MongoDB Press

<packt>

Architectures for the Intelligent AI-Ready Enterprise

Building real-world solutions with MongoDB

Boris Bialek | Sebastian Rojas Arbulu | Taylor Hedgecock

Foreword by:
Jim Scharf,
Chief Technology Officer,
MongoDB, Inc.

Architectures for the Intelligent AI-Ready Enterprise

Building real-world solutions with MongoDB

Boris Bialek, Sebastian Rojas Arbulu, Taylor Hedgecock

‹packt›

Architectures for the Intelligent AI-Ready Enterprise

Portfolio Director: Sunith Shetty
Relationship Lead: Sathya Mohan
Project Managers: Aniket Shetty & Sathya Mohan
Content Engineers: Siddhant Jain, David Sugarman, & Divya Poliyath
Technical Editor: Aniket Shetty
Copy Editor: Safis Editing
Indexer: Pratik Shirodkar
Proofreader: Siddhant Jain
Production Designer: Deepak Chavan

First published: September 2025

Production reference: 1290825

Published by Packt Publishing Ltd.
Grosvenor House
11 St Paul's Square
Birmingham
B3 1RB, UK.

ISBN 978-1-80611-715-4
www.packtpub.com

Foreword

If you had told me a few years ago that I'd be writing the foreword to a book on AI, I might've raised an eyebrow. And yet, here we are, right in the middle of one of the fastest-moving technological shifts in modern history.

Unlike prior technological shifts, what is different this time is the pace. Change is not unfolding over years; it is happening in months, weeks, sometimes even days.

The question facing every organization is not whether AI will reshape their business; it is how fast they can adapt and whether they will lead or fall behind.

Throughout my career, I have seen what separates the companies that thrive during moments like these from those that fall behind. It is rarely just the technology. It is the mindset—the willingness to rethink how you operate, how you deliver value, and how you use data to drive better outcomes. AI, especially generative and agentic systems, demands exactly that kind of rethink.

At MongoDB, we have worked closely with thousands of organizations across various industries, including financial services, healthcare, insurance, retail, and manufacturing, as they navigate their AI journeys. And the pattern is clear: the companies that succeed do not start with algorithms or models. They start with the foundation. That foundation is the **data**.

Consider a major financial institution that transitioned from traditional batch-based fraud detection to real-time monitoring using MongoDB Atlas, reducing rollout times from weeks to minutes and saving millions annually. Or consider a global pharmaceutical company that used MongoDB and generative AI to cut clinical report generation from twelve weeks to just ten minutes. These AI-enabled systems were transformational for both businesses and were made possible by modern, flexible infrastructure built to support intelligent workloads at scale.

The lesson is clear: if you want to implement AI successfully, your data foundation has to come first.

That is why this book matters.

It doesn't just talk about what is possible with AI; it shows you **how to make it a reality**. You will find hard-earned lessons from those already deploying AI in production, plus architectural blueprints and implementation strategies that you can apply immediately across any industry.

You will also learn why document-based data models are quickly becoming central to AI applications, how vector search unlocks meaning from unstructured data, and why blending operational and analytical capabilities creates real architectural leverage.

Just as important, this book addresses the often-overlooked realities of trust, governance, security, and scale. Building powerful AI is one thing. Making it responsible, resilient, and production-ready is something else entirely.

Whether you are a technology leader, architect, engineer, data scientist, or anyone responsible for implementing AI in your organization, this book offers a clear, practical path forward.

I am excited to see what you will build next.

Jim Scharf

Chief Technology Officer,

MongoDB, Inc.

Note from the author

The advent of **large language models** (**LLMs**), predicated on transformer-based architectures, began in 2018. At the same time, advancements in GPU technology facilitated greater parallel computational capabilities, culminating in the Nvidia V100 and catalyzing the **generative AI** (**GenAI**) domain. Natural language processing moved beyond reliance on rule-based systems and narrow subject area training, benefiting from comprehensive internet data and yielding significant advancements. The 2022 public beta release of ChatGPT, along with the introduction of competing LLMs, most notably Google Gemini and Anthropic Claude (utilized in AWS Bedrock), precipitated a shift in software development paradigms. Software design, coding, and business use cases expanded in unprecedented ways.

