen2.5, DeepSeek) combined with OCR to ensure accurate modelling of domain-specific records, while validating semantic quality with BLEU and ROUGE metrics. Ehtesham et al. (2025) advanced this further by embedding agentic workflows where LLMs dynamically reason over structured FHIR resources retrieved via MCP. Their approach illustrates how ML/DL techniques can manage both structured and unstructured data simultaneously, providing robust data pipelines that ensure semantic consistency, explainability, and adaptability. These contributions demonstrate that DL-driven orchestration and agentic integration can extend beyond healthcare to BIS domains, supporting predictive analytics, automated modelling, and context-aware decision-making.

Research question 4: What are the main challenges faced in developing intelligent virtual assistants that consolidate business data from multiple sources?

Developing intelligent virtual assistants capable of consolidating business data from multiple sources entails several challenges. One of the foremost is Data Privacy and Security. Ensuring the confidentiality of sensitive organisational data is critical, especially when leveraging public LLMs like ChatGPT. Enterprise-specific LLMs address this issue by embedding advanced privacy controls and isolating data environments (O'Leary 2024a). Another significant challenge is Bias and Contextual Understanding, where virtual assistants must navigate biases inherent in training datasets while ensuring accurate contextual comprehension. This is crucial for delivering equitable and relevant responses (Yue et al. 2024). Finally, Prompt Engineering Limitations pose hurdles, as crafting effective prompts to guide virtual assistants requires a nuanced understanding of both the AI system and organisational needs (O'Leary 2024a). Makarius et al. (2020) also points to the difficulty of integrating virtual assistants with existing enterprise systems. Modular chatbot frameworks, while offering adaptability, face challenges in achieving full interoperability with diverse data environments. Additionally, fostering trust in AI systems among employees is critical; sociotechnical frameworks suggest that addressing the relational and structural dimensions of AI adoption can enhance collaboration and integration. Pereira et al. (2022) research highlights the complexities of achieving noise-robust speech recognition, ensuring semantic consistency, and integrating heterogeneous data sources. These challenges emphasize the need for robust system architecture and user-centric design. Youn and S. V. Jin (2021) explained, developing intelligent virtual assistants for CRM involves fostering trust through balanced design and addressing ideological diversity (e.g., technopians vs. luddites). Building parasocial interactions and scaling emotional engagement while maintaining competence and clear AI identity are critical challenges. Role framing (e.g., "assistant" vs. "friend") further influences user trust and satisfaction, requiring adaptive conversational strategies. Garcia et al. (2024) highlight several challenges in developing intelligent virtual assistants like their LLMs-based system, FILLIS.

Key issues include dependency on external agents for tasks such as mathematical computations, which LLMs cannot perform reliably. Effective management of memory—balancing long-term datasets and short-term conversational context—is also complex, especially when integrating real-time data. Additionally, fine-tuning LLMs for domain-specific tasks requires significant computational resources, making prompt engineering a preferred but limited alternative. These challenges emphasise the need for robust frameworks and tools to optimise virtual assistants in consolidating and processing business data. Krishnareddy et al. (2022) study identifies challenges in developing AI-driven CRM systems, such as meeting customer expectations for chatbots. It emphasizes the need for improved decision-making algorithms, such as AI-FIS, to enhance chatbot performance in understanding and responding to customer queries. Challenges include ensuring the system's adaptability to varied customer inputs and maintaining high accuracy in sentiment analysis and response generation. Asan and Tecim (2023) study identifies challenges in developing AI-driven ERP support systems, including the difficulty of transforming unstructured data into structured solutions. It highlights the need for robust NLP and ML models to handle diverse and complex user requests. Security concerns are also noted, emphasizing the importance of integrating user verification mechanisms and maintaining data integrity within ERP operations. Additionally, the scalability of AI-driven systems to adapt to dynamic organisational needs presents another key challenge. Challenges in developing intelligent virtual assistants include integrating legacy systems with modern multi-modal AI architectures, ensuring secure blockchain-based data exchanges for sensitive business transactions, and addressing the complexity of autonomous decision-making capabilities that rely on heterogeneous data sources (R. Sharma et al. 2023). Intelligent virtual assistants in business face challenges such as limited context-awareness, data integration complexities, scalability issues, and user expertise dependency, which affect their adaptability, effectiveness, and seamless deployment across diverse business environments (Hüsson, Holland, Fathi, et al. 2021). S. Sharma et al. (2024) stated that small and medium enterprises face several challenges in adopting AI-based chatbots, including perceived complexity, as many struggle with chatbot setup and integration. High implementation costs further limit access for smaller firms, while a lack of internal IT expertise makes chatbot management difficult. Additionally, limited vendor support forces smalland medium enterprises to rely on external expertise, and customer resistance remains a barrier, as many users still prefer human interactions over AI-driven systems. Hüsson, Holland, and Arteaga Sanchez (2020) also identifies challenges such as user acceptance, accuracy of speech recognition, and privacy concerns in integrating chatbots into ERP systems, partially addressing this question. W. Jin et al. (2025) noted difficulties with semantic consistency, error propagation, and computational overhead when orchestrating multiple specialised models under MCP. Ehtesham et al. (2025) similarly identified issues around context management, reproducibility, interoperability, and explainability, particularly when scaling assistants across heterogeneous datasets. Both works stress the importance of transparent agentic workflows and traceability of outputs to maintain user trust. These challenges, while framed in clinical scenarios, parallel those faced in ERP and CRM assistants, where assistants must query multiple databases, maintain contextual continuity across interactions, and provide verifiable, business-critical insights. Together, these findings suggest that robust orchestration frameworks, careful model selection, and governance mechanisms are essential to overcoming the technical and organisational barriers in deploying virtual assistants for BIS.

2.2.3 Discussion

This section critically examines the findings for each research question, providing a comprehensive synthesis of how AI technologies are being applied in BIS. The discussion situates these findings within the broader context of AI research and enterprise applications, while also identifying key challenges and future research directions.

RQ1: What artificial intelligence technologies have been used for the integration and modelling of data from different BIS?

The first research question addressed the artificial intelligence technologies employed for the integration and modelling of data from different BIS. The findings suggest that recent advancements in AI, particularly in RAG and transformer-based language models, have significantly enhanced data consolidation across enterprise systems. Unlike conventional methods that rely on structured database queries, these AI models enable context-aware information retrieval and synthesis, making them particularly effective for multi-source data integration. The deployment of modular chatbot frameworks further facilitates enterprise-wide AI adoption by allowing organisations to tailor AI-driven interactions for specific business functions. The integration of LLMs within enterprise systems has also contributed to improved data modelling, particularly through the use of vector databases, which enhance semantic data retrieval and enable sophisticated pattern recognition. Additionally, the implementation of ontology-driven AI systems, such as FILLIS, has improved the interpretability of AI-generated insights by aligning them with structured enterprise data models.

Despite these advancements, several challenges persist. One of the most significant issues is the lack of standardisation in data formats, which complicates the seamless integration of AI into different BIS. Furthermore, the reliance on cloud-based AI models introduces concerns related to data privacy and security, as businesses increasingly deal with proprietary and sensitive information. Computational efficiency remains a limiting factor, as large-scale AI models require significant processing power, which can hinder real-time adaptability in enterprise environments. Future research should focus on the development of hybrid AI architectures that combine neurosymbolic reasoning with retrieval-augmented models to enhance decision-making processes. Additionally, investigating on-premises LLM deployments could mitigate security risks while ensuring compliance with data protection regulations. Advancements in cross-BIS interoperability, particularly in the integration of ERP, CRM, and MES systems, will be crucial for further improving AI-driven business intelligence.

AI technologies used for data integration and modelling in BIS have traditionally relied on approaches such as ETL pipelines, data warehouses, and more recently vector databases for semantic search. These methods focus on unifying heterogeneous datasets across ERP, CRM, and MES systems to enable decision support and business intelligence. However, recent research demonstrates that orchestration protocols such as the MCP can provide a more modular and scalable alternative to hard-coded integrations. W. Jin et al. (2025) showed how MCP can coordinate multimodal LLMs and OCR components to transform unstructured data into structured records, ensuring semantic accuracy while maintaining interoperability. Similarly, Ehtesham et al. (2025) proposed an MCP-FHIR framework where LLMs dynamically query structured repositories and generate explainable natural language summaries tailored to user roles. These contributions indicate a clear evolution: integration technologies in BIS are moving from static pipelines toward dynamic, declarative, and agent-based architectures. Such advances can enhance not only interoperability across ERP/CRM but also transparency and adaptability, two aspects crucial for enterprise-level trust in AI.

RQ2: What are the most commonly used methods in the application of NLP for interpreting and responding in natural language in enterprise systems?

The second research question explored the most commonly used methods in the application of NLP for interpreting and responding in natural language within enterprise systems. The findings indicate that transformer-based NLP models, including BERT, GPT-4, and T5, have fundamentally altered how enterprises process natural language queries. These models have significantly improved contextual understanding by leveraging self-attention mechanisms to capture complex dependencies within business-specific texts. In addition to model advancements, the use of prompt engineering techniques, such as chain-of-thought reasoning and few-shot learning, has enhanced the adaptability of LLMs to domain-specific applications without requiring extensive retraining. This approach enables virtual assistants to interpret industry-specific jargon more effectively, thereby reducing inaccuracies in complex business queries.

Another major development in NLP for enterprise applications is the integration of speech-based AI systems, which facilitate voice-enabled interactions with business platforms. ASR combined with retrieval-augmented conversational agents has enabled organisations to automate internal communication, customer service interactions, and supply chain inquiries. However, several challenges remain in achieving fully optimised NLP-based enterprise systems. One of the primary concerns is the ability of these models to retain long-term conversational context, as most current implementations lack persistent memory, making multi-turn interactions prone to inconsistencies. Additionally, domain-specific adaptation remains a challenge, as fine-tuning general-purpose LLMs for highly specialised business contexts is often resource-intensive and requires ongoing refinement. Bias mitigation in NLP-driven business systems is another critical issue, particularly in applications related to hiring and financial decision-making, where fairness and transparency are essential. Future advancements should focus on memory-augmented NLP systems that retain contextual awareness across sessions, self-supervised enterprise NLP models trained on internal corporate data, and explainable AI techniques that enhance transparency in AI-driven decision-making.

RQ3: How have ML and DL techniques been used for the integration and modelling of data from BIS?

The third research question examined how ML and DL techniques have been applied to the integration and modelling of data from BIS. The findings reveal that predictive analytics has become a cornerstone of ERP, enabling businesses to make data-driven decisions through the application of ML algorithms such as Gradient Boosting, Random Forest, and XGBoost. In CRM and ERP systems, these techniques have been employed for sales forecasting, inventory optimisation, and customer segmentation, demonstrating their effectiveness in enhancing decision-support processes. In addition to traditional ML methods, the use of DL models, particularly LSTM networks and transformer-based time series forecasting, has provided significant improvements in predicting demand fluctuations and managing supply chain logistics.

Another area where ML and DL techniques have shown substantial impact is anomaly detection, particularly in financial systems and manufacturing execution environments. Unsupervised learning models, such as autoencoders, have been successfully deployed to identify fraudulent transactions, detect operational inefficiencies, and enhance risk management strategies. However, the widespread adoption of these techniques is constrained by several factors. The availability of high-quality labelled training data remains a bottleneck, as enterprise data is often fragmented across multiple systems, making it difficult to train accurate predictive models. Computational inefficiencies also pose a challenge, as real-time AI deployment in high-volume business environments requires substantial processing power. Moreover, the lack of explainability in deep learning-driven decision support systems raises concerns about the transparency of AI-generated recommendations. Addressing these limitations will require further research into automated machine learning solutions that enable low-code AI development for businesses, as well as the advancement of explainable AI frameworks that ensure interpretability in enterprise analytics. The integration of edge AI solutions could further decentralise ML-driven decision-making, allowing businesses to process data locally and reduce dependency on cloud-based computing resources.

ML and DL techniques have long been employed in BIS for predictive analytics, clustering, anomaly detection, and decision optimisation. Classical approaches such as random forests, neural networks, and LSTMs remain central to forecasting and segmentation tasks, while unsupervised models support anomaly detection and fraud prevention. Beyond these established methods, recent advances demonstrate the value of deep learning orchestration for data integration and modelling. W. Jin et al. (2025) employed multimodal transformers (e.g., Qwen2.5, DeepSeek) alongside OCR to automatically convert unstructured documents into structured, machine-readable records, validating outputs with BLEU and ROUGE metrics. This approach is particularly relevant to BIS contexts where contracts, invoices, and communications must be integrated into ERP/CRM systems. Building on this, Ehtesham et al. (2025) illustrated how MCP-based workflows enable LLMs to reason over structured data repositories (FHIR) while providing traceable and explainable outputs. Together, these findings expand the scope of ML/DL in BIS from isolated models to multimodal, agentic pipelines capable of ensuring semantic consistency, explainability, and governance. This direction strengthens predictive modelling, enhances fraud detection, and supports real-time decision making across enterprise systems.

RQ4: What are the main challenges faced in developing intelligent virtual assistants that consolidate business data from multiple sources?

The final research question investigated the main challenges associated with developing intelligent virtual assistants that consolidate business data from multiple sources. The findings indicate that while virtual assistants have significantly improved enterprise data accessibility, they face several critical limitations. One of the foremost challenges is ensuring data privacy and security, particularly when using cloud-based AI models that require external processing. The integration of federated learning and homomorphic encryption has been proposed as a potential solution to mitigate security risks by enabling decentralised AI training without exposing sensitive data to third-party servers. Another major challenge is cross-BIS integration, as many enterprise systems lack standardised Application Programming Interfaces (APIs), making it difficult for AI-driven virtual assistants to retrieve real-time data from multiple sources. The adoption of ontology-based data mapping has improved interoperability to some extent, but further advancements are needed to achieve seamless data exchange across heterogeneous BIS.

In addition to technical challenges, user adoption remains a significant barrier to the widespread implementation of intelligent virtual assistants. Many employees exhibit scepticism toward AI-generated recommendations, particularly in high-stakes business environments where decision accuracy is paramount. This resistance is often attributed to the lack of transparency in AI-driven decision-making and the perceived risk of automation replacing human expertise. To address these concerns, AI developers must prioritise the design of hybrid human-AI collaboration models, where virtual assistants provide recommendations that are subject to human validation. Furthermore, the implementation of adaptive, self-learning virtual assistants that continuously refine their responses based on user interactions could enhance trust and usability. Future research should focus on developing domain-specific, privacy-preserving LLMs for enterprise applications, multi-agent AI systems that integrate rule-based and DL-based reasoning, and intelligent virtual assistants with improved context retention and dynamic knowledge graph updates.

The development of intelligent virtual assistants that consolidate data from multiple business sources continues to face both technical and organisational challenges. Issues such as user trust, transparency, and the mitigation of hallucinations remain central concerns, alongside data privacy and compliance with regulations like GDPR and the AI Act. Recent work provides additional insights into these challenges. W. Jin et al. (2025) highlighted the risks of semantic inconsistency and error propagation when orchestrating multiple specialised models under MCP, emphasising the computational overhead of such integrations. Ehtesham et al. (2025) identified challenges around context management, interoperability, and explainability in their MCP-FHIR framework, stressing the importance of reproducible, modular workflows. Both studies underline the need for careful governance, robust orchestration mechanisms, and transparent auditing to ensure that assistants provide accurate and verifiable outputs. In the BIS context, these findings confirm that while AI-driven assistants are capable of enhancing productivity, their successful deployment depends on overcoming limitations in accuracy, scalability, and trust. Addressing these challenges will be essential for building assistants that are not only intelligent but also reliable, compliant, and enterprise-ready.

Chapter 3

Methodology and Data

Following the state-of-the-art review, this chapter consolidates the data foundation and the methodological framework of the study. Section 3.1 details the exploration, selection, and characteristics of the dataset used, including the rationale for adopting an enterprise-style warehouse that reflects realistic business operations. Building on that foundation, Section 3.2 presents the methodological design of the intelligent virtual assistant, covering the system architecture, middleware, safeguards, and model integration. This arrangement emphasises the direct dependency of the methods on the chosen data, maintaining continuity from problem context to implementation.

3.1 Data retrieval

The success of any AI model is fundamentally dependent on the quality, structure, and representativeness of the data used during its development. In the context of this research, data serves as the backbone for training, validating, and evaluating the intelligent virtual assistant designed to facilitate business data access. Given the sensitive nature of business information, as discussed in Section 1.5, working with real-world enterprise data presents significant challenges related to confidentiality, compliance, and ethical considerations.

One of the primary challenges in obtaining real-world business data is its highly sensitive and proprietary nature. Many organisations maintain strict access controls to prevent data breaches and ensure compliance with legal and regulatory requirements, such as the GDPR. Moreover, business data often contains personally identifiable information and financial records, requiring careful handling to prevent ethical and legal complications.

To overcome these constraints, this study explores a set of datasets, evaluating their characteristics, strengths, and limitations to determine the most suitable one for this research. Various datasets will be compared based on factors such as data completeness, relevance to business applications, scalability, and compliance with ethical guidelines. The selection process ensures that the chosen dataset aligns with the research objectives while maintaining applicability to real-world enterprise scenarios.

The evaluation of multiple datasets is crucial for ensuring that the AI model is trained on data that closely mirrors real-world conditions while upholding ethical standards. By systematically comparing different datasets, this study aims to mitigate biases, enhance model performance, and ensure that the intelligent virtual assistant effectively processes and retrieves business data. This chapter outlines the data retrieval process, describing the datasets under consideration, their structure, key features, and the methodological approach taken to extract, preprocess, and utilise this information for AI model training. Additionally, it discusses the challenges and considerations involved in data selection, highlighting the importance of data integrity and relevance in developing robust AI-driven solutions for business applications.