My preliminary investigations into vector utilization for semantic search and enhancements to MongoDB's existing text search functionality predated these developments. The domains of embeddings, models, and nearest neighbors converged with the emerging field of LLMs. This convergence enabled the rapid development of a preliminary **retrieval-augmented generation** (**RAG**) solution, which facilitated the interpretation of PDF documents and the generation of responses based on natural language queries and vectorized document data. The MongoDB Industry Solutions team recognized the innovative potential and expedited implementation through the MongoDB Atlas platform. The advantages of an integrated platform over multi-component systems were substantial. However, the proliferation of components and permutations created challenges, particularly the overreliance on singular, nascent components. Consequently, the focus shifted toward solution design with specific business outcomes, referred to as the *art of the possible*. Client requests increasingly emphasized implementation details (*how*) rather than conceptualization (*what*).

During 2023 and 2024, reference architectures and established designs were developed, warranting broader dissemination. Solutions originating in specific use cases demonstrated cross-industry applicability. The concept of the consolidated data store was refined and required further documentation.

In the summer of 2025, a directive was issued to the team to compile industry-specific solution designs, based on current agentic AI patterns and templates, into a comprehensive publication. Foundational chapters were subsequently incorporated to accommodate varied experience levels.

Collaboration with strategic business partners has provided diverse perspectives, enriching the content. The insights of the CXO Advisory team, focused on application modernization and the use of generative AI tools for legacy system enhancement, have also been incorporated.

It is anticipated that this publication will serve as a valuable resource for those interested in industry solutions leveraging GenAI.

Boris Bialek

Vice President and Field CTO,

MongoDB, Inc.

Acknowledgements

Every book has its origin story, and ours began with a simple yet overwhelming challenge. As MongoDB's Industry Solutions team, we specialize in presenting MongoDB as a solution for specific industries. We speak our customers' language and understand their industry needs, roadblocks, market trends, and competitors.

Over the years, we have spent countless hours documenting industry solutions across our blogs, the MongoDB solution library, and numerous articles and presentations that we have passionately created to help developers and organizations solve their most challenging data problems. But when clients asked us to help them navigate this wealth of knowledge, we faced an uncomfortable truth: having thousands of scattered links, no matter how valuable each one might be, had become overwhelming rather than helpful.

It was Raghu Viswanathan, our remarkable leader in education and documentation, who identified the opportunity we hadn't yet seen. In a conversation that began as a brainstorm about how to better serve our community, he suggested something that felt both obvious and audacious: "*Why not turn all these insights into a book?*" Without his clarity and persistent encouragement, this project would have remained nothing more than a collection of good intentions instead of becoming something real.

Writing a book in the technology space is never a solo endeavor. This is especially true when discussing real-world architectures and solutions. We are deeply grateful to the teams at Iguazio (acquired by QuantumBlack, AI by McKinsey), Fireworks AI, Dataworkz, Encore, Cognigy, and RegData, whose real-world implementations and feedback helped us understand what actually works in practice. Equally important are our technology partners such as Amazon Web Services, Microsoft Azure, Google Cloud, Confluent, Capgemini, and others. Their platforms and expertise make the solutions we discuss possible.

We would also like to thank our publishing partners at Packt, who proved that the best collaborations happen when expertise meets passion. Through countless revision cycles, they pushed us to think about our readers at every step. They asked the hard questions that helped transform our technical expertise into something both authoritative and accessible.

And to our colleagues across Industry Solutions, Product, and other internal teams, you know who you are. Your fingerprints are on every chapter.

This book is for every builder, architect, and strategist working to solve what comes next with AI. It is for you.

Contributors

About the authors

Boris Bialek has worked in the IT industry since the 1990s and was one of the initial drivers of Linux in Europe, delivering the first SAP port to Linux, conducting the first benchmarks, and securing the first clients. Since then, he has led product and development teams across IBM and FIS, driving innovation for both the end product and development productivity. Boris Bialek joined MongoDB in 2019, igniting a focus on industry solutions based on MongoDB's document model. Promoted to global field CTO and VP of industries, he drives technical design. He works directly with numerous clients, helping them gain the benefits of the MongoDB Atlas data platform. Boris holds a master's in computer science from the Karlsruhe Institute of Technology.

Sebastian Rojas Arbulu is an industry solutions specialist at MongoDB, where he collaborates with numerous stakeholders across diverse industries to help customers realize the transformative value of MongoDB through tailored, data-driven solutions, particularly for AI integration. Sebastian also leads his team's content strategy, including numerous additions such as blogs, white papers, magazines, and other thought leadership pieces. With a background in IT consulting, marketing, and digital transformation, among other areas, he has extensive experience in identifying customer needs and developing innovative solutions that prepare data for intelligent applications and unlock new possibilities. He holds a bachelor of business administration degree.

Taylor Hedgecock is a strategic program leader and transformation partner who turns vision into velocity. With a career spanning startups to multinationals, she brings a mix of operational rigor, narrative clarity, and cross-functional orchestration. At MongoDB, she has led high-impact programs across AI, partner ecosystems, and services modernization, often serving as the connective tissue between vision and execution. Her work has guided C-level priorities, enabled go-to-market readiness, and driven large-scale change, establishing her as a trusted leader in aligning stakeholders, translating strategy into story, and driving outcomes that last. Taylor currently serves as senior program manager on the industry solutions team, partnering with ISVs and AI innovators to bring next-generation solutions to market. Previously, she was chief of staff for professional services leadership, where she helped launch new offerings and guided modernization strategy, shaping MongoDB's vision for applying AI to its hardest problems.

Benjamin Lorenz has been a key contributor to MongoDB since 2016, driving growth across the Central European sales region. With deep expertise in strategic customer initiatives, he partners with decision-makers to align tailored solutions with business goals—leveraging the power of MongoDB's developer data platform. As industry solutions principal for telco & media, Benjamin guides global clients through digital transformation, helping them unlock innovative, data-driven revenue streams.

Francesc Mateu is a principal of industry solutions at MongoDB, with 20+ years in B2B SaaS and IT innovation, including 15 years in digital health. As a startup founder and product leader, he possesses an entrepreneurial mindset aimed at helping healthcare organizations modernize their data architectures. His work includes designing digital platforms that support patient-centered care and value-based models, including telemedicine solutions for capturing patient-reported outcomes. At MongoDB, he works globally with sales, partners, and product teams to help healthcare systems adopt AI-ready, standards-based architectures—leveraging technologies such as FHIR and openEHR—to create tailored solutions to meet each organization's unique needs.

Genevieve Broadhead is the global lead for retail solutions at MongoDB, based in Barcelona. She helps global retailers and retail software companies modernize data architectures for real-time personalization, omnichannel experiences, and AI-powered operations. With a computer engineering degree from Trinity College Dublin and a decade of experience in system design, Genevieve has extensive experience bridging business needs with cutting-edge technology. A recognized thought leader, she speaks regularly on cloud-native data pipelines, AI adoption, and composable commerce. She also serves on the MACH Alliance tech council, shaping standards for modern retail architectures.

Dr. Humza Akhtar is the smart manufacturing and automotive expert at MongoDB. Prior to joining MongoDB, he worked at Ernst & Young Canada in its digital operations consultancy practice. After completing his education in Singapore, he worked in the Singapore manufacturing industry for many years on Industry 4.0 research and implementation. He has spent his entire career enabling connected factories and connected cars for global manufacturing and automotive clients. He is a published author on Industry 4.0, and these days, his interest lies in enabling the use of generative AI within the automotive sector. Humza holds a master's degree in embedded systems and a doctorate in computer science from Nanyang Technological University, Singapore.

Jeff Needham is MongoDB's insurance industry expert, with nearly 30 years of experience in software delivery. As former senior director of architecture at Travelers, he led one of the industry's most successful MongoDB adoptions. His career spans leadership at major software companies and healthcare giants such as Aetna/CVS. Jeff's technical expertise and strategic insight drive exceptional outcomes for MongoDB's complex enterprise engagements, helping organizations navigate digital transformation. He holds a master's in political strategy from George Washington University.

Ken Wiebke has been working in the software development industry for over 30 years, with a career spanning development, architecture, and leadership. Throughout his career, Ken has driven change in organizations ranging from small to Fortune 500, including the shift from waterfall to agile. Serving in leadership roles for over 15 years, Ken's focus has been on driving efficiencies and building high-performance teams that consistently deliver on time and within budget. Ken joined MongoDB in December 2024 as a CxO advisor to help organizations leverage the power of MongoDB and transform their legacy software to modern tech stacks.

Luis Pazmino Diaz holds over two decades of experience in the technology sector, particularly within banking and finance. Previously, he served as global strategy architect for Backbase, director of innovation at Temenos, and solutions advisor at major enterprise software firms such as SAP and Oracle. As MongoDB's industry principal for financial services, he delivers strategic guidance to clients and solution partners across Europe, the Middle East, and Latin America. Based in Madrid, Spain, Luis has been widely recognized as a financial innovation expert.

Peyman Parsi began his career in financial services software engineering at SS&C, focusing on building wealth management software for the banking industry. In 2001, he joined the Toronto Stock Exchange (TSX), leading the development of capital markets solutions. Over 18 years at TSX, Peyman delivered several large-scale transformations and held the position of chief technology delivery officer. In 2020, he embarked on a new journey in FinTech, serving as CTO at Blanc Labs, with a primary focus on banking and digital lending solutions. Peyman is a member of the advisory board of the CIO Association of Canada and joined MongoDB in 2024 as senior principal of financial services industry solutions for the Americas.