The process of selecting an appropriate dataset is a critical step in ensuring the effectiveness of the AI model. Given the wide range of publicly available and synthetic datasets, it is essential to carefully evaluate each option to determine which best aligns with the requirements of this research. This section will present the methodology used to explore, compare, and ultimately select the dataset that meets the needs of this study. The evaluation process considers multiple factors, including the dataset's structure, comprehensiveness, scalability, and ethical considerations, to ensure that the AI model is trained on high-quality, representative data.

3.1.1 Criteria for Dataset Selection

To ensure that the selected dataset effectively supports the research objectives, the evaluation process considers relevance to business applications, data completeness and structure, scalability and compatibility with AI models, and ethical and compliance considerations. This four criteria are summarised concisely in Table 3.1 and serve as the normative basis for the comparisons that follow.

Criterion	**Focus in this study**
Relevance to business applications	Alignment with real-world enterprise environments and the business data access tasks defined for this study.
Data completeness and structure	Presence of diverse, well-organised records and relationships representative of enterprise systems.
Scalability and compatibility with AI models	Capacity to be processed effectively by machine learning and natural language processing techniques and to integrate with the execution environment.
Ethical and compliance considerations	Strict adherence to GDPR; avoidance of personally identifiable or otherwise sensitive data to prevent regulatory violations.

Table 3.1: Summary of dataset selection criteria.

3.1.2 Comparison of Available Datasets

Microsoft provides a range of sample databases designed to illustrate different business scenarios, which makes them valuable resources for AI research and enterprise applications. These datasets are widely used in database management training, business intelligence solutions, and AI-driven data analysis. Their structured nature, industry relevance, and accessibility make them ideal for testing AI models in a controlled environment. The three datasets selected for this study—Adventure Works, Wide World Importers, and Northwind and Pubs—each offer unique advantages depending on the complexity of business processes they represent. By comparing these datasets, this research ensures that the selected data model aligns with the objectives of developing an intelligent virtual assistant capable of efficient business data retrieval and analysis (Microsoft 2024c).

In this section, three SQL-based data models from Microsoft will be compared: Adventure Works, Wide World Importers, and Northwind and Pubs. Each dataset is designed to represent different business scenarios and provide structured data that can be utilised for

AI-driven applications. The evaluation will consider their suitability for this study based on their content, complexity, and scalability.

Adventure Works is a comprehensive dataset designed to simulate a large-scale enterprise environment, specifically modeled after Adventure Works Cycles, a fictitious multinational manufacturing company. This company produces and sells metal and composite bicycles across North American, European, and Asian commercial markets. Microsoft provides different versions of this dataset, spanning from 2008 to 2022, with three main types of sample databases tailored for different workloads, Online Transaction Processing (OLTP), Data Warehouse (DW) and Lightweight (LT). The OLTP database is designed for handling typical business transactions, making it ideal for AI applications focused on real-time data retrieval and business intelligence. The DW version is optimised for analytical workloads, supporting AI-driven insights generation and predictive modelling. The LT version provides a simplified version of the OLTP database, facilitating rapid prototyping and proof-of-concept implementations. These variations allow Adventure Works to serve as a flexible dataset, adaptable to different AI-driven business intelligence applications, reinforcing its suitability for the intelligent virtual assistant developed in this research (Microsoft 2024a).

For example, the AdventureWorks2012 OLTP database, provided by Microsoft for SQL Server, consists of 70 tables and includes four years of historical sales data (2005–2008) (Ordonez et al. 2014). This extensive dataset, allows for the simulation of various enterprise operations, providing valuable insights into sales trends, supply chain management, and human resources, making it highly suitable for complex business intelligence applications. It encompasses data related to sales, purchasing, human resources, and product management, making it highly suitable for complex business intelligence applications. This dataset's structure follows a normalised schema that reflects real-world relationships between entities such as customers, employees, products, and transactions. The richness of its dataset makes it an optimal choice for AI models requiring extensive and interconnected data for training and evaluation. However, its complexity also presents a challenge, as pre-processing steps such as data transformation and feature selection may be required to ensure compatibility with machine learning algorithms. Furthermore, due to its size and relational intricacy, computational efficiency may become a limiting factor, necessitating optimisations in query execution and indexing.

Wide World Importers is a comprehensive sample database provided by Microsoft, designed to simulate the operations of a mid-sized wholesale distribution business. This dataset models a realistic company engaged in importing and distributing goods, encompassing a variety of business processes, including purchasing, sales, stock management, and financial transactions. Unlike older sample databases, it is structured to support both OLTP and Online Analytical Processing (OLAP) workloads, making it highly versatile for AI-driven business intelligence applications.

The database is particularly relevant for AI applications focused on supply chain optimisation and predictive analytics. Wide World Importers features an advanced schema that incorporates real-world business complexities such as inventory tracking, order fulfillment, and supplier management. The OLTP version of the database is designed for transaction-heavy environments, ideal for testing AI-driven automation in order processing and real-time inventory management. The OLAP version, on the other hand, is optimised for business intelligence and analytical workloads, supporting historical trend analysis, sales forecasting, and decision-making models.

Additionally, Wide World Importers includes extensive metadata and sample data that mimic real-world business transactions, allowing AI models to be trained on structured and time-sensitive data. This makes it an excellent resource for AI-driven applications requiring dynamic decision-making capabilities, such as automated stock replenishment, customer demand forecasting, and supplier performance evaluation. The dataset also incorporates time-sensitive transaction history, enabling the development of models that can detect seasonal sales patterns and supply chain inefficiencies. Its modern design and adherence to best practices in data modelling allow for seamless integration with machine learning and forecasting models, which can enhance demand prediction, inventory optimisation, and automated decision-making.

While Wide World Importers presents a highly structured and modernised database, it also introduces some challenges. As it is relatively new compared to datasets like Adventure Works and Northwind, it may require additional integration efforts to be compatible with legacy business applications that rely on older database structures. Additionally, its complexity in handling both OLTP and OLAP workloads means that AI-driven applications utilising this dataset may require significant preprocessing and optimisation for performance efficiency. Despite these challenges, its well-documented structure, real-world business scenarios, and robust transactional and analytical data make it a strong candidate for AI-driven business intelligence applications. Its versatility and comprehensive nature further reinforce its potential as a valuable dataset for training AI models focused on business process automation and intelligent data retrieval (Microsoft 2023).

Northwind and Pubs are legacy datasets provided by Microsoft, historically used for database training and small-scale business modelling. Northwind models a small retail and distribution business, with structured data on customer orders, supplier relationships, and shipping logistics, making it useful for understanding fundamental business transactions. Pubs, by contrast, focuses on the publishing industry, featuring data on authors, books, publishers, and sales, providing a simplified framework for inventory and sales tracking.

Both datasets are characterized by their straightforward schema, making them easily accessible for AI prototyping and rapid model iteration. Their lightweight nature minimizes computational requirements, allowing for quick experimentation. However, their primary limitation lies in their outdated business structures and relatively narrow scope, which restrict their applicability to modern, large-scale enterprise scenarios. They lack the advanced transactional depth, multi-departmental interactions, and real-world business complexity required for training AI models capable of handling diverse enterprise operations. While they serve as useful introductory datasets for testing basic AI functionalities, their constraints make them less suitable for sophisticated AI-driven applications in contemporary business environments (Microsoft 2022).

3.1.3 Discussion for dataset selection

Considering the criteria outlined in Section 3.1.1, Adventure Works is the most suitable dataset for this research as it provides a comprehensive and realistic representation of enterprise-scale business processes. Unlike Wide World Importers, which is tailored for wholesale operations, and Northwind and Pubs, which focus on smaller business environments, Adventure Works encompasses a wide range of corporate activities, including sales, purchasing, human resources, and product management. This extensive scope enables the simulation of complex business workflows that an AI-driven virtual assistant must handle effectively.

The dataset's highly normalised schema mirrors real-world corporate data structures, making it ideal for training AI models that process structured queries and retrieve relevant business insights. Its detailed granularity, incorporating transactional records, employee data, and supply chain logistics, supports machine learning models in generating accurate business intelligence applications. Although its complexity requires preprocessing and computational optimisation, these efforts are justified by its rich and adaptable structure.

Adventure Works' relational data model facilitates robust query formulation, a key requirement for natural language processing (NLP)-based AI assistants. Its broad coverage across multiple business functions ensures that AI-driven systems can provide insights across different departments. Compared to Wide World Importers, which primarily focuses on distribution, and Northwind and Pubs, which are more simplistic training datasets, Adventure Works offers a depth and diversity that better aligns with modern business challenges.

While preprocessing and managing its large-scale dataset may require additional computational resources, the benefits of its structured design outweigh these challenges. The dataset's robustness enables sophisticated AI-driven business intelligence applications, supporting efficient and scalable data retrieval processes.

In line with the evaluation dimensions established in Section 3.1.1, Adventure Works achieves the best overall balance of business relevance, structural completeness, scalability, and compliance. Given the objectives of this research, which involve developing an AI assistant capable of handling diverse business queries, Adventure Works provides the optimal foundation for accurate and efficient experimentation.

Having established the dataset and its properties, the focus now turns to the methodology that operationalises this data within the Intelligent Virtual Assistant (IVA). The subsequent sections describe the architectural choices, governance mechanisms, and runtime behaviours that enable secure, on-premises, and resource-efficient access to business information.

3.2 Methodology

The effective integration of AI into BIS requires a robust, scalable, and computationally efficient architecture. As outlined in the previous chapter, the dataset utilised in this research originates from structured business repositories such as ERP systems, CRM platforms, and MES. Given the critical importance of data privacy described in Section 1.5, the methods and tools selected for this pipeline must operate entirely on-premises while maintaining a balance between accuracy, latency, and computational efficiency.

This chapter presents the methodology adopted to design and implement the proposed IVA. Unlike classical machine learning approaches that rely on offline preprocessing, model training, and evaluation pipelines, this work emphasises a lightweight, privacy-preserving, and on-premises solution for querying enterprise data. The chapter first frames the problem, then introduces the system architecture, before detailing the core component—the MCP server—its data preparation mechanisms, its integration with a large language model, and the interaction flow with the end user.

3.2.1 Problem Framing

The core problem addressed in this dissertation is how to enable secure, efficient, and user-friendly access to enterprise data through natural language queries. Business organisations increasingly rely on intelligent assistants embedded into platforms such as Microsoft Copilot, Salesforce Einstein, or SAP Joule (Microsoft 2024b; Salesforce 2024; SAP 2023). While these tools demonstrate powerful capabilities, they are primarily cloud-based, transmitting sensitive enterprise data to third-party servers for processing. Such architectures raise concerns under the GDPR and the forthcoming European Union AI Act, where the handling of personally identifiable information, trade secrets, and financial data must remain within strict organisational control.

The systematic review in Chapter 2 showed that most state-of-the-art solutions for IVAs in business contexts rely on either fine-tuning large language models with corporate datasets or outsourcing inference to external providers. Both approaches present significant obstacles in environments where data sovereignty and privacy preservation are non-negotiable. For example, training requires extensive preprocessing and storage of corporate data, while cloud inference raises risks of unauthorised access or regulatory violations.

In contrast, the methodological approach in this dissertation is to eliminate both training and external inference. Instead, the IVA is designed to query data dynamically through a middleware layer, avoiding permanent storage of sensitive content outside the enterprise database. This middleware is realised through the MCP, which provides a standardised mechanism for exposing structured resources and tools to an LLM client.

To clarify the distinction, consider the comparison in Table 3.2:

Feature	**Cloud-based IVA**	**on-premises MCP-based IVA**
Data storage	Vendor servers	Enterprise SQL Server
Model training	Fine-tuning / API	Runtime SQL execution
Privacy & compliance	GDPR/AI Act risks	Local, compliant
Deployment cost	Subscription, usage-based	Fixed local hardware
Customisation	Vendor-limited	Full MCP control
Latency & control	Internet-dependent	Local, low-latency

Table 3.2: Comparison between cloud-based and on-premises MCP-based IVA solutions.

This dissertation therefore frames the problem not as "how to train the best IVA model", but rather as "how to design an architecture that enables secure, lightweight, and useful interaction with enterprise data without violating privacy constraints."

The pivot from offline preprocessing and model training to runtime query-oriented preparation reflects both a technical and governance-driven choice. By ensuring that data never leaves the corporate SQL Server and that the IVA only accesses a filtered, semantically meaningful subset of attributes, the methodology directly addresses the tension between AI-enabled insight generation and corporate data protection.

3.2.2 System Architecture

The proposed IVA is implemented as a modular, on-premises system that integrates a Microsoft SQL Server database, a MCP server, a locally LLM, and a lightweight client interface. The architecture was deliberately designed to balance three requirements: (i) privacy preservation, (ii) computational efficiency, and (iii) user accessibility.

At a high level, the workflow follows a query–response cycle: a user poses a natural language question, the LLM interprets it and interacts with the MCP server, the MCP server validates and executes the corresponding SQL query, and the results are returned, preprocessed, and transformed into a natural language response. This section details each component and explains how they interact.

The overall architecture is illustrated in Figure 3.1. It consists of four main components: the Client, the Local LLM, the MCP Server, and the SQL Server database. These components form a closed-loop interaction pipeline that enables natural language queries to be translated into safe SQL statements and converted into business insights.

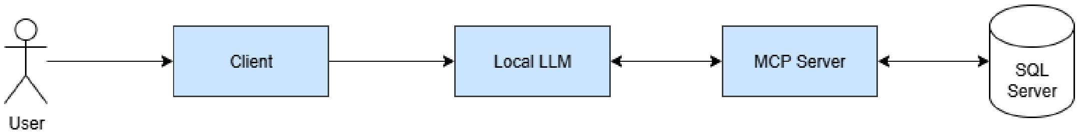


Figure 3.1: System architecture of the proposed IVA, showing the interaction between Client, Local LLM, MCP Server, and SQL Server.

The client is a lightweight, terminal-based application through which the user interacts with the system. It accepts natural language queries and displays both textual and visual responses. Despite its simplicity, the client plays a crucial role in bridging human communication with the technical backend. It is deliberately minimal to highlight that the contribution of this dissertation lies not in interface design, but in the secure and efficient middleware pipeline.

The reasoning component of the IVA is a locally deployed Qwen3 30B model, hosted via the Ollama runtime (Ollama 2025a). The local deployment avoids external API calls and guarantees that no sensitive corporate data leaves the organisational environment. The LLM interprets natural language questions, consults the MCP server for schema information, generates SQL queries, and formulates human-readable answers. By operating locally, it ensures compliance with GDPR and AI Act requirements while demonstrating the feasibility of large-model inference on controlled infrastructure.

The MCP server, implemented in Python using the FastMCP framework, acts as the middleware that exposes database resources and tools in a controlled, structured manner. Its responsibilities include:

- Establishing and maintaining a persistent ODBC connection to SQL Server.
- Exposing table schemas and metadata as resources.
- Providing tools for safe SQL query execution and data visualisation (bar charts, scatter plots, line plots, histograms, heatmaps).
- Enforcing read-only access by rejecting unsafe operations such as `INSERT`, `UPDATE`, `DELETE`, or `ALTER`.
- Performing runtime data preparation, including column filtering, prioritisation, and output normalisation.

The MCP server ensures that only meaningful and secure subsets of the database are exposed to the assistant.

The system relies on Microsoft's AdventureWorksDW2022 dataset, hosted in SQL Server, as its data source. The database simulates a realistic data warehouse environment with fact and dimension tables for sales, products, customers, and geography. It provides a rich testbed for evaluating the IVA's capacity to answer business-oriented queries, such as identifying revenue trends or comparing regional performance. Hosting SQL Server locally maintains full control of the data and prevents unauthorised external access.

Together, these components operate in the following sequence:

1. The user submits a query through the client.
2. The Local LLM analyses the request and, if necessary, queries schema information from the MCP server.
3. The MCP server validates and executes the SQL query against SQL Server, applying runtime data preparation.
4. The MCP server returns filtered, JSON-serialised results (and optionally visualisations).
5. The Local LLM integrates the structured data into a natural language answer, which is displayed by the client.

This architecture ensures that the IVA provides meaningful insights without requiring model training or exposing sensitive data outside the enterprise environment.

3.2.3 MCP Server

The MCP server constitutes the central middleware of the proposed IVA architecture. It provides a structured and standardised interface between the SQL Server database and the local LLM, ensuring that all interactions remain safe, privacy-preserving, and semantically meaningful. Unlike a direct SQL connector, the MCP server performs multiple functions: it abstracts the complexity of the database schema, enforces read-only security, applies runtime data preparation, and provides higher-level analytical tools.

The implementation was developed in Python using the FastMCP framework, which simplifies the construction of MCP-compliant servers (Prefect 2025). Through this framework, the server exposes three categories of resources to the client: resources, tools, and prompts.

Resources provide read-only, structured context that the LLM can consult before composing SQL. They are designed to be cheap to fetch, stable over a session, and cacheable on the client to reduce latency.

The `database_overview_resource` returns compact metadata about the connected database (name, server instance, number of tables) and the active filtering rules (e.g., exclusion of IDs, GUIDs, binaries, audit columns). Used once per session to prime the LLM's mental model of what is available and which classes of fields will be hidden at query time. Caching is recommended; the content changes rarely unless the connection switches database or filtering policy.

The `table_schema_resource` returns the effective schema after runtime filtering and column capping. Each column includes at least its name and type; some implementations also return nullability and basic notes (e.g., "excluded columns present but hidden"). This is the default schema view that the LLM should rely on for composing SQL, because it mirrors what the execution tool will allow. Fetched immediately before drafting a new query or when errors indicate a name/type mismatch. Avoids requesting multiple tables at once unless needed, to control token usage. For a full retrieval, the `table_schema_full_resource` returns the complete, raw schema as in SQL Server, including columns that will be filtered in normal operation. Reserved for debugging or expert inspection when a query fails repeatedly due to ambiguous joins or legacy fields. This view should not be used to compose production queries; it is provided to explain why certain columns are unavailable and to guide schema curation decisions.