Prashant Juttokonda is an expert in enterprise data architecture and modernization at MongoDB. Previously, he held leadership roles at EPAM, TCS, and IBM, advising global clients on cloud adoption, data transformation, and AI-ready architectures. With over 30 years of experience across banking, retail, and energy sectors, he has led large-scale modernization programs and driven the adoption of frameworks such as Data Mesh and Data Fabric. He is a frequent speaker and published thought leader in data strategy. His focus is on enabling generative AI and resilient data platforms. He holds a B.Sc. in mathematics and numerous certifications in cybersecurity and cloud technologies.

Raphael Schor is a mechanical engineer with 20+ years of experience in mechanical development, plant engineering, and industrial maintenance. He has served as CTO in the automotive and packaging industries, leading R&D and digital transformation. Since 2023, Raphael has served as principal for manufacturing and motion at MongoDB. In this role, he bridges the gap between industrial engineering challenges and modern data architecture. His focus includes digital twins, smart factories, and generative AI. Raphael holds a bachelor's in mechanical engineering and a Master of Advanced Studies in Management, Technology, and Economics from ETH Zurich.

Rodrigo Leal, with over 20 years in the technology industry, is the principal retail industry solutions for Latin America at MongoDB. Prior to MongoDB, Rodrigo served as a senior principal solution specialist at Qualtrics, enhancing employee, customer, brand, and product experiences. Earlier, he was part of NCR's Walmart Global Team, playing a role in launching self-checkout technology in Mexico and Central America, and was recognized with two NCR President's Club awards. He previously worked with Oracle and MicroStrategy as an account manager and sales engineer. Known for strengths in consultative multi-product selling, he is dedicated to uncovering customer needs and crafting solutions that often reveal previously unseen opportunities.

Thorsten Walther boasts over 25 years in tech leadership, blending enterprise expertise, entrepreneurial drive, and deep technical acumen to drive digital transformation. He founded and led INSPIFY, an AI-powered SaaS platform for luxury retail. His career spans leadership roles at Credit Suisse and SOFGEN Services, as well as extensive advisory work in finance, retail, pharmacy, and enterprise software. Currently, as managing director, CXO advisory for Asia at MongoDB, he guides senior executives on digital transformation. Uniquely, Thorsten was a professional footballer in Germany's Bundesliga and France's Ligue 1 and 2. He holds an MBA from the University of Liverpool.

Wei You Pan is the global director of financial services industry solutions at MongoDB. With over 25 years spanning fintech, data architecture, and financial services, he empowers institutions to overcome complex data challenges and drive innovation. His expertise includes trading, loan origination, risk management, and sustainability, supported by credentials in enterprise architecture (SCEA), financial risk management (FRM), and climate risk (SCR). His cross-disciplinary background enables him to uniquely bridge technology and business, helping organizations realize the full value of their data.

About the reviewers

Coral Parmar serves as lead product manager on MongoDB's search portfolio, bringing more than 20 years of technology and systems experience to help developers navigate modern data challenges across diverse verticals. His career spans leadership roles in MongoDB technical services, AdTech companies, and data development at UPS, providing deep expertise in scaling customer solutions for complex data systems across supply chain, logistics, and advertising technologies. Throughout his career, he has maintained a passion for solving complex problems and empowering teams to build effective, scalable solutions in rapidly evolving technical landscapes. He holds a master's in information systems from New Jersey Institute of Technology.

James Osgood is a staff solutions architect at MongoDB with over 30 years of experience spanning software development and financial services. He began his career in the audio and video industry before moving into financial services, where he specialized in low-latency trading and market surveillance systems. Since joining MongoDB in 2017, James has partnered with major customers across London and Europe. Today, his focus is on helping global enterprises transform mission-critical financial systems and modernize broader application estates with purpose-built AI-driven modernization solutions.

Jim Blackhurst is a distinguished solutions architect at MongoDB, with more than 20 years of experience designing and delivering distributed data systems. He currently works with MongoDB's application modernization team, helping some of the world's largest organizations modernize their estates through purpose-built AI modernization tools, liberating them from the grip of legacy technologies. Before joining MongoDB, Jim spent his career in the video game industry, architecting and operating backend systems for some of the most iconic global gaming brands, working with data at scale before "scale" became a thing. Based in London, Jim continues to focus on pushing the boundaries of distributed systems design and helping enterprises unlock new possibilities with data.