Filtered schemas should always be preferred, since they ensure alignment with the constraints enforced at execution time. Only the schemas that are strictly necessary should be requested, because long schema dumps increase context size and add latency. When an error occurs, the correct practice is to re-request the relevant filtered schema rather than attempting to guess column names. In multi-turn conversations, small and frequently referenced schemas, such as DimProduct or DimGeography, can be cached to reduce the number of round-trips and improve responsiveness.

Tools are the action interfaces the LLM can invoke. They are side-effect-free by design and implement strict read-only behaviour.

The `execute_sql_query` tool validates and runs a safe `SELECT` statement, returning results in JSON with columns, rows, and row counts. It blocks modification or definition commands, disallows stored procedures, and rejects `SELECT *` in favour of explicit projections. Returned values are normalised so that decimals, datetimes, or other complex types appear in predictable formats.

Using Matplotlib and Seaborn, the server can render SQL results as bar charts, line graphs, scatter plots, histograms, or heatmaps and return them as static images. These outputs help summarise comparisons, trends, or distributions in a compact form (Hunter 2007; Waskom 2021). Some examples of usability are:

- Bar charts (e.g., revenue per product category).
- Line plots (e.g., sales trends over time).
- Scatter plots (e.g., correlation between discount rate and sales amount).
- Histograms (e.g., distribution of order quantities).
- Heatmaps (e.g., revenue matrix by region and product category).

These visualisation tools help the IVA provide richer responses that go beyond text and summarise patterns in the data visually.

Tools such as `test_database_connection` or `get_table_schema_full` are provided for troubleshooting connectivity or investigating filtering rules.

Effective use of tools follows a clear loop: inspect a schema, draft a query, execute it, and, if useful, request a chart. Queries should project explicit columns to keep outputs concise and secure, and grouped aggregates are generally easier to interpret than raw record dumps. When errors occur, the LLM should display them and use the feedback to refine the query rather than retrying unchanged.

Prompts are scaffolds provided by the server to guide the LLM toward safe, type-aware SQL that aligns with the schema. They capture common patterns and reduce trial and error.

- Basic data summary: Produces SQL that computes row counts and descriptive statistics for numerical fields. Used to obtain a brief summary of a single table, often with optional grouping by a categorical field.
- Top records analysis: Generates ordered selections with `SELECT TOP N` and a clear sorting criterion, supporting ranking questions such as "top products by sales."
- Group and count analysis: Produces frequency tables and aggregates over categorical attributes, useful for questions involving categories, regions, or channels.
- Time-series analysis: Casts date/time fields into standard formats and aggregates results by day, month, or year to analyse trends over time.

Prompts are most effective when treated as templates that are adapted with the exact column names from filtered schemas. Explicit projections and `WHERE` clauses should be added early to limit scope. If a query fails, the safer approach is to return to the template, re-insert validated columns, and rerun it. Because prompts already encode the governance rules of the system—explicit projection, read-only assumptions, and safe casting—they increase consistency and reduce invalid SQL. In a CPU-only environment with low-temperature decoding, they also help the model produce deterministic, correct SQL.

Security Enforcement

A critical requirement of the MCP server is that it enforces **strict read-only behaviour**. Since the assistant interacts directly with a production-style SQL Server database, any risk of unauthorised modifications must be eliminated. The security model implemented in the MCP server focuses on query validation, deny-listing of dangerous commands, and controlled connection management.

Query validation. All incoming SQL statements are first stripped and converted to lowercase to check that they begin with the keyword `SELECT`. Queries that do not satisfy this requirement are immediately rejected. In addition, the server applies a deny-list of prohibited keywords, including `INSERT`, `UPDATE`, `DELETE`, `DROP`, `ALTER`, `TRUNCATE`, `EXEC`, and procedure prefixes such as `sp_` or `xp_`. This prevents the assistant from executing any destructive or administrative operations. Listing 3.1 illustrates the Python function used for this validation, which enforces column-level restrictions and prevents potentially harmful operations.

```python
def is_select_query(query: str) -> bool:
    query = query.strip().lower()
    if not query.startswith("select "):
        return False
    blocked_keywords = [
        " insert ", " update ", " delete ", " drop ", " alter ",
        " create ", " truncate ", " exec ", " execute ",
        " sp_", " xp_", " grant ", " revoke ", " deny "
    ]
    for keyword in blocked_keywords:
        if keyword in f" {query} ":
            return False
    return True
```

Listing 3.1: Query validation function from the MCP server implementation.

Read-only guarantees. By enforcing this validation layer, the MCP server ensures that the assistant can only retrieve data and never perform operations that modify the database. This guarantees that database integrity and confidentiality are preserved during all interactions.

Connection management. The server maintains a global singleton connection to SQL Server and periodically validates its state by issuing a trivial `SELECT 1` command. If the connection is lost, the MCP server attempts to reconnect securely with trusted parameters. This prevents insecure or dangling connections, which could otherwise create vulnerabilities.

Transparency. Whenever schema information is provided, the server explicitly communicates whether any filtering has been applied. This ensures that both users and the assistant remain aware of the restricted access policy and that excluded attributes cannot be requested.

Together, these mechanisms guarantee that the MCP server operates as a secure, read-only gateway to SQL Server, providing a controlled environment for the IVA to explore enterprise data without risking corruption or unauthorised access.

To illustrate the MCP server in operation, consider the following query scenario:

1. **User:** "What is the total sales amount per product category?"
2. **LLM action:** The language model requests the schema of *FactInternetSales* and *DimProductCategory* via MCP resources.
3. **MCP server:** The server returns filtered schemas, excluding irrelevant fields such as identifiers and binary objects.
4. **LLM:** Based on the schema, the model generates the SQL statement as shown in Listing 3.2:

```sql
SELECT pc.EnglishProductCategoryName,
       SUM(fs.SalesAmount) AS TotalSales
FROM FactInternetSales fs
INNER JOIN DimProductCategory pc
       ON fs.ProductCategoryKey = pc.ProductCategoryKey
GROUP BY pc.EnglishProductCategoryName
ORDER BY TotalSales DESC;
```

Listing 3.2: Example SQL query generated via the MCP server.

5. **MCP execution:** The server validates the query, executes it on SQL Server, applies runtime preprocessing (e.g., filtering, normalisation), and returns results.
6. **LLM final answer:** The model presents a ranked list of product categories with their total sales, optionally accompanied by a bar chart generated by the visualisation tools.

3.2.4 Runtime Data Preparation in the MCP Server

The IVA does not rely on conventional offline preprocessing pipelines such as data cleaning, feature engineering, or dataset splitting for training. Instead, all preparation occurs dynamically within the MCP server whenever schema information is requested or a query is executed. This runtime data preparation ensures that only semantically meaningful, business-relevant attributes are exposed to the LLM, while technical identifiers and low-value fields are excluded. The approach is designed to reduce cognitive load for the model, minimise hallucinations, and prevent leakage of sensitive information.

When retrieving schema information from SQL Server, the MCP server applies a clear sequence of filtering rules so that only business-relevant attributes are exposed to the assistant:

- Excluded by name patterns: Columns ending in `id`, containing `guid`, `uuid`, `photo`, `image`, `logo`, `hash`, or similar terms are removed.
- Excluded by data types: Columns of type `varbinary`, `image`, `uniqueidentifier`, `xml`, and spatial datatypes (`geometry`, `geography`) are excluded.
- Excluded by explicit deny-list: Known technical attributes such as `rowguid`, `modifieddate`, and `thumbnailphoto` are removed.

For illustration, the *DimProduct* table originally contains columns such as `ProductPhoto` (binary image) and `rowguid`. After filtering, only textual, numerical, and categorical attributes such as `EnglishProductName`, `StandardCost`, and `Color` are retained.

To avoid overwhelming the assistant with wide tables, the MCP server limits the number of exposed columns per table (default: 15). When a table exceeds this threshold, the server prioritises fields by data type:

1. Textual attributes (`varchar`, `nvarchar`, `char`)
2. Numerical attributes (`int`, `decimal`, `float`, `money`)
3. Temporal attributes (`date`, `datetime`)
4. Boolean attributes (`bit`)
5. Low-priority attributes (`text`, `ntext`)

This prioritisation ensures that business-relevant columns are preserved, while less useful ones are truncated.

Instead of using `SELECT *`, which could expose all columns including excluded ones, the MCP server generates queries with explicit column lists. This ensures that only filtered attributes are returned to the IVA, reinforcing both data minimisation and query clarity. Listing 3.3 demonstrates how the model generates a SELECT statement that retrieves only the relevant attributes instead of the entire table.

```sql
SELECT [EnglishProductName], [StandardCost], [Color]
FROM [DimProduct];
```

Listing 3.3: Example SQL projection.

For statistical and aggregative queries, the MCP server adjusts SQL statements at runtime so that numerical and temporal values are in forms that work reliably with standard SQL functions. Numerical values are cast to `FLOAT` to ensure compatibility with aggregation operators such as `AVG`, `MIN`, and `MAX`, while temporal fields are normalised to the `DATE` type so that grouping by day, month, or year behaves consistently. As shown in Listing 3.4, the system normalises a natural language request with a date condition into SQL syntax that applies explicit conversion.

```sql
SELECT CONVERT(DATE, [OrderDate]) AS OrderDay,
       COUNT(*) AS Orders
FROM [FactInternetSales]
GROUP BY CONVERT(DATE, [OrderDate]);
```

Listing 3.4: Example SQL query applying date conversion and aggregation in the MCP server.

Finally, the MCP server serialises query results into JSON-compatible types so that the assistant receives predictable, uniform structures regardless of the original SQL types. Where necessary, non-primitive values—such as decimals or timestamps—are converted into string representations; this prevents type mismatches during consumption and keeps the interface between tools stable. Listing 3.5 illustrates the transformation of a tabular SQL output into a JSON-normalised format.

```json
{
  "columns": ["EnglishProductName", "TotalSales"],
  "rows": [
    ["Mountain-200 Black, 46", 125000.50],
    ["Road-350 Red, 52", 98000.00]
  ],
  "row_count": 2
}
```

Listing 3.5: Example of JSON-normalised output returned by the MCP server.

To demonstrate the practical effect of runtime data preparation, this subsection provides a before-and-after illustration of schema filtering as applied to the *DimProduct* table in the AdventureWorksDW2022 database. The original schema includes a variety of technical attributes, such as identifiers, globally unique identifiers (GUIDs), binary image fields, and audit columns, which are not relevant for business-level analysis. After applying the MCP server's preprocessing rules, these fields are excluded, leaving only semantically meaningful attributes such as product names, costs, and descriptive categories. Figure 3.2 comparison highlights how the server reduces schema complexity while preserving analytical value, ensuring that the IVA operates on a concise and business-relevant representation of the database.

(a) Original schema (DimProduct)

Column	Type
ProductKey	int
ProductAlternateKey	nvarchar
ProductSubcategoryKey	int
WeightUnitMeasureCode	nchar
SizeUnitMeasureCode	nchar
EnglishProductName	nvarchar
StandardCost	money
FinishedGoodsFlag	bit
Color	nvarchar
SafetyStockLevel	smallint
ReorderPoint	smallint
ListPrice	money
Size	nvarchar
SizeRange	nvarchar
Weight	float
DaysToManufacture	int
ProductLine	nchar
DealerPrice	money
Class	nchar
Style	nchar
ModelName	nvarchar
LargePhoto	varbinary
EnglishDescription	nvarchar
StartDate	datetime
EndDate	datetime
Status	nvarchar

(b) Filtered schema exposed by MCP

Column	Type
EnglishProductName	nvarchar
ProductSubcategoryKey	int
Color	nvarchar
Size	nvarchar
Weight	float
ProductLine	nchar
Class	nchar
Style	nchar
StandardCost	money
ListPrice	money
DealerPrice	money
ModelName	nvarchar
StartDate	datetime
EndDate	datetime
Status	nvarchar

Note: Identifiers, multilingual descriptions, binary images, and technical stock-control fields are excluded. The IVA only sees attributes relevant for analysis and reporting.

Figure 3.2: Before-and-after illustration of runtime schema filtering applied by the MCP server to *DimProduct*. The original schema (a) lists all database columns; the filtered view (b) exposes only business-relevant fields.

3.2.5 LLM Integration

This section explains how a locally hosted large language model (LLM) is integrated to interpret user queries, orchestrate MCP tools, and produce natural language answers grounded in SQL Server results. All inference is executed on-premises to preserve privacy and regulatory compliance; no enterprise data is transmitted to external APIs.

The LLM is hosted using the Ollama runtime, which ensures operational consistency across models while retaining full local control. Ollama handles model loading, quantisation, and memory allocation, making it possible to test different LLMs in the same environment without modifying integration code.

All experiments were conducted on an on-premises workstation equipped with an AMD Ryzen 5 7600X (6-core) processor, 32 GB RAM, and an NVIDIA GeForce RTX 4070 GPU. This configuration enables CUDA-accelerated inference in Ollama while preserving full local control and privacy.

Aspect	Workstation specification	Methodological implication
CPU (inference)	AMD Ryzen 5 7600X (6 cores)	Strong single-thread performance benefits token-by-token decoding; maintain conservative, deterministic decoding for stable SQL generation.
RAM (capacity)	32.0 GB	Sufficient for mid-size models and schema metadata caching; use quantised variants for larger contexts or multiple concurrent sessions.
GPU	NVIDIA GeForce RTX 4070	Enables CUDA-accelerated inference (e.g., FP16/BF16), reducing latency and permitting larger models/contexts than CPU-only execution.
Storage/IO	Local	No external data transfer; aligns with privacy-preserving, on-premises constraints.

Table 3.3: Hardware profile and implications for on-premises LLM integration and runtime configuration.

Models were served in the format most appropriate for the available accelerator: FP16/BF16 on the GPU and 4–5 bit quantised variants when a lighter footprint was required. Decoding remained conservative (temperature 0.0–0.2) to encourage deterministic SQL, and context windows typically ranged from 4,096 to 8,192 tokens depending on query complexity. Larger contexts supported multi-table reasoning, whereas smaller contexts offered faster responses for simple analytical tasks.

Caching schema information retrieved through MCP reduced redundant context usage and sustained responsiveness in multi-turn scenarios. This configuration, combined with MCP's runtime data preparation mentioned in Section 3.2.4, enabled stable performance on the updated workstation while maintaining compliance with privacy and governance requirements.

Several open-weight models were evaluated to balance reasoning ability, SQL reliability, and on-premises feasibility:

- `granite 3.dense:8b`: strong general reasoning but higher latency and memory pressure on commodity hardware; joins and grouping were competent yet sometimes verbose.
- `granite 3.dense:2b`: efficient footprint and quick responses, but weaker performance on schema-aware SQL (multi-table joins, grouped aggregates).
- `llama3.2:3b`: good linguistic fluency; intermittent instability in constructing correct join paths and grouping expressions across the AdventureWorks star schema.
- `deepseek-r1:8b`: solid contextual reasoning; fluctuating accuracy on schema-specific selection, grouping, and consistent column naming.
- `qwen3-30b` (*selected*): most consistent balance overall; better adherence to MCP prompts and tools, a higher success rate on multi-join, `GROUP BY`, and filter conditions, and acceptable latency when quantised.

Evaluation criteria emphasised syntactic correctness of SQL, semantic correctness relative to exposed schemas (i.e., correct column and table names after filtering), robustness on typical business queries (aggregations over fact tables joined to relevant dimensions), and resource usage. Very small models underperformed on schema-aware reasoning, whereas much larger models were over-demanding for on-premises constraints. `qwen3-30b` was ultimately selected for integration due to its superior SQL reliability, reasoning depth, and stability when controlled with strict decoding parameters.

No fine-tuning was performed on any of the models. This was an intentional methodological choice to avoid issues of data retention, compliance, and additional training overhead. Instead, the approach relies on MCP's runtime filtering and projection mechanisms to simplify the schema space, combined with prompt engineering to keep the model grounded in authoritative data.

The IVA employs an "MCP-first retrieval" pattern rather than a vector-database retrieval-augmented generation (RAG) pipeline. In this approach, the MCP server itself functions as the retrieval and shaping layer.

The interaction sequence is as follows:

1. The user poses a business question.
2. The LLM queries MCP resources for relevant schemas (for example, filtered columns of *FactInternetSales* and *DimProduct* or *DimGeography*).
3. Using MCP-provided templates, the LLM drafts a query with explicit projections, avoiding `SELECT *` and matching the filtered schema exposed by MCP.
4. The MCP server validates the query (ensuring it is read-only), executes it against SQL Server, applies runtime filtering and type conversions, and returns the results as structured JSON.
5. The LLM integrates the structured output into a concise natural language response, optionally supported by charts generated by the MCP's visualisation tools.

System and developer prompts instruct the model to inspect schemas before proposing SQL, to prefer the server's analysis templates (such as basic data summary, group-and-count, and time-series) as safe starting points, to use explicit column lists respecting the filtered view (IDs and binaries excluded, capped columns per table), and to iterate when errors arise by re-requesting schema details and correcting the query rather than guessing.

The server exposes ready-to-adapt prompt templates (for example, `basic_data_summary`, `group_and_count_analysis`, `time_series_analysis`, `top_records_analysis`) that already include type-safe casting (numerics to `FLOAT` for `AVG/MIN/MAX`; datetimes to `DATE` for grouping). By grounding SQL in these templates, the LLM inherits the server's safety and data-minimisation conventions with fewer opportunities to drift.

Returned JSON structures (`columns`, `rows`, `row_count`) are consumed directly by the LLM. When visual interpretation is beneficial, the response references a plot generated by the MCP visualisation tools (bar, line, scatter, histogram, or heatmap). The textual answer highlights key figures and trends, and the chart provides a compact visual summary.

Despite the improvements gained by combining Qwen3-30B with MCP scaffolding, limitations remain. Typical errors include use of excluded columns, casing mismatches, or under-specified joins. These are handled by returning explicit database error messages, which the model uses to request schema information again and regenerate a corrected query. Ambiguity in schema attributes is reduced through filtered and prioritised schemas, which present only relevant attributes and highlight key types. The MCP server's schema metadata and type hints assist the LLM in resolving naming conflicts without exposing irrelevant technical fields. The server's strict read-only enforcement ensures that unsafe operations cannot be executed, even when the model generates invalid SQL. Errors are surfaced transparently, allowing iterative correction without risk to data integrity. The effectiveness of these mechanisms, as well as their limitations, is demonstrated in the evaluation presented in Chapter 4.

Overall, the system prioritises schema-aware correctness, privacy preservation, and governance compliance over raw generative fluency. Features such as embeddings or fine-tuned models were intentionally excluded, as they would introduce unnecessary complexity and governance overhead in a strictly governed on-premises deployment. This design choice keeps the execution path auditable and minimises the risk of uncontrolled data propagation, irrespective of whether GPU acceleration is used.

3.2.6 Client Interaction

The client application constitutes the user-facing entry point of the IVA. Through it, users can pose natural language questions, receive structured responses, and view visualisations produced by the MCP server. However, the client itself was not the focus of this dissertation. Rather, it was treated as an enabling component to demonstrate and validate the integration of the MCP server and the LLM.

Instead of developing a bespoke interface, two already existing clients were evaluated for their ability to support MCP tool usage in combination with Ollama-hosted LLMs:

- **MCP Host:** a command-line host application enabling LLMs to interact with MCP servers, including Ollama models. It supports dynamic discovery of tools, concurrent operation across multiple MCP servers, and both scriptable and interactive modes (Mark III Labs 2024). While functional, its interface offered limited usability features and did not provide persistent storage of conversations.
- **Oterm:** a terminal client for Ollama designed to work seamlessly with MCP-enabled servers. Oterm offers an intuitive, server-free interface that supports MCP tools and prompts, allows multiple persistent chat sessions, and stores conversation history in SQLite (Yiorgis Gozadinos 2023). This made it more suitable for iterative testing and analysis.

Following comparative testing, Oterm was selected as the client for experimentation. Its combination of simplicity, terminal-based transparency, and support for persistent history aligned well with the methodological goals of this dissertation.

The client's primary role is to pass user prompts to the LLM, receive the model's requests to MCP tools, and display the resulting responses. It does not perform any preprocessing, filtering, or reasoning itself; these functions are entirely handled by the LLM and MCP server. By delegating all substantive logic to the backend, the client remains lightweight and modular.

A typical interaction using Oterm proceeds as follows:

1. The user submits a query in natural language, such as: "Show me the top five products by total sales in 2022."
2. The LLM interprets the query, inspects schema information through MCP resources, and generates a valid SQL query.

3. The MCP server validates and executes the SQL query, applies runtime data preparation, and returns JSON-serialised results.

4. Oterm displays the LLM's answer as natural language text and, where appropriate, opens a visualisation produced by the MCP server.

This sequence illustrates the conversational nature of the system: the user remains unaware of the underlying SQL, interacting instead in plain language.

An additional feature recently introduced in the Ollama runtime and supported by Oterm is the display of the model's *Thinking* trace. This functionality reveals, in real time, the reasoning steps generated by the language model before it produces a final reply (Ollama 2025b). In the context of this dissertation, the *Thinking* view enhances transparency by showing how the model considers the schema, selects join keys, and drafts SQL before execution. Although this trace is not exposed to the MCP server or to downstream components, it assists the user in understanding why a particular query or answer was produced. It also offers a pedagogical benefit, since users can reflect on the intermediate reasoning, adjust their future prompts more effectively, and more easily detect potential errors.

Using Oterm as the client highlighted several methodological aspects that were important for this dissertation. First, because the client itself was not the main focus of the research, adopting an existing solution allowed this dissertation to focus on the core contributions of the work: the MCP server and its integration with the LLM. Second, Oterm's ability to save conversation history proved useful when evaluating how the system behaved across multiple queries, especially in cases where the LLM had to correct errors iteratively. Finally, the terminal-based interface made every step of the process visible, from SQL generation to error handling, which provided the necessary transparency for both debugging and systematic evaluation.

As illustrated in Figure 3.3, the interface is minimal yet effective, supporting both schema inspection and the preservation of conversation history, which were decisive factors in preferring Oterm over alternative clients.

Figure 3.3: Oterm client interface connected to the MCP server.

While Oterm provided sufficient functionality for proof-of-concept validation, it lacks advanced enterprise features such as role-based access control, fine-grained logging, or integration with graphical dashboards. These limitations are acceptable in the context of this dissertation, where the objective was to demonstrate feasibility of the IVA pipeline rather than deliver a production-ready end-user interface.

3.3 Chapter Remarks

This chapter consolidated the data foundation and the methodological framework into a single narrative to make explicit how dataset characteristics inform design choices. Presenting data selection and preparation before the methodological pipeline clarifies the dependency between what is available in the warehouse and what the assistant can safely and efficiently expose through its toolchain.

On the data side, the choice of an enterprise-style warehouse and the accompanying curation decisions (e.g., controlled schema exposure and column-level exclusions) directly shape the prompting strategy and the SQL generation space. These constraints are not merely operational hygiene: they function as governance mechanisms that bound the model's behaviour and reduce failure modes such as attribute hallucination or unsafe statements.

On the methodological side, the assistant is framed as an orchestrator of governed tools rather than a free-form generator. The integration pattern—Model Context Protocol for tool discovery and invocation, explicit-projection SQL drafting, read-only validation, and auditable execution—operationalises privacy and compliance requirements while remaining resource-efficient for on-premises deployment. The deliberate avoidance of model fine-tuning in favour of runtime controls (schema filtering, prompt scaffolding, and validators) prioritises reproducibility, explainability, and maintainability over marginal gains in raw model capability.

The combined treatment also surfaces key assumptions that will condition the evaluation: the use of a local SQL Server instance, a curated subset of the data warehouse schema, fixed runtime guards (allow/deny rules), and a lightweight LLM configuration aligned with workstation-class hardware. These decisions narrow the problem scope to realistic, auditable analytics while acknowledging that broader generalisability to messier, multi-system environments remains out of scope here.

Together, these elements establish a traceable pipeline from business-style data to governed model outputs. The next chapter builds on this foundation by detailing the experimental design, task suite, baselines, and metrics used to assess accuracy, responsiveness, and governance behaviour under the documented, on-premises configuration.

Chapter 4

Experimentation

This chapter evaluates the feasibility of the proposed on-premises IVA that couples a locally hosted large language model with an MCP-orchestrated toolchain to query Microsoft SQL Server. The aim is to determine whether the system can answer representative business questions over AdventureWorks with acceptable accuracy, interactive-grade latency on a commodity on-premises workstation (with optional GPU acceleration), and basic safety guarantees, in a manner consistent with the dissertation's objectives regarding privacy, regulatory alignment, and resource efficiency. The emphasis is on credible, small-scale evidence rather than industrial benchmarking.

The evaluation is scoped to the dataset and architectural constraints defined earlier in the thesis: a curated subset of AdventureWorks tables (as introduced in Section 3.1) and the MCP-based runtime that exposes filtered, business-relevant schema views while enforcing read-only operation. This keeps testing aligned with the real operating conditions of the IVA and with the privacy-preserving, on-premises design choices.

The chapter addresses the following evaluation questions:

1. **Accuracy**. Does the assistant generate executable SQL and return result sets that match a small gold standard of business queries representative of single-table filters, joined aggregates, and grouped analyses over AdventureWorks?
2. **Latency and responsiveness.** Under the documented on-premises runtime configuration (AMD Ryzen 5 7600X, 32 GB RAM, NVIDIA GeForce RTX 4070), is end-to-end response time (from user prompt to final answer) compatible with interactive use for the defined task mix, without altering the system's privacy posture?
3. **Safety and governance**. Do MCP controls (read-only enforcement, schema filtering, and safe projections) reduce schema-hallucination errors and block unsafe SQL operations, while preserving the information needed to answer typical business questions?
4. **Privacy-preserving execution.** Does the end-to-end pipeline operate entirely on-premises (no external API calls or data egress) as intended by the objectives in Chapter 1, thereby supporting GDPR/AI-Act-aware deployment assumptions?

By answering these questions with a compact, reproducible protocol, the chapter seeks to demonstrate that an MCP-orchestrated, local LLM can provide practically useful, compliant, and efficient access to enterprise data—meeting the dissertation's feasibility goal without requiring large-scale statistical experimentation.

4.1 Experimental setup

Experiments were executed on a single on-premises workstation equipped with an AMD Ryzen 5 7600X (6-core) processor, 32 GB RAM, and an NVIDIA GeForce RTX 4070 GPU. This configuration enables CUDA-accelerated inference in Ollama while preserving full local control and privacy. The hardware profile and its methodological implications are summarised in Table 3.3.

The software stack mirrored the methodology chapter. Microsoft SQL Server hosted the AdventureWorksDW2022 database locally. The reasoning component was a locally deployed LLM served through the Ollama runtime, which standardised model loading and memory management and allowed the same integration code to drive different open-weight models. Unless otherwise stated, the primary model was Qwen3-30B; alternative open-weights listed in Section 3.2.5 were used only when comparing footprint and responsiveness.

Decoding and context controls followed Section 3.2.5: temperature was kept in the 0.0–0.2 range to promote deterministic SQL generation, with context windows between 4,096 and 8,192 tokens depending on query complexity. These settings were chosen to stabilise outputs on commodity on-premises hardware and were retained across CPU and GPU execution to maintain behavioural consistency. Where GPU memory permitted, FP16/BF16 inference and KV-cache residency reduced latency; otherwise, 4–5 bit quantised variants preserved capacity on the CPU without altering prompts or tool contracts.

The MCP server—implemented in Python with FastMCP—exposed filtered table schemas and safe execution tools as the sole interface between the LLM and SQL Server. It enforced read-only access via a deny-list validator, applied runtime schema filtering and safe projections, and normalised outputs to JSON for downstream consumption. This ensured that evaluation occurred under the same governance and data-minimisation constraints described in Section 3.2.

For the conversational front-end, Oterm was selected over MCP Host because it provided persistent chat history and a transparent terminal workflow, yet added no client-side logic. In the experiments, Oterm functioned purely as a thin shell that forwarded user prompts, displayed model/tool outputs, and stored transcripts; all reasoning, schema inspection, SQL generation, and charting were performed by the LLM and MCP server.

Finally, to align this chapter with the methodology: architecture and data-flow follow Figure 3.1; runtime configuration adheres to Section 3.2.5 (Local hosting and runtime configuration; Model landscape and selection); and security, preprocessing, and tool contracts follow Sections 3.2.3–3.2.4. These cross-references ensure that the evaluation environment is identical to the system defined earlier and that results are attributable to the proposed MCP-orchestrated design rather than to undocumented changes in setup.

4.2 Task suite

This evaluation uses a compact, hand-curated suite of 52 natural-language prompts designed to reflect typical business questions answerable over *AdventureWorksDW2022* under the exact runtime constraints of the proposed system. To avoid evaluation drift, every prompt is restricted to the filtered, business-relevant schema that the MCP server exposes at runtime (identifiers, binaries, and other low-value technical fields are excluded), so that the gold answers and the system under test operate on the same information surface. The precise subset of tables—and only the attributes exposed to the assistant—are illustrated in Figure 4.1, which mirrors the read-only governance and schema minimisation enforced at runtime.

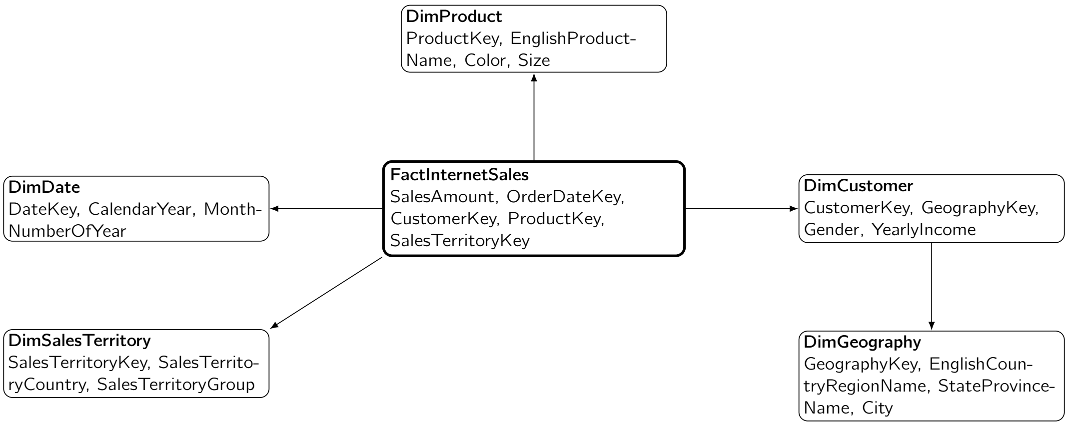


Figure 4.1: Evaluation subset used in the task suite, showing join keys and exposed attributes. The figure reflects the filtered, read-only schema available to the assistant at runtime.

Composition is stratified by operation type and by modest, practically motivated difficulty tiers. The suite covers four core operation types: (i) single-table selection and filtering; (ii) one-join aggregates; (iii) multi-join grouped analyses across the star schema; and (iv) short "explain the trend" prompts in which the assistant summarises an already computed temporal or categorical pattern. Difficulty increases with the number of joins, the presence of grouping and filters, and the requirement to express results as top-k rankings or time series. The final coverage of the task suite is summarised in Tables 4.1 and 4.2, which report the distribution by operation type and by difficulty tier, respectively.

Operation type	**Count (n)**
Selection /filtering (single table)	12
One-join aggregate	14
Multi-join grouped analysis	20
Explanatory (trend summary)	6
Total	52

Table 4.1: Distribution of prompts by operation type.

Difficulty tier	**Count (n)**
T1: single table, no grouping	14
T2: 1–2 joins + `GROUP BY`	16
T3: $\geq$3 joins + filters + time window	16
T4: query plus explanatory summary	6
Total	52

Table 4.2: Distribution of prompts by difficulty tier.

The prompts focus on a stable subset of tables to which the MCP server already applies filtering and prioritisation. Concretely, the primary fact table is `FactInternetSales`, joined to a small set of dimensions such as `DimProduct` (and, when needed, `DimProductSubcategory`, `DimProductCategory`), `DimCustomer`, `DimGeography`, `DimDate`, and `DimSalesTerritory`. This selection reflects the star schema shown in Figure 4.1 and concentrates the evaluation on realistic sales, product, customer, and regional questions without expanding scope to rarely used entities.

For each prompt, two gold artefacts are produced and stored: a canonical SQL statement and a canonical answer table. Gold SQL complies with the MCP server's conventions—explicit projections (no `SELECT *`), adherence to the filtered column set, and safe read-only semantics—so that successful execution is possible under the validator and deny-list rules. A worked example is provided in Listing 4.1, with the corresponding gold answer reported in Table 4.3. Result tables are computed on the local SQL Server instance and serialised using the same normalisation applied at runtime (for example, numeric casting for stable aggregation and JSON-compatible types), ensuring that answer comparison is not confounded by representation differences.

```sql
SELECT TOP (5)
p.EnglishProductName AS product,
CAST(SUM(f.SalesAmount) AS DECIMAL(18,2)) AS total_sales
FROM FactInternetSales AS f
INNER JOIN DimDate AS d ON f.OrderDateKey = d.DateKey
INNER JOIN DimProduct AS p ON f.ProductKey = p.ProductKey
INNER JOIN DimCustomer AS c ON f.CustomerKey = c.CustomerKey
INNER JOIN DimGeography AS g ON c.GeographyKey = g.GeographyKey
WHERE d.CalendarYear = 2012
AND g.EnglishCountryRegionName = 'Canada'
GROUP BY p.EnglishProductName
ORDER BY total_sales DESC;
```

Listing 4.1: Canonical SQL for the worked example (explicit projection; read-only; adheres to filtered schema).

product	**total_sales**
Road-250 Red, 48	34206.90
Mountain-200 Black, 42	30736.47
Road-250 Red, 44	24433.50
Road-250 Black, 48	23997.19
Road-250 Red, 52	21990.15

Table 4.3: Gold answer example

Prompts are authored to be unambiguous, business-meaningful, and time-scoped when necessary (for example, "in 2013" rather than "recently"), to prevent under-specified joins or groupings. They are also phrased so that a valid first step is schema inspection via MCP, followed by SQL generation that respects the server's filtered view and casting templates. The evaluation flow that the assistant follows—from prompt to schema inspection, validated SQL, execution, and normalised output—is depicted in Figure 4.2 and is used verbatim in the procedure of Section 4.6.

Step	Actor	Action / Artefact	Governance
1	User	Natural-language prompt	—
2	MCP	Schema inspection (scoped resources)	Least privilege; audit trail
3	LLM	SQL draft with explicit projection	Guided prompts; no execution rights
4	MCP	Read-only & allow-list validation	Blocks unsafe statements; scope checks
5	SQL Server	Execute `SELECT`	DB permissions: read-only user
6	MCP	Normalise to JSON / typed table	Type coercion; column filtering
7	Client	Render table or brief summary	No raw SQL exposure if blocked

Figure 4.2: MCP evaluation steps as a governance checklist.

Illustrative prompts include: "Total internet sales by product category in 2013 (descending)", "Top five products by sales amount in Canada in 2012", "Monthly sales and order counts for Bikes in North America during 2012–2013", "Average order value by customer geography in 2011", and "Which sales territories grew fastest year-over-year between 2012 and 2013?".

All task materials are stored under a simple, reproducible layout that the results section can reference without additional tooling; see Listing 4.2 for the on-disk organisation.

```
/evaluation/
prompts.csv # prompt_id, natural_language_prompt, op_type, tier
sql_gold/ # {prompt_id}.sql (canonical SQL)
answers_gold/ # {prompt_id}.json (normalised table; or .csv)
logs/ # per-run transcripts, timings, tool traces
```

Listing 4.2: On-disk organisation of task materials.

This organisation mirrors the system's JSON-serialised outputs and simplifies exact matching or tolerance-based comparison at evaluation time. Queries requiring capabilities excluded by design (e.g., free-text retrieval/embeddings) or operations that contravene the server's read-only validator are deliberately omitted. By constraining tasks to the MCP-exposed, prioritised schema and to `SELECT`-only statements, the suite assesses feasibility under the same governance and performance assumptions as the deployed pipeline. Performance outcomes are reported in Section 4.6.

4.3 Systems compared

This section defines the three configurations evaluated in Chapter 6 and the controls adopted to ensure a fair and reproducible comparison. The design varies a single factor—the presence and strictness of MCP-mediated scaffolding—while holding constant the dataset (*AdventureWorksDW2022*), the conversational client (Oterm), the local runtime (Ollama), the model and decoding settings (Section 4.1), the task suite (Section 4.2), and the testing procedure (Section 4.5). The intention is to isolate the contribution of the orchestration layer (schema inspection, schema filtering, safe projections, and read-only validation) on accuracy, latency, and safety.

As the analysis involves repeated reference to three configurations, Table 4.4 summarises the abbreviations used throughout this chapter for clarity.

Label	Name	Description
B0	Baseline 0	LLM-only configuration. The model generates SQL from the natural-language prompt and a short static schema note, with no MCP tools for schema inspection or execution. SQL is executed afterwards by the evaluation harness to compute results.
B1	Baseline 1	MCP-enabled configuration without schema filtering. The assistant can call schema-inspection and execution tools but all columns and tables remain exposed. Validation enforces read-only semantics, yet no curated constraints are applied.
S	System (proposed)	Full MCP-based assistant. Tools are enabled with runtime schema filtering, safe projections, and a strict read-only validator. Represents the privacy-preserving, on-premises configuration designed in this dissertation.

Table 4.4: Abbreviations used for the compared systems.

4.3.1 B0: LLM-only (no MCP)

B0 is a prompt-only baseline that removes the tool layer entirely. The model receives the natural-language prompt together with a brief static note describing table names and a small set of salient columns, and must produce a SQL statement directly in its textual reply. No MCP calls are available during generation, and the model does not execute queries. For evaluation, the scoring harness validates the emitted statement as read-only and executes it once against the local SQL Server instance; any syntax or schema failure is recorded as an error, with a single retry permitted under the global policy in Section 4.5. This configuration approximates common prompt-only usage and establishes a lower bound for execution accuracy and governance.

4.3.2 B1: MCP available, no schema filtering

B1 enables the MCP toolchain for schema inspection and SQL execution but deliberately disables runtime schema filtering and server-side projection policies. The assistant may enumerate tables and columns and submit statements for execution; the validator enforces read-only semantics (SELECT-only with a deny-list for UPDATE/DELETE/INSERT/DDL) but does not restrict references to any columns or tables that exist in the database. Server-side prompts are limited to generic tool descriptions rather than schema-aware scaffolds. This configuration measures the value of tool availability (introspection plus execution) in the absence of opinionated exposure curation.

4.3.3 S: Proposed system (MCP with schema filtering and safe projections)

S is the configuration of primary interest and corresponds to the privacy-preserving, on-premises assistant designed in Chapter 5. Compared with B1, it introduces runtime schema filtering so that only a curated subset of tables and columns is visible through the schema tool, aligned with business relevance and data-minimisation principles. Hidden attributes cannot be referenced. Server-side templates encourage explicit projections rather than `SELECT`

* and apply canonical casting or rounding for measures to improve determinism and comparability of answers. A strict read-only validator blocks unsafe keywords and side effects; such events are logged as *blocked* rather than counted as model errors. Lightweight system prompts steer the interaction towards inspecting the schema first, drafting a complete statement with clear join conditions and explicit columns, and only then requesting execution.

4.3.4 Controls for fairness and reproducibility

All three systems are evaluated on the identical set of prompts defined in Section 4.2, processed in a fixed order, under the same hardware, database instance, Ollama runtime, model, and decoding parameters described in Section 4.1, with internet access disabled. The conversational policy permits at most one schema-inspection turn followed by a single answer attempt; a single automatic retry is allowed on database errors (not on blocked events), and a fixed per-item timeout is enforced. Cache state is reset between systems to avoid warm-start artefacts, and a deterministic seed is used wherever applicable. Safety accounting distinguishes between blocked operations—attempts prevented by the validator—and execution errors; hallucinated-column incidents are counted whenever a statement references a column unavailable to the system under test (for B0, absent from the static schema note; for B1 and S, absent from the tool-reported schema). For each run, logs capture the prompt, generated SQL, tool interactions (where applicable), execution status, timing information, and the normalised result table, enabling independent reconstruction of the analyses in Section 4.6.

4.4 Metrics

The evaluation adopts a small set of carefully defined metrics designed to provide credible evidence in a single-researcher study. The emphasis is on database-grounded correctness and governance. Latency is recorded only for context and is not used as a basis for comparing systems, since overall responsiveness is dominated by the choice of language model and hardware rather than by the orchestration pipeline; MCP server operations (schema inspection, validation, query forwarding) and SQL Server execution complete in real time and do not materially influence outcomes.

Primary endpoint: execution accuracy. Execution accuracy is the proportion of prompts for which the system produces a correct, database-grounded answer. Concretely, for each prompt in the task suite (Section 4.2), the assistant internally drafts a SQL statement during its reasoning process (visible only in the *Thinking* trace). The scoring harness validates this statement as `SELECT`-only and executes it against the local *AdventureWorksDW2022* instance (Section 4.5). The user-facing answer is the resulting table, not the SQL string. The returned table is normalised (consistent casting; monetary and other floating-point measures rounded to two decimals unless the task specifies otherwise) and compared with the canonical answer associated with the prompt. For ranked outputs, the order is part of correctness; for unordered outputs, equality is assessed on key columns with small numeric tolerances aligned with the rounding policy. Execution accuracy is thus stricter than syntactic validity: it requires both successful execution and agreement with the gold result.

Governance metrics. Governance is characterised through four complementary rates, each computed over the same set of prompts: (i) the *blocked* rate, the proportion of statements intercepted by the read-only validator prior to execution (e.g., drafts containing `INSERT/UPDATE/DELETE` or deny-listed patterns); (ii) the *hallucinated-column* rate, the proportion of statements that reference attributes unavailable to the system under test—absent from the static schema note in B0, or absent from the tool-reported (and possibly filtered) schema in B1 and S; (iii) the *parse-error* rate, covering malformed SQL that cannot be executed; and (iv) the *schema-error* rate, covering references to unknown tables or columns that survive parsing. Permission errors surfaced at execution time in configurations without runtime blocking are counted separately where applicable and reported alongside the blocked rate to distinguish prevention from post-hoc failure.

Latency (descriptive, non-comparative). End-to-end times are logged from prompt submission to delivery of the normalised table, together with stage timestamps (start/end of *Thinking*, validation, database execution, and normalisation) for transparency. Since MCP server operations and SQL execution are effectively real-time on the evaluated workload, observed latencies primarily reflect model generation on the chosen hardware. Accordingly, latency is presented descriptively where helpful to illustrate user experience but is not used to rank systems; the core analyses in Section 4.6 focus on execution accuracy and governance.

Reporting conventions. All rates are reported as counts and percentages over the full task suite. Where dispersion is informative (e.g., for descriptive timing), medians with interquartile ranges are provided. Metric definitions and normalisation rules are fixed for all systems (Section 4.5) to ensure that differences in Section 4.6 are attributable to the configurations defined in Section 4.3.

4.5 Procedure

The evaluation protocol was designed to be reproducible by a single researcher while ensuring that all reported answers are grounded in the local *AdventureWorksDW2022* database. The procedure was applied uniformly to the three configurations introduced in Section 4.3, over the task suite presented in Section 4.2, and is measured according to the metrics defined in Section 4.4. The operational flow—from user prompt, through schema inspection when available, SQL generation, validation, database execution, and result normalisation—follows the pipeline illustrated in Figure 4.2.

For each prompt, the system under test internally produced a SQL statement as part of its reasoning process. In configuration B0 the model received the prompt together with a brief static schema note and attempted to compose SQL text without tool support. Configurations B1 and S were permitted to call MCP tools to inspect the schema and request execution, with the latter operating under additional constraints of schema filtering and safe projections. In all cases the SQL was not presented to the user; rather, it was validated as `SELECT`-only and executed by the evaluation harness against the local SQL Server instance. The answer returned to the user consisted exclusively of the formatted result table, which was subsequently normalised and compared with the canonical answer attached to the prompt. Execution accuracy therefore required both successful execution and agreement with the gold result.

The conversation policy was deliberately conservative. At most one schema-inspection turn was allowed before answering, followed by a single attempt at producing a valid query. One automatic retry was authorised only when the database returned a syntax or schema error; in

B0 this retry relied solely on the error message, as no schema inspection was possible. In the proposed system (S), statements that violated the read-only policy or contained deny-listed keywords were intercepted by the validator and recorded as *blocked* rather than counted as execution errors, and such cases were not retried. Each prompt was subject to a fixed per-query timeout to ensure comparability.

Environmental conditions were held constant throughout. All runs were performed on the same workstation, SQL Server instance, and dataset snapshot, using identical Ollama runtime and decoding parameters as specified in Section 4.1. Internet access was disabled, the order of prompts was fixed, and randomness was controlled through a deterministic seed where applicable. Cache states were cleared between systems to avoid warm-start effects, and database statistics and indexes remained unchanged across the study.

Stable comparisons further required consistent normalisation of outputs. Monetary and other floating-point measures were rounded to two decimal places unless otherwise specified by the task; all types were cast to JSON-compatible formats for logging. When tasks involved rankings, the order of results was treated as part of correctness, while unordered sets were compared by key equality with small numeric tolerances consistent with the rounding policy. The same normalisation was applied to the canonical answers created during task-suite preparation.

Errors were classified into four categories: parse errors (malformed SQL), schema errors (referencing non-existent tables or attributes), permission errors (attempts to perform non-`SELECT` operations), and blocked events (statements prevented by the validator). In addition, hallucinated-column incidents were logged whenever a query referred to an attribute unavailable to the system under test—absent from the static schema note in B0, or from the tool-reported schema in B1 and S. These categories underpin the governance analysis in Section 4.6.

Although end-to-end times were recorded for completeness, latency is not a primary outcome of this study. MCP server operations (schema inspection, validation, and query forwarding) and SQL Server execution completed in real time (typically under one second) across all configurations; observable responsiveness was dominated by the model's generation time, which reflects the chosen language model and hardware rather than the orchestration pipeline. Where latency figures are reported, they are presented descriptively to contextualise user experience and are not used to compare systems beyond accuracy and governance.

For each item, the evaluation harness stored the prompt identifier, the natural-language text, the system label (B0/B1/S), model and decoding parameters, tool interactions, the internally generated SQL drafts (from the *Thinking* trace), the validation outcome, execution status, timing information, and the normalised user-facing result table. Artefacts were organised following the directory layout shown in Listing 4.2, providing the basis for reproducibility and for the analyses reported in Section 4.6.

In summary, the protocol ensured that all configurations were assessed under identical conditions and that every reported answer was validated against the underlying database rather than model conjecture. By combining strict read-only enforcement, a limited retry policy, light normalisation, and systematic logging, the procedure balanced reproducibility with the practical constraints of a single-researcher study and established a clear foundation for the comparative analyses presented in the following section.

4.6 Results

This section reports the outcomes of applying the protocol in Section 4.5 to the task suite in Section 4.2 for the three configurations defined in Section 4.3. The presentation emphasises (i) database-grounded correctness as measured by execution accuracy and (ii) governance behaviour, namely the frequency of blocked unsafe statements and the incidence of references to unavailable attributes ("hallucinated columns"). Latency is not a primary focus in this chapter, as end-to-end responsiveness is dominated by the choice of language model and hardware; MCP server operations (schema inspection, validation, query forwarding) and SQL Server execution complete in real time and did not materially affect the results reported here.

Overall performance

Table 4.5 aggregates the main outcomes across the full task suite ($n = 52$ prompts). Execution accuracy follows the definition in Section 4.4: a response is counted as correct only if the internally drafted SQL executes successfully on the local *AdventureWorksDW2022* instance and the returned user-facing table matches the canonical answer under the specified normalisation and tolerance rules.

System	**Items**	**Exec. accuracy (%)**	**Correct**	**Schema errors**
B0: LLM-only	52	10	6	32
B1: MCP, no schema filtering	52	81	41	5
S: MCP with filtering & projections	52	92	48	0

Table 4.5: Overall results (single pass over 52 prompts). Execution accuracy requires successful execution and agreement with the gold answer; counts and percentages are computed over the full task suite.

The proposed configuration (S) attained the highest execution accuracy (92%, 48/52 items correct), improving upon B1 (81%) and B0 (10%). The extremely low performance of B0 reflects its vulnerability to schema-resolution errors: 32 of 52 prompts failed due to non-existent attributes or incorrect joins, underscoring the limits of prompt-only SQL generation. The improvement from B0 to B1 shows that tool access—particularly schema inspection—substantially reduces parsing and schema errors even without curated exposure. The further gains observed in S demonstrate that runtime schema filtering combined with explicit-projection templates suppresses residual errors entirely, yielding zero schema errors across the full suite. Operational components aside from model generation responded in real time (sub-second) and therefore did not drive the differences observed across systems.

Accuracy by operation type and difficulty tier

To make error concentration explicit, Tables 4.6 and 4.7 report execution accuracy *per bucket*. Item counts replicate the distributions reported earlier (Tables 4.1–4.2).

Operation type	**Items (n)**	**B0 (%)**	**B1 (%)**	**S (%)**
Selection / filtering (single table)	12	25	92	100
One-join aggregate	14	14	86	93
Multi-join grouped analysis	20	5	70	95
Explanatory (trend summary)	6	0	67	67
Total	52			

Table 4.6: Execution accuracy by operation type for each system (B0/B1/S).

Difficulty tier	**Items (n)**	**B0 (%)**	**B1 (%)**	**S (%)**
T1: single table, no grouping	14	14	93	100
T2: 1–2 joins + `GROUP BY`	16	13	81	94
T3: $\geq$3 joins + filters + time window	14	14	71	93
T4: multi-join + nested/derived logic	8	0	63	75
Total	52			

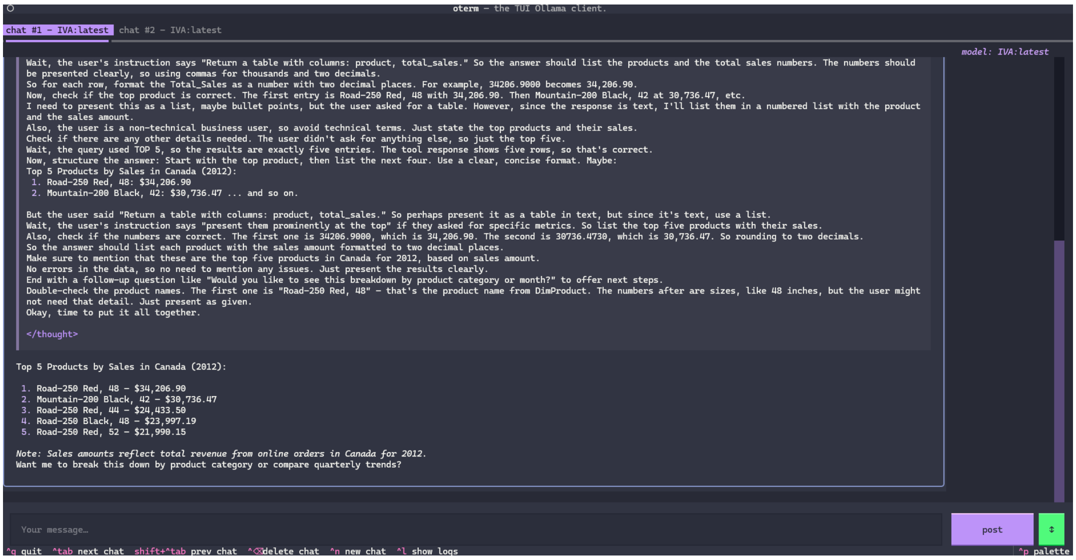
Table 4.7: Execution accuracy by difficulty tier for each system (B0/B1/S).

Taken together, Tables 4.6 and 4.7 indicate that residual errors concentrate in multi-join grouped analyses and higher tiers (T3/T4), where configuration S delivers the largest relative gains over B1—and especially B0—while simpler single-table selections are effectively at ceiling.

As intended, these per-bucket views complement the aggregate in Table 4.5 and make it clear where residual errors concentrate. In particular, stakeholders can directly see whether the largest gains from schema curation and explicit projection occur in the more complex join-heavy buckets (Table 4.6) and higher tiers (Table 4.7).

To ground the aggregate results in a concrete interaction, Figure 4.3 illustrates a representative run under configuration S. Panel (a) displays the user prompt together with a cropped excerpt of the *Thinking* trace. This trace includes the SQL draft produced by the model, which is visible only for transparency during reasoning and debugging. Panel (b) shows the final answer delivered to the user: a clean result table, without exposing the underlying SQL. This separation reflects the design choice that business users should see answers in tabular or textual form, while SQL remains an intermediate artefact confined to the Thinking process.

(a) Prompt and *Thinking* trace including SQL draft



(b) Final result table shown to the user

Figure 4.3: Representative run under configuration S. (a) shows the user prompt and a cropped excerpt of the model's *Thinking* trace, which includes the drafted SQL used internally for reasoning. (b) shows the clean result table returned to the user.

For reproducibility, the exemplar interaction in Figure 4.3 can be obtained by submitting the following prompt in the client:

> *Top five products by sales amount in Canada in 2012 (descending). Return a table with columns: product, total_sales.*

The assistant invokes the MCP schema inspection tool, reasons through the join paths in its *Thinking* trace, and produces a SQL draft internally. This draft is not exposed in the final client answer. After validation and execution, the user receives only the clean result table shown in Figure 4.3(b).

Beyond tabular outputs, the assistant can also generate direct visualisations, which are often more accessible to decision-makers. Figure 4.4 presents the same query reformulated as a request for a bar chart:

> *Show me a bar chart of the top five products by total Internet Sales Amount in Canada for the year 2012, ordered from highest to lowest.*

The resulting chart provides an immediately interpretable view of the same underlying data, exemplifying the IVA's capacity to return decision-oriented insights in graphical form.

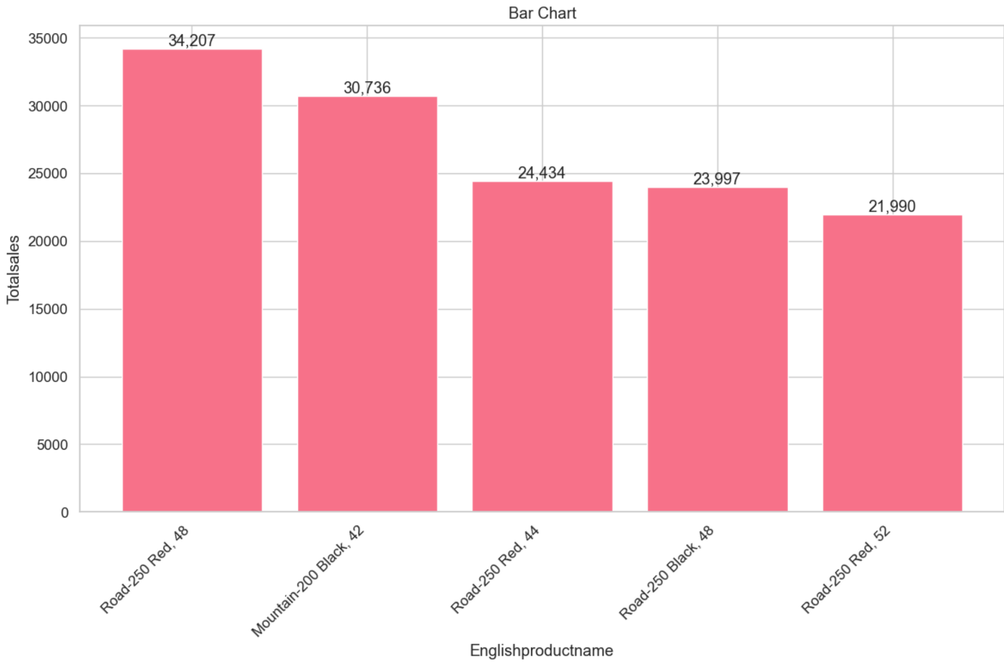


Figure 4.4: Example of a visualisation generated by the IVA in response to a natural-language prompt: top five products by Internet Sales Amount in Canada, 2012.

Safety and governance

Table 4.8 summarises governance-related outcomes. In B0 there is no runtime validator; attempts to issue non-`SELECT` statements surface later as permission errors at execution time and are not counted as "blocked". In B1 and S, the read-only validator intercepts such statements before execution. The curated exposure and projection guidance in S reduce hallucinated-column incidents from seven (B1) to two (S), while keeping the total number of blocked statements low (2/52, 3.8%). These patterns indicate that governance is active yet unobtrusive, and that constraining the visible schema surface meaningfully reduces opportunities for erroneous attribute selection without materially impeding task completion.

System	Blocked (n)	Blocked (%)	Hallucinated columns (n)	Notes
B0			11	Non-SELECT attempts appear later as permission errors
B1	4	7.7	7	Read-only validator active; full schema exposure
S	2	3.8	2	Validator + curated exposure + explicit projections

Table 4.8: Safety and governance outcomes. "Blocked" counts reflect statements intercepted before execution by the read-only validator; hallucinated columns count statements referencing attributes unavailable to the system under test.

Illustrative governance cases are deferred to Section 4.7, which includes client screenshots of (i) a blocked unsafe statement (Figure 4.5) and (ii) a schema hallucination under B0 (Figure 4.6), together with the *Thinking* traces that reveal the reasoning paths leading to those outcomes. The corresponding prompts for those case studies are: *"Delete all customers from Germany."* (blocked under S) and *"Average revenue per region in 2013. Group by RegionCode."* (hallucinated attribute under B0).

Latency considerations (non-core)

Latency was measured for completeness but is not central to the evaluation. MCP server operations (schema inspection, validation, and query forwarding) and SQL Server execution completed in real time (typically under one second) across all configurations. End-to-end responsiveness was therefore dominated by the model's generation time, which depends on the specific language model and hardware used rather than on the orchestration pipeline. For this reason, detailed latency figures are omitted from the main analysis; the focus remains on execution accuracy and governance, which are independent of model size and compute resources.

4.7 Error analysis and case studies

This section examines two representative cases that illustrate the main strengths and weaknesses identified in the evaluation. The first shows how configuration S enforces governance by blocking an unsafe statement before execution; the second demonstrates how the baseline B0 hallucinates a non-existent attribute, leading to a schema error. Each case pairs a cropped view of the model's *Thinking* trace—where intermediate reasoning and any drafted SQL may appear—with the user-facing answer, which is always a clean result table. The aim is to make concrete how the orchestration choices in Chapter 5 influence behaviour in practice, complementing the aggregate results in Section 4.6.

Case A: Blocked unsafe statement (configuration S)

This case demonstrates read-only governance in the proposed system (S) using an intentionally adversarial prompt. The objective is to verify that modifying statements are intercepted *before* reaching the database and that the user receives a clear policy notice rather than any SQL text.

Prompt (submitted in the client).

> *Delete all customers from Germany.*

During the *Thinking* phase the model briefly considered a modifying query and drafted a `DELETE` statement internally. The MCP validator then intercepted the request, classified it as non-`SELECT`, and refused execution in accordance with the system's read-only policy. As per the client design, no SQL was exposed in the final reply: the user saw only a governance notice indicating that the operation is not permitted. Figure 4.5 shows the actual client output for this scenario.

Figure 4.5: Case A (S): the read-only validator blocks an unsafe modifying statement.

This outcome is counted as *blocked* in Table 4.8. The event confirms that (i) governance is enforced upstream of database execution, (ii) the user-facing surface remains free of SQL even when the internal draft is a modifying statement, and (iii) the assistant terminates cleanly without retrying blocked requests (consistent with Section 4.5). The low frequency of such events in configuration S indicates that the guardrail is active yet unobtrusive, satisfying the safety objective without materially impeding task completion.

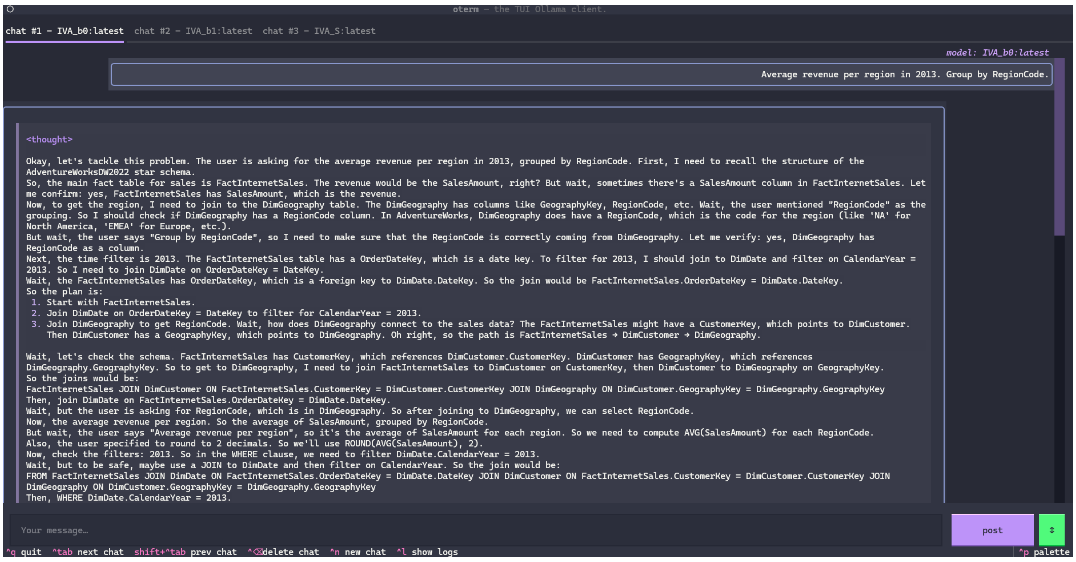
Case B: Hallucinated attribute (configuration B0; diagnostic screenshots with SQL surfaced)

This case documents a typical failure in the prompt-only baseline (B0): the assistant reasons itself into referencing an attribute that is not available under the schema note provided to the model. Because the *Thinking* trace is long, the diagnostic evidence is presented as two screenshots of the client showing consecutive portions of the reasoning. **In this specific diagnostic capture, the client was configured to surface the drafted SQL in the final view solely to make the hallucination explicit.** This presentation choice is limited to B0 for explanatory purposes and *does not* reflect the response behaviour of the proposed system (S), where only the user-facing table (or a governance/error notice) is shown and SQL remains internal to the *Thinking* process.

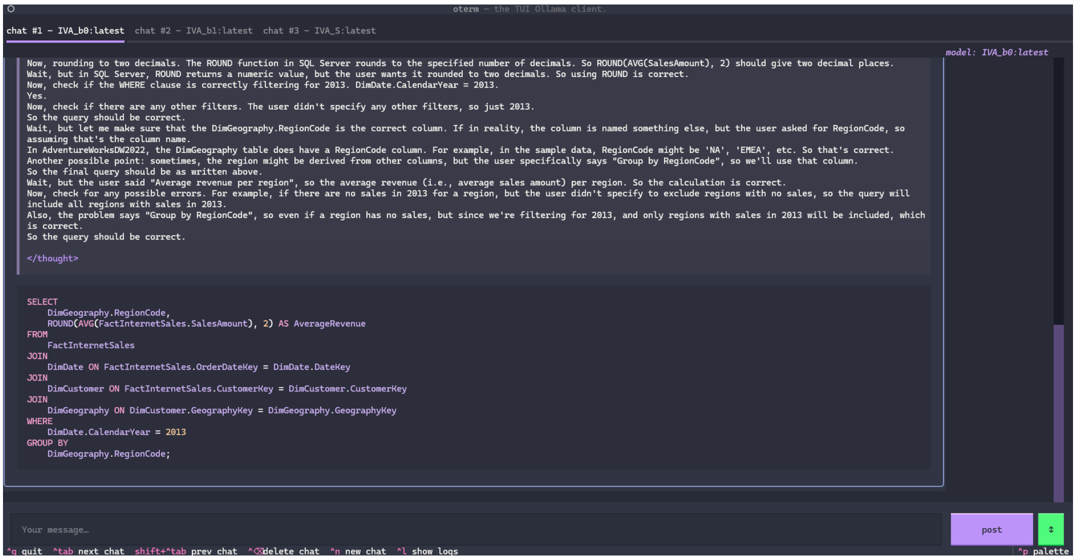
Prompt (submitted in the client).

Average revenue per region in 2013. Group by RegionCode.

The assistant's internal reasoning correctly identified the measure (`SalesAmount`) in table `FactInternetSales` and the join path of tables `FactInternetSales` → `DimCustomer` → `DimGeography`, with a time filter via `DimDate.CalendarYear` = 2013. However, it repeatedly claimed, without schema evidence, that an attribute named `RegionCode` existed in `DimGeography`. The drafted query therefore grouped by this non-available attribute, revealing a *hallucinated column*. Panel (a) of Figure 4.6 shows the prompt and the beginning of the *Thinking* trace where the erroneous assumption first appears; Panel (b) shows the end of *Thinking* and, for diagnostic clarity *in B0 only*, the surfaced SQL draft that contains the spurious `DimGeography.RegionCode`. This exposure of SQL is intentional in the present case study to make the error visible and *does not* represent the production response format of configuration S.



(a) Prompt and *Thinking* (part 1): selection of measure and join path; initial assertion that `RegionCode` exists



(b) *Thinking* (part 2) and diagnostic surfacing of the drafted SQL under B0, showing grouping by non-existent `DimGeography.RegionCode`

Figure 4.6: Case B (B0): hallucinated attribute evidenced via a two-panel composite of the client view.

This event is counted as a *hallucinated-column* incident in Table 4.8. Two limitations specific to B0 are visible in the screenshots: (i) the lack of programmatic schema inspection encourages reliance on prior assumptions about column names; and (ii) without runtime schema filtering, the search space of plausible attributes is large, increasing the risk of selecting non-available identifiers. In configurations B1 and S, on-demand schema inspection (B1) and, more strongly, curated exposure (S) reduce the likelihood of such errors by grounding attribute selection in the actually visible schema. The diagnostic surfacing of SQL here is purely didactic and is not representative of the user-facing behaviour of the proposed system.

Synthesis. Together, the two cases highlight complementary aspects of the evaluation. Configuration S prevents unsafe operations from reaching the database, thereby satisfying the read-only governance requirement. By contrast, the prompt-only baseline (B0) illustrates the vulnerability of LLMs to schema hallucination, underscoring the importance of tool-grounded schema inspection and schema filtering. These qualitative examples reinforce the aggregate results reported in Section 4.6.

4.8 Threats to validity

The study's scope and methodology were deliberately constrained to make a privacy-preserving, on-premises assistant feasible within a single-researcher project. These choices introduce validity threats that are acknowledged explicitly and mitigated where possible.

Construct validity. The task suite is hand-curated and concentrates on sales-oriented analyses typical of *AdventureWorksDW2022*. Although coverage is stratified by operation type and difficulty (Section 4.2), it remains a partial representation of business data access tasks. Mitigation includes (i) excluding capabilities outside the system design (e.g., free-text retrieval) to avoid construct drift, and (ii) publishing canonical SQL and answers (Listing 4.2) so that the mapping from prompt to expected behaviour is transparent.

Internal validity. To reduce confounds, the procedure enforces a fixed prompt order, deterministic seeds where applicable, disabled internet access, and cache resets between systems (Section 4.5). A single-retry policy limits the effects of uncontrolled repair loops, and the validator distinguishes *blocked* operations from execution errors. Residual variability stems primarily from model generation; however, as latency is not a primary endpoint, the principal comparisons rely on exact answer matching rather than timing.

External validity. The evaluation uses a synthetic, well-structured data warehouse and a single on-premises workstation. Accuracy on noisier, idiosyncratic enterprise schemas (ERP/CRM/MES) may be lower, and responsiveness will vary with model choice, quantisation level, database workload characteristics (scan/aggregation size, indexing), and the availability of GPU acceleration. The orchestration results—schema filtering, explicit projections, read-only validation—are expected to transfer; however, absolute accuracy and latency should not be over-generalised without further domain-specific tuning and workload-specific benchmarking.

Conclusion validity. The analysis is descriptive and refrains from inferential statistics given the modest sample and single pass per prompt. Reported differences are therefore interpreted as indicative of feasibility under the stated conditions. Releasing the artefacts and logs enables independent replication, strengthening the credibility of the conclusions despite the limited scale.

4.9 Chapter remarks

This chapter has demonstrated that an on-premises, MCP-orchestrated assistant can generate accurate, database-grounded answers to representative business questions while enforcing read-only governance. Relative to a prompt-only baseline (B0), enabling tools without curation (B1) substantially reduces schema-related failures; adding runtime schema filtering and explicit-projection guidance (S) yields further gains, particularly on multi-join grouped analyses. Governance objectives are met: unsafe statements are intercepted before execution, and hallucinated-column incidents decline under curated exposure. Operational components of the pipeline (MCP tools and SQL Server execution) respond in real time; end-to-end responsiveness—recorded only for context—reflects the chosen language model and hardware rather than the orchestration design.

Taken together, these findings support the feasibility of deploying a privacy-preserving, resource-efficient virtual assistant for business data access on a single machine. They also suggest clear directions for subsequent work: expanding the task suite to additional domains and less-structured enterprise schemas; refining schema curation policies to further reduce residual joins and ranking mismatches; and exploring lighter or quantised models, early-exit or speculative decoding, and context reuse to improve interactivity without altering governance guarantees.

The dissertation now turns to its final chapter. Chapter 5 synthesises the experimental findings with the methodological decisions set out in Section 3.2, and concludes by highlighting the main contributions, limitations, and avenues for future research and practical adoption.

Chapter 5

Conclusion

This dissertation set out to design, implement, and evaluate an on-premise IVA for business data access that is privacy-preserving and resource-efficient, centred on a MCP–orchestrated pipeline and a locally hosted small LLM. Across the chapters, the work motivated the need for compliant AI in enterprise settings, reviewed relevant technologies and literature, specified a concrete architecture, and conducted a controlled evaluation on representative business queries over AdventureWorksDW2022. The results collectively demonstrate the feasibility of delivering accurate, database-grounded answers under strict read-only governance on commodity on-premises hardware, while maintaining regulatory alignment with GDPR/AI-Act principles.

5.1 Summary of contributions

Methodologically, the dissertation contributes a practical design pattern for enterprise-grade, on-premises conversational analytics built on four mutually reinforcing pillars:

- **Governed tool-use via MCP.** The IVA uses MCP to discover and invoke a curated set of tools (schema inspection and SQL execution) within a scoped, auditable session. This decouples the LLM from direct database access and enables enterprise controls (least-privilege access; read-only validation; logging).
- **Runtime schema curation.** The MCP server enforces data minimisation by filtering away technical identifiers, binary payloads, and low-value attributes, imposes explicit projections (no `SELECT *`), and standardises casting/rounding to promote determinism. This reduces schema-hallucination, narrows the surface for accidental disclosure, and lightens the model's cognitive load.
- **Local LLM execution with Ollama.** Reasoning is executed entirely on-premises using a local runtime (Ollama), with decoding kept deterministic and context windows sized for reliable operation on commodity hardware; when a GPU is available, the runtime leverages accelerator memory while preserving the same governed behaviour. This preserves data sovereignty by eliminating external API calls and aligns with the privacy-preserving objectives.
- **Reproducible evaluation protocol.** The study isolates the effect of orchestration and curation by comparing three systems—B0 (prompt-only), B1 (MCP without filtering), and S (proposed MCP with filtering and safe projections)—on the same hardware, model, prompts, and procedure, with safety accounting and audit logs.

Beyond the implementation itself, the dissertation advances **evidence that governance-aware orchestration matters**: enabling tools materially improves robustness relative to prompt-only baselines, and adding curation (schema filtering, explicit projections) further reduces schema-related failures, while a validator blocks unsafe statements before execution—without sacrificing the information required to answer typical business questions.

5.2 Answers to the research questions

The SLR examined techniques and challenges spanning AI integration with BIS (ERP/CRM/MES), NLP for enterprise dialogue, ML/DL for integration and modelling, and barriers to IVA deployment. Its findings framed the design and evaluation in this work.

- **RQ1 (AI for integrating/modelling BIS data):** The literature shows increasing reliance on RAG, vector search, ontology-aware mediation, and tool-augmented orchestration. The present IVA operationalises these insights with an MCP layer that unifies schema discovery and execution under explicit governance, and with runtime schema curation to harmonise the information surface the model sees.
- **RQ2 (NLP methods for enterprise natural-language interaction):** Transformer-based assistants benefit from prompt scaffolding and tool-first behaviours. The proposed system codifies this by steering the model to inspect schema, draft explicit SQL with clear joins, and then request execution—reducing brittle, end-to-end "guessing" typical of prompt-only approaches.
- **RQ3 (ML/DL for integration and modelling):** Beyond predictive models, recent work emphasises agentic pipelines that transform heterogeneous artefacts into structured records and verifiable outputs. This dissertation implements a lightweight variant for relational settings: deterministic small-LLM reasoning coupled with server-side normalisation, yielding reproducible and inspectable results.
- **RQ4 (challenges for IVAs consolidating multi-source business data):** The SLR highlights privacy, interoperability, trust, and explainability as recurring barriers. The present approach mitigates these through on-premises execution, schema minimisation, read-only validators, and tool-level audit logs, thereby linking technical controls to governance requirements outlined earlier in the thesis.

5.3 Assessment against the dissertation objectives

The objectives formulated in Chapter 1 are revisited in this section and assessed against the experimental results presented in Chapter 4. Each objective is examined with direct reference to the outcomes reported in the evaluation, allowing a systematic verification of whether the dissertation's initial aims were achieved in practice.

- **O1 (IVA for natural-language access).** *Achieved.* The assistant consistently accepts plain-language prompts and returns validated, database-grounded tables across a stratified task suite of 52 queries spanning single-table filters, one-join aggregates, multi-join grouped analyses, and short trend explanations, as summarised in Tables 4.1 and 4.2. A representative interaction under configuration S follows the sequence prompt $\rightarrow$ schema inspection $\rightarrow$ internal SQL draft $\rightarrow$ validated execution $\rightarrow$ clean tabular answer.

- **O2 (Privacy and compliance on-premises).** *Achieved.* All experiments run locally on on-premises workstations with Microsoft SQL Server and a locally served LLM (Ollama), with internet access disabled. The MCP server mediates read-only access to the database and enforces the privacy-preserving objective established for the study through explicit contracts and schema filtering described in Chapter 4.

- **O3 (Accuracy and relevance).** *Partially achieved but promising.* On the full task suite (n=52), execution accuracy increases from 10% in the prompt-only baseline (B0) to 81% with MCP tools but no curation (B1), and to 92% with the proposed curated exposure and explicit-projection templates (S). Schema errors fall from 32 (B0) to 5 (B1) and to 0 (S). Remaining errors are concentrated in the hardest multi-join, grouped analyses, where ranking or aggregation mismatches persist, with the quantitative breakdown reported in Chapter 4.

- **O4 (Methodological balance of accuracy/efficiency/scalability).** *Achieved in scope.* MCP operations (schema inspection, validation, forwarding) and SQL execution run in real time, and overall responsiveness is dominated by model generation rather than orchestration overhead. This demonstrates interactive viability on commodity on-premises hardware (CPU, with optional GPU acceleration) while preserving the governed pipeline described in Chapter 4.

- **O5 (Feasibility of small LLMs).** *Achieved.* With conservative decoding and context limits suited to commodity on-premises hardware (typically 4,096–8,192 tokens), the locally hosted model—served through Ollama—delivers the accuracy gains noted above when combined with MCP tooling and schema curation, as detailed in Chapter 4. Where GPU memory permits, FP16/BF16 inference and KV-cache residency reduce latency; otherwise, 4–5 bit quantised variants preserve capacity on the CPU. This shows that useful, auditable answers do not require cloud-scale models under the studied workload.

- **O6 (Critically evaluate challenges and opportunities).** *Achieved.* Governance metrics show that the read-only validator blocks unsafe statements before execution (2/52 blocked events, 3.8%), while curated exposure reduces hallucinated-column incidents from seven (B1) to two (S). A blocked case with a clear client notice and no SQL leakage is illustrated in Fig. 4.5, and Chapter 4 synthesises residual risks (e.g., edge-case joins; dependence on curated attribute visibility) alongside mitigation paths.

The evaluation used a curated subset of AdventureWorksDW2022, a fixed small-LLM setup, and a compact prompt suite. These choices support internal validity and reproducibility but limit generalisation to larger, noisier enterprise schemas. Local hardware constraints bound context size and batchability: GPU memory saturation forces slower CPU fallbacks, while large database scans dominate execution time. The governance layer blocks unsafe SQL but may also hide attributes occasionally required, leading to residual failure modes.

For practitioners, the results support **protocol-first integration**: use the LLM as an orchestrator of governed tools, not a free-form generator, and keep schema exposure curated. This reduces risk, supports auditability, and aligns with **privacy-by-design** under European regulation. The demonstrated stack—SQL Server + MCP server + local LLM runtime—offers a practical, compliant blueprint for on-premises deployment.

5.4 Directions for future work

The study demonstrates the feasibility of deploying an on-premise intelligent virtual assistant for business data access, but it remains constrained by several deliberate simplifications. The evaluation was conducted on a single, carefully curated schema that avoids the complexity of large-scale enterprise environments. No persistent conversational memory was implemented, which limited continuity across interactions and precluded personalisation. Finally, hardware constraints restricted the experiments to bounded computational resources, which limited scalability and responsiveness under heavier workloads. Addressing these limitations opens several promising directions for future research and development:

- **Memory persistence.** Extend the assistant with privacy-preserving long-term memory mechanisms that retain conversational context across sessions. This would enable continuity, support personalisation of responses, and reduce the need for repeated schema rediscovery, while still respecting organisational data-governance constraints.
- **Broader schemas.** Move beyond a single curated dataset to encompass larger, noisier, and more heterogeneous enterprise databases. This includes handling multi-system joins, inconsistencies in naming conventions, and schema drift, thereby bringing the system closer to real-world deployment scenarios.
- **Smarter curation.** Investigate machine-learned schema linking, adaptive join planning, and selective schema exposure to reduce spurious matches and hallucinated attributes. Such techniques can cut down on query errors and improve both robustness and efficiency in schema-intensive environments.
- **Efficiency.** Explore optimisation techniques such as quantisation of language models, early-exit mechanisms, and context reuse to lower computational cost and reduce latency. These methods would enable faster responses without sacrificing the reliability of query execution.
- **Hybrid retrieval.** Combine relational database querying with knowledge graph lookups or embedding-based retrieval to enrich semantic coverage. A hybrid pipeline could integrate symbolic and sub-symbolic representations, offering deeper insights and supporting more complex analytical tasks.

5.5 Final Remarks

This dissertation set out to investigate how intelligent virtual assistants can be deployed in a privacy-preserving, on-premises environment to support business data access. By leveraging the MCP as an orchestration layer and combining it with lightweight large language models, schema filtering, and governance mechanisms, the proposed solution demonstrates that accurate, efficient, and regulation-compliant query answering is achievable without reliance on cloud-based or fine-tuned models.

The study contributes not only a practical implementation but also a methodological framework that balances technical feasibility with ethical and legal considerations, namely those imposed by the GDPR and the AI Act. While experiments were necessarily scoped to a limited set of datasets, tasks, and model configurations, the results provide evidence of the viability of resource-efficient IVA deployments in enterprise contexts.

Several avenues remain open for exploration. Future work should evaluate the assistant in larger and more heterogeneous databases, extend the task suite to cover more complex analytical queries, and incorporate human–computer interaction studies to assess usability and acceptance. Comparative analysis with larger foundation models, as well as hybrid cloud–edge deployments, would further clarify trade-offs in scalability and performance.

In conclusion, the dissertation demonstrates that protocol-driven orchestration of governed tools, combined with efficient local LLMs, offers a promising pathway towards trustworthy AI assistants for business information systems. By addressing both technical and governance requirements, this work contributes to advancing the responsible integration of AI into enterprise decision-making.

Bibliography

Ahaneku, Oguike, Maximilian Siegl, Sarah Stromberger, and Radu Vidrascu (Sept. 2023). "A Scalable AI-Driven Chatbot for Real-Time Diagnostics in Manufacturing Plants: Merging Google Dialogflow, BERT, and a Self-Learning Module". In: *2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*. Vol. 1, pp. 853–858. doi: 10.1109/IDAACS58523.2023.10348738.

Amendaño-Murrillo, Soa, Christian Dután-Gómez, Efrén Lema-Condo, and Vladimir Robles-Bykbaev (Oct. 2020). "Personal robotic assistants: a proposal based on the intelligent services of the IBM cloud and additive manufacturing". In: *2020 IEEE ANDESCON*, pp. 1–6. doi: 10.1109/ANDESCON50619.2020.9271971.

Anthropic (2024). *Model Context Protocol Specification*. https://modelcontextprotocol.io/specification/2024-11-05. Accessed: 17 August 2025.

Asan, Hakan and Vahap Tecim (July 2023). "Model Design and Development for a User Support System Using Artificial Intelligence Techniques in Enterprise Resource Planning Software". In: *INTERNATIONAL JOURNAL OF CONTEMPORARY ECONOMICS AND ADMINISTRATIVE SCIENCES* 13.2, pp. 836–864. issn: 1925-4423. doi: 10.5281/zenodo.10476230.

Barocas, S., M. Hardt, and A. Narayanan (2023). *Fairness and Machine Learning: Limitations and Opportunities*. Adaptive Computation and Machine Learning series. MIT Press. isbn: 9780262376525. url: https://books.google.pt/books?id=HuGwEAAAQBAJ.

Bender, Emily M., Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell (2021). "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? " In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. FAccT '21. Virtual Event, Canada: Association for Computing Machinery, pp. 610–623. isbn: 9781450383097. doi: 10.1145/3442188.3445922. url: https://doi.org/10.1145/3442188.3445922.

Binns, Reuben (23–24 Feb 2018). "Fairness in Machine Learning: Lessons from Political Philosophy". In: *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*. Ed. by Sorelle A. Friedler and Christo Wilson. Vol. 81. Proceedings of Machine Learning Research. PMLR, pp. 149–159. url: https://proceedings.mlr.press/v81/binns18a.html.

Bommasani, Rishi, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony

Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang (2022). *On the Opportunities and Risks of Foundation Models*. arXiv: `2108.07258 [cs.LG]`. url: `https://arxiv.org/abs/2108.07258`.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei (2020). *Language Models are Few-Shot Learners*. arXiv: `2005.14165 [cs.CL]`. url: `https://arxiv.org/abs/2005.14165`.

Budagov, A. (Oct. 2020). *Problems Of Effective Business Digital Transformation Management*. doi: `10.15405/epsbs.2020.10.03.48`.

Calo, Ryan (Jan. 2018). "Artificial Intelligence Policy: A Primer and Roadmap". In: *University of Bologna Law Review* 3.2, pp. 180–218. doi: `10.6092/issn.2531-6133/8670`. url: `https://bolognalawreview.unibo.it/article/view/8670`.

Choudhury, Musfiq Mannan and Paul Harrigan (2014). "CRM to social CRM: the integration of new technologies into customer relationship management". In: *Journal of Strategic Marketing* 22.2, pp. 149–176. doi: `10.1080/0965254X.2013.876069`. eprint: `https://doi.org/10.1080/0965254X.2013.876069`. url: `https://doi.org/10.1080/0965254X.2013.876069`.

Davenport, Thomas H. (Oct. 2018). "AI in the Enterprise". In: *The AI Advantage: How to Put the Artificial Intelligence Revolution to Work*. The MIT Press. isbn: 9780262350631. doi: `10.7551/mitpress/11781.003.0004`. eprint: `https://direct.mit.edu/book/chapter-pdf/2332373/9780262350631\_caf.pdf`. url: `https://doi.org/10.7551/mitpress/11781.003.0004`.

Davenport, Thomas H. and James E. Short (1990). "The New Industrial Engineering: Information Technology and Business Process Redesign". In: *Sloan Management Review* 31.4, pp. 11–27. url: `https://sloanreview.mit.edu/article/the-new-industrial-engineering-information-technology-and-business-process-redesign/`.

Davis, Fred (Sept. 1989). "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology". In: *MIS Quarterly* 13, pp. 319–. doi: `10.2307/249008`.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (June 2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. doi: `10.18653/v1/N19-1423`. url: `https://aclanthology.org/N19-1423/`.

Doering, Christopher (May 2024). *Coca-Cola turns to Microsoft's AI services for its supply chain | Supply Chain Dive*. url: https://www.supplychaindive.com/news/coca-cola-ai-artificial-intelligence-microsoft/714889/.

Doshi, Jash (Sept. 2021). "Chatbot User Interface for Customer Relationship Management using NLP models". In: *2021 International Conference on Artificial Intelligence and Machine Vision (AIMV)*, pp. 1–4. doi: 10.1109/AIMV53313.2021.9670914.

Doshi-Velez, Finale and Been Kim (2017). *Towards A Rigorous Science of Interpretable Machine Learning*. arXiv: 1702.08608 [stat.ML]. url: https://arxiv.org/abs/1702.08608.

Drucker, Peter F. (1969). *2 - The New Industries and their Dynamics*. Ed. by PETER F. DRUCKER. Butterworth-Heinemann, pp. 10–38. isbn: 978-0-434-90395-5. doi: https://doi.org/10.1016/B978-0-434-90395-5.50006-7. url: https://www.sciencedirect.com/science/article/pii/B9780434903955500067.

Ehtesham, Abul, Aditi Singh, and Saket Kumar (2025). "Enhancing Clinical Decision Support and EHR Insights through LLMs and the Model Context Protocol: An Open-Source MCP-FHIR Framework". In: *2025 IEEE World AI IoT Congress (AIIoT)*, pp. 0205–0211. doi: 10.1109/AIIoT65859.2025.11105280.

Element Labs, Inc. (2025). url: https://lmstudio.ai/app-privacy#lm-studio-desktop-app-privacy-policy.

European Parliament (2024). *Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act)*. Official Journal of the European Union, L 2024/1689. url: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32024R1689.

Forbes (2020). *How Amazon uses AI to integrate global operations*. Retrieved from https://www.forbes.com/.

Garcia, Cristian I., Marcus A. DiBattista, Tomás A. Letelier, Hunter D. Halloran, and Jaime A. Camelio (2024). "Framework for LLM applications in manufacturing". In: *Manufacturing Letters* 41. 52nd SME North American Manufacturing Research Conference (NAMRC 52), pp. 253–263. issn: 2213-8463. doi: https://doi.org/10.1016/j.mfglet.2024.09.030. url: https://www.sciencedirect.com/science/article/pii/S2213846324000920.

Garg, Anshumaan and Dolly Sharma (Nov. 2023). "Generative AI for Software Test Modelling with a focus on ERP Software". In: *2023 International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT)*, pp. 187–193. doi: 10.1109/ICAICCIT60255.2023.10466102.

Grant, Maria J. and Andrew Booth (2009). "A typology of reviews: an analysis of 14 review types and associated methodologies". In: *Health Information & Libraries Journal* 26.2, pp. 91–108. doi: https://doi.org/10.1111/j.1471-1842.2009.00848.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1471-1842.2009.00848.x. url: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1471-1842.2009.00848.x.

Gupta, Ruchi, Kiran Nair, Mahima Mishra, Blend Ibrahim, and Seema Bhardwaj (2024). "Adoption and impacts of generative artificial intelligence: Theoretical underpinnings and research agenda". In: *International Journal of Information Management Data Insights* 4.1, p. 100232. issn: 2667-0968. doi: https://doi.org/10.1016/j.jjimei.2024.100232. url: https://www.sciencedirect.com/science/article/pii/S2667096824000211.

Huang, Yifei (Aug. 2020). "Situation Awareness and Information Fusion in Sales and Customer Engagement: A Paradigm Shift". In: *2020 IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA)*, pp. 113–121. doi: 10.1109/CogSIMA49017.2020.9215990.

Hunter, J. D. (2007). "Matplotlib: A 2D graphics environment". In: *Computing In Science & Engineering* 9.3, pp. 90–95. doi: 10.1109/MCSE.2007.55.

Hüsson, Daniel, Alexander Holland, and Rocio Arteaga Sanchez (Nov. 2020). "Intelligent Personal Assistant in Business-Context: Key-feature Evaluation for User Acceptance". In: *BUSINESS SYSTEMS RESEARCH JOURNAL* 11.3, pp. 147–166. issn: 1847-8344. doi: 10.2478/bsrj-2020-0032.

Hüsson, Daniel, Alexander Holland, Madjid Fathi, and Rocío Arteaga Sánchez (2021). "Analysis and illustration of the practical impact of Artificial Intelligence and Intelligent Personal Assistants on business processes in small- and medium-sized service enterprises". In: *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 3303–3310. doi: 10.1109/SMC52423.2021.9659298.

Jin, Wenzhong, Yilan Sun, Kaiyuan Ji, Xiaoyan Jiang, Yufeng Hu, Jinwu Wang, and Jiannan Liu (2025). "MedScreenDental: Automated structured dental record generation via multimodal language model integration". In: *Displays* 90, p. 103119. issn: 0141-9382. doi: https://doi.org/10.1016/j.displa.2025.103119. url: https://www.sciencedirect.com/science/article/pii/S0141938225001568.

Jurafsky, Daniel and James Martin (2007). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Vol. 2. Prentice Hall, p. 1044.

Kern, Thomas, Julian Stang, Michael Milde, Andreas Hofer, Lasse Streibel, and Michael F. Zaeh (2024). "A modular chatbot framework for assisting workers in diverse production tasks". In: *Procedia CIRP* 130. 57th CIRP Conference on Manufacturing Systems 2024 (CMS 2024), pp. 1268–1275. issn: 2212-8271. doi: https://doi.org/10.1016/j.procir.2024.10.238. url: https://www.sciencedirect.com/science/article/pii/S2212827124013957.

Kletti, Jürgen (2007). *Manufacturing Execution Systems - MES*. Berlin, Heidelberg: Springer-Verlag. isbn: 3540497439.

Krishnareddy, Kr, T V Aravinda, Kiran Nair, Umesh Kumar Patel, Gaukhar Sadvokasova, and V Shyamala Susan (Nov. 2022). "AI-based Fuzzy Clustering System for Improving Customer Relationship Management". In: *2022 Sixth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pp. 673–677. doi: 10.1109/I-SMAC55078.2022.9987262.

Kuner, Christopher, Lee A Bygrave, Christopher Docksey, and Laura Drechsler (Feb. 2020). *The EU General Data Protection Regulation (GDPR): A Commentary*. Oxford University Press. isbn: 9780198826491. doi: 10.1093/oso/9780198826491.001.0001. url: https://doi.org/10.1093/oso/9780198826491.001.0001.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (May 2015). "Deep learning". In: *Nature* 521.7553, pp. 436–444. issn: 1476-4687. doi: 10.1038/nature14539. url: https://doi.org/10.1038/nature14539.

Li, Zhuoyan (Dec. 2024). "Review of Application of AI in Amazon Warehouse Management". In: *Advances in Economics, Management and Political Sciences* 144, pp. 1–8. doi: 10.54254/2754-1169/2024.GA18980.

Lipton, Zachary C. (June 2018). "The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery." In: *Queue* 16.3, pp. 31–57.

issn: 1542-7730. doi: 10.1145/3236386.3241340. url: https://doi.org/10.1145/3236386.3241340.

Maiya, Arun S. (2025). *OnPrem.LLM: A Privacy-Conscious Document Intelligence Toolkit*. arXiv: 2505.07672 [cs.CL]. url: https://arxiv.org/abs/2505.07672.

Makarius, Erin E., Debmalya Mukherjee, Joseph D. Fox, and Alexa K. Fox (2020). "Rising with the machines: A sociotechnical framework for bringing artificial intelligence into the organization". In: *Journal of Business Research* 120, pp. 262–273. issn: 0148-2963. doi: https://doi.org/10.1016/j.jbusres.2020.07.045. url: https://www.sciencedirect.com/science/article/pii/S0148296320305002.

Mantravadi, Soujanya, Charles Møller, Chen Li, and Reto Schnyder (2022). "Design choices for next-generation IIoT-connected MES/MOM: An empirical study on smart factories". In: *Robotics and Computer-Integrated Manufacturing* 73, p. 102225. issn: 0736-5845. doi: 10.1016/j.rcim.2021.102225. url: https://www.sciencedirect.com/science/article/pii/S0736584521001071.

Mark, P. and H. Roberts (2006). "Why Do We Need Systematic Reviews?" In: *Systematic Reviews in the Social Sciences*. John Wiley Sons, Ltd. Chap. 1, pp. 1–26. isbn: 9780470754887. doi: https://doi.org/10.1002/9780470754887.ch1. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470754887.ch1. url: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470754887.ch1.

Mark III Labs (2024). *MCPHost: CLI host application enabling LLMs to use MCP tools*. https://github.com/mark3labs/mcphost. Accessed October 2, 2025.

McAfee, Andrew and Erik Brynjolfsson (2017). *Machine, platform, crowd : harnessing our digital future / Andrew McAfee, Erik Brynjolfsson*. eng. New York: W. W. Norton Company. isbn: 9780393356069.

Microsoft (Aug. 2022). *Northwind and pubs sample databases for Microsoft SQL Server*. url: https://github.com/microsoft/sql-server-samples/tree/master/samples/databases/northwind-pubs.

– (May 2023). *Wide World Importers sample databases for Microsoft SQL*. url: https://learn.microsoft.com/en-us/sql/samples/wide-world-importers-what-is?view=sql-server-ver16.

– (Sept. 2024a). *AdventureWorks sample databases – SQL Server*. url: https://learn.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver16&tabs=ssms.

– (2024b). "Microsoft Dynamics 365 Copilot: AI-powered ERP and CRM Assistance". In: *Microsoft AI*. url: https://www.microsoft.com/pt-pt/microsoft-365/copilot/copilot-for-work.

– (Sept. 2024c). *Microsoft SQL samples – SQL Server*. url: https://learn.microsoft.com/en-us/sql/samples/sql-samples-where-are?view=sql-server-ver16.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (2013). *Efficient Estimation of Word Representations in Vector Space*. arXiv: 1301.3781 [cs.CL]. url: https://arxiv.org/abs/1301.3781.

Moher, David, Alessandro Liberati, Jennifer Tetzlaff, Douglas G. Altman, and The PRISMA Group (July 2009). "Preferred Reporting Items for Systematic Reviews and Meta-Analyses: The PRISMA Statement". In: *PLOS Medicine* 6.7, pp. 1–6. doi: 10.1371/journal.pmed.1000097. url: https://doi.org/10.1371/journal.pmed.1000097.

Motors, General (2023). "AI in Manufacturing: Predictive Maintenance at GM". In: *GM AI Blog*. url: https://www.gm.com/company/growth-strategy/artificial-intelligence.

Müller, Oliver, Maria Fay, and Jan vom Brocke (Mar. 2018). "The Effect of Big Data and Analytics on Firm Performance: An Econometric Analysis Considering Industry Characteristics". In: *Journal of Management Information Systems* 35. doi: 10.1080/07421222.2018.1451955.

Murphy, Kevin P. (2013). *Machine learning : a probabilistic perspective*. Cambridge, Mass. [u.a.]: MIT Press. isbn: 9780262018029 0262018020. url: https://www.amazon.com/Machine-Learning-Probabilistic-Perspective-Computation/dp/0262018020/ref=sr_1_2?ie=UTF8&qid=1336857747&sr=8-2.

Neumann, J. von (1993). "First draft of a report on the EDVAC". In: *IEEE Annals of the History of Computing* 15.4, pp. 27–75. doi: 10.1109/85.238389.

O'Leary, Daniel E. (2024a). "Large Language Models and Applications: The Rebirth of Enterprise Knowledge Management and the Rise of Prompt Libraries". In: *IEEE Intelligent Systems* 39.2, pp. 72–75. doi: 10.1109/MIS.2024.3366648.

– (Feb. 2024b). "The Rise and Design of Enterprise Large Language Models". In: *IEEE Intelligent Systems* 39.1, pp. 60–63. issn: 1541-1672. doi: 10.1109/MIS.2023.3345591. url: https://doi.org/10.1109/MIS.2023.3345591.

Ollama (2025a). url: https://ollama.com/search.

– (2025b). url: https://ollama.com/blog/thinking.

Ordonez, Carlos, Sofian Maabout, David Sergio Matusevich, and Wellington Cabrera (2014). "Extending ER models to capture database transformations to build data sets for data mining". In: *Data Knowledge Engineering* 89, pp. 38–54. issn: 0169-023X. doi: https://doi.org/10.1016/j.datak.2013.11.002. url: https://www.sciencedirect.com/science/article/pii/S0169023X13001298.

Ozay, Dervis, Mohammad Jahanbakht, Paul J Componation, and Atefeh Shoomal (Nov. 2023). "State of the Art and Themes of the Research on Artificial intelligence (AI) Integrated Customer Relationship Management (CRM): Bibliometric Analysis and Topic Modelling". In: *2023 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD)*, pp. 1–6. doi: 10.1109/ICTMOD59086.2023.10438124.

Page, Matthew J, Joanne E McKenzie, Patrick M Bossuyt, Isabelle Boutron, Tammy C Hoffmann, Cynthia D Mulrow, Larissa Shamseer, Jennifer M Tetzlaff, Elie A Akl, Sue E Brennan, Roger Chou, Julie Glanville, Jeremy M Grimshaw, Asbjørn Hróbjartsson, Manoj M Lalu, Tianjing Li, Elizabeth W Loder, Evan Mayo-Wilson, Steve McDonald, Luke A McGuinness, Lesley A Stewart, James Thomas, Andrea C Tricco, Vivian A Welch, Penny Whiting, and David Moher (2021). "The PRISMA 2020 statement: an updated guideline for reporting systematic reviews". In: *BMJ* 372. doi: 10.1136/bmj.n71. eprint: https://www.bmj.com/content/372/bmj.n71.full.pdf. url: https://www.bmj.com/content/372/bmj.n71.

Pennington, Jeffrey, Richard Socher, and Christopher Manning (Oct. 2014). "GloVe: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans. Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543. doi: 10.3115/v1/D14-1162. url: https://aclanthology.org/D14-1162/.

Pereira, Tiago F., Arthur Matta, Carlos M. Mayea, Frederico Pereira, Nelson Monroy, João Jorge, Tiago Rosa, Carlos E. Salgado, Ana Lima, Ricardo J. Machado, Luís Magalhães, Telmo Adão, Miguel Ángel Guevara López, and Dibet Garcia Gonzalez (2022). "A web-based Voice Interaction framework proposal for enhancing Information Systems user experience". In: *Procedia Computer Science* 196. International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement /

HCist - International Conference on Health and Social Care Information Systems and Technologies 2021, pp. 235–244. issn: 1877-0509. doi: https://doi.org/10.1016/j.procs.2021.12.010. url: https://www.sciencedirect.com/science/article/pii/S187705092102233X.

Prefect (2025). *Welcome to FastMCP 2.0!* https://gofastmcp.com/getting-started/welcome. Accessed: 2025-08-31.

Przegalinska, Aleksandra, Tamilla Triantoro, Anna Kovbasiuk, Leon Ciechanowski, Richard B. Freeman, and Konrad Sowa (2025). "Collaborative AI in the workplace: Enhancing organizational performance through resource-based and task-technology fit perspectives". In: *International Journal of Information Management* 81, p. 102853. issn: 0268-4012. doi: https://doi.org/10.1016/j.ijinfomgt.2024.102853. url: https://www.sciencedirect.com/science/article/pii/S0268401224001014.

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever (2019). "Language Models are Unsupervised Multitask Learners". In: *OpenAI*. Accessed: 2024-11-15. url: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu (2020). *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. arXiv: 1910.10683 [cs.LG]. url: https://arxiv.org/abs/1910.10683.

Sai, Brinda, Shachika Thanigaivelu, Shivaani N, Shyamala Babu C S, and Ramaa A (Dec. 2022). "Integration of Chatbots in the Procurement Stage of a Supply Chain". In: *2022 6th International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*, pp. 1–5. doi: 10.1109/CSITSS57437.2022.10026367.

Salesforce (2024). "Einstein AI: The AI Behind Smarter Customer Relationships". In: *Salesforce AI Research*. url: https://www.salesforce.com/eu/artificial-intelligence/.

SAP (2023). "AI in ERP: Enhancing Business Efficiency with SAP AI". In: *SAP AI Solutions*. url: https://www.sap.com/products/artificial-intelligence.html.

Sharma, Ruchi, Girish Lakhera, and Manu Sharma (Nov. 2023). "Artificial Intelligence in KM Processes: Emerging Trends and Challenges". In: *2023 1st DMIHER International Conference on Artificial Intelligence in Education and Industry 4.0 (IDICAIEI)*. Vol. 1, pp. 1–6. doi: 10.1109/IDICAIEI58380.2023.10406876.

Sharma, Shavneet, Gurmeet Singh, Nazrul Islam, and Amandeep Dhir (2024). "Why Do SMEs Adopt Artificial Intelligence-Based Chatbots?" In: *IEEE Transactions on Engineering Management* 71, pp. 1773–1786. issn: 1558-0040. doi: 10.1109/TEM.2022.3203469.

Siemens (2024). "MindSphere: Industrial AI and IoT Solutions". In: *Siemens AI Innovations*. url: https://www.siemens.com/global/en/products/automation/topic-areas/it-ot-convergence/siemens-iiot.html.

Sutton, Richard S. and Andrew G. Barto (1998). *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press.

Taylor, Frederick W. (1911). *The Principles of Scientific Management*. New York, NY: Harper & Brothers. url: https://archive.org/details/principlesofscie00taylrich.

TechCrunch (2021). *Tesla's AI supply chain challenges*. Retrieved from https://techcrunch.com/.

Tene, O. and Jules Polonetsky (Jan. 2014). "Privacy in the Age of Big Data: A Time for Big Decisions". In: *Stanford Law Review Online* 64, pp. 63–69.

Todoric, Andjela, Teodora Vuckovic, Rogério Dionisio, Dusanka Dakic, and Darko Stefanovic (June 2024). "Integrating Artificial Intelligence with Salesforce: A Literature Review". In:

2024 IEEE 22nd Mediterranean Electrotechnical Conference (MELECON), pp. 1122–1126. doi: 10.1109/MELECON56669.2024.10608758.

Tsimpoukelli, Maria, Jacob Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, and Felix Hill (2021). *Multimodal Few-Shot Learning with Frozen Language Models*. arXiv: 2106.13884 [cs.CV]. url: https://arxiv.org/abs/2106.13884.

Unni, Manu Vasudevan, S Rudresh, Bh Rashmi, K Renjith Krishnan, Rohit Kar, and S. Devichandrika (Dec. 2023). "Automation using Artificial Intelligence in Business Landscape". In: *2023 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS)*, pp. 1519–1523. doi: 10.1109/ICACRS58579.2023.10404252.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). *Attention Is All You Need*. arXiv: 1706.03762 [cs.CL]. url: https://arxiv.org/abs/1706.03762.

Voigt, Paul and Axel von dem Bussche (2017). *The EU General Data Protection Regulation (GDPR): A Practical Guide*. 1st. Springer Publishing Company, Incorporated. isbn: 3319579584.

Wachter, Sandra, Brent Mittelstadt, and Chris Russell (2018). *Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR*. arXiv: 1711.00399 [cs.AI]. url: https://arxiv.org/abs/1711.00399.

Wagner, Bret and Ellen Monk (2008). *Enterprise Resource Planning*. 3rd. Boston, MA, USA: Course Technology Press. isbn: 1423901797.

Waskom, Michael L. (2021). "seaborn: statistical data visualization". In: *Journal of Open Source Software* 6.60, p. 3021. doi: 10.21105/joss.03021. url: https://doi.org/10.21105/joss.03021.

Yao, Yifan, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang (2024). "A survey on large language model (LLM) security and privacy: The Good, The Bad, and The Ugly". In: *High-Confidence Computing* 4.2, p. 100211. issn: 2667-2952. doi: https://doi.org/10.1016/j.hcc.2024.100211. url: https://www.sciencedirect.com/science/article/pii/S266729522400014X.

Yiorgis Gozadinos (2023). *Oterm: Terminal client for Ollama with MCP tool support*. https://github.com/ggozad/oterm. Accessed October 2, 2025.

Youn, Seounmi and S. Venus Jin (2021). ""In A.I. we trust?" The effects of parasocial interaction and technopian versus luddite ideological views on chatbot-based customer relationship management in the emerging "feeling economy"". In: *Computers in Human Behavior* 119, p. 106721. issn: 0747-5632. doi: https://doi.org/10.1016/j.chb.2021.106721. url: https://www.sciencedirect.com/science/article/pii/S0747563221000431.

Yue, Cen April, Linjuan Rita Men, Renee Mitson, Donna Z. Davis, and Alvin Zhou (2024). "Artificial intelligence for internal communication: Strategies, challenges, and implications". In: *Public Relations Review* 50.5, p. 102515. issn: 0363-8111. doi: https://doi.org/10.1016/j.pubrev.2024.102515. url: https://www.sciencedirect.com/science/article/pii/S0363811124000948.

Zarsky, Tal (Jan. 2017). "Incompatible: The GDPR in the Age of Big Data". In: *The Seton Hall Law Review* 47.4, p. 2. url: https://scholarship.shu.edu/cgi/viewcontent.cgi?article=1606&context=shlr.

Zhang, Kaiyan, Jianyu Wang, Ermo Hua, Biqing Qi, Ning Ding, and Bowen Zhou (2024). *CoGenesis: A Framework Collaborating Large and Small Language Models for Secure Context-Aware Instruction Following*. arXiv: 2403.03129 [cs.CL]. url: https://arxiv.org/abs/2403.03129.