Julia Pak is a product manager at MongoDB on the enterprise initiatives and tools team. She currently focuses on developing internal products that enhance organizational productivity through data centralization and AI-powered features. Prior to joining MongoDB, Julia worked in the insurance and advertising technology industries, building external products from the ground up. She earned her bachelor of arts degree in history from Princeton University.

Shash Thakor is a senior product manager at MongoDB, primarily responsible for Atlas networking. He has 15+ years of product and engineering leadership experience in developing highly distributed, scalable, and secure software systems. Before moving into product management, he was a software developer with Cisco Systems and Juniper Networks, responsible for developing switching and routing software systems deployed in many data centers across the world and used by hyperscalers, big enterprises, and small businesses. He has multiple patents in distributed systems, security, and zero-touch provisioning. He holds a master's in computer science from the University of Maryland, College Park (UMCP).

Table of Contents

Chapter 3: The System of Action 45

Preface

This book is about how organizations can move beyond surface-level AI adoption and implement AI as a true driver of business transformation. It explains the strategic importance of distinguishing between modernization and innovation, and how both are essential for successful AI deployment. Through real-world implementations, success stories, and practical frameworks, it provides a roadmap for navigating the AI inflection point, aligning data infrastructure with AI goals, and building trustworthy, scalable, and context-aware AI systems.

The book is organized into three parts. The first part lays the foundation, covering core AI concepts, system architectures, governance, and modernization approaches that prepare organizations for large-scale adoption. The second part explores industry applications, showing how agentic and **generative AI** (**GenAI**) can reshape sectors such as manufacturing, media, retail, financial services, insurance, and healthcare. The final part looks ahead, presenting advanced implementation patterns, governance models, and emerging technologies such as **Model Context Protocol** (**MCP**) and causal AI, equipping readers with strategies to sustain innovation and adapt to the next wave of intelligent systems.

How this book will help you

Inside, you will learn the core patterns for building intelligent architectures, with a focus on GenAI, **retrieval-augmented generation** (**RAG**), and agentic systems powered by AI agents. You will see how these capabilities are applied across industries, supported by mapped reference architectures and detailed implementation guidance. The book also explores emerging directions such as causal intelligence, MCP, and advanced multi-agent design patterns. Whether you are modernizing legacy infrastructure or creating new platforms, it equips you with the vocabulary, frameworks, and practical context to move faster, reduce guesswork, and build reliable, scalable, and context-aware AI systems.

Who this book is for

This book is for:

- **IT decision-makers** exploring where to place strategic AI bets
- **Enterprise and solution architects** rethinking their data and application stack
- **Technical ears and curious builders** who want to understand how intelligent systems are structured and deployed
- **Business strategists and domain owners** seeking to translate AI hype into domain-specific outcomes

You don't need deep AI experience to get value from this book, but you should feel comfortable thinking in terms of data systems, application layers, and business architecture. If you're already experienced with AI concepts covered here, feel free to skip the early chapters and jump into the real-world case studies and future-focused content.

What this book covers

Chapter 1, *AI Modernization to Innovation*, outlines the difference between modernization and true innovation and how to structure teams, data, and processes to turn AI experiments into business outcomes.

Chapter 2, *What Sets GenAI, RAG, and Agentic AI Apart*, defines GenAI, RAG, and agentic systems, and explains when to use each approach.

Chapter 3, *The System of Action*, describes the document-oriented system of action and why unified, low-latency access to multimodal data and embeddings is critical for AI workloads.

Chapter 4, *Trustworthy AI, Compliance, and Data Governance*, summarizes governance, privacy, explainability, and risk management practices required for production AI.

Chapter 5, *Modernization Using AI*, gives practical patterns for using AI to accelerate legacy modernization while preserving correctness and governance.

Chapter 6, *Practical Applications of Agentic and GenAI in Manufacturing – Part 1*, focuses on supply-chain and inventory use cases, including embedding-driven classification and autonomous procurement helpers.

Chapter 7, *Practical Applications of Agentic and GenAI in Manufacturing – Part II*, focuses on factory-floor operations, including predictive maintenance, quality inspection, and multi-agent production orchestration.

Chapter 8, *AI-Driven Strategies for Media and Telecommunication Industries*, covers personalization, search experiences, AI operations, and fraud detection tailored to media and telecom sectors.

Chapter 9, *Cognigy's Voice and Chatbots in the Time of Agentic AI*, examines voice and chat systems for high-throughput, goal-oriented customer interactions.

Chapter 10, *Harnessing AI to Transform the Retail Industry*, explains personalization, demand forecasting, inventory optimization, and real-time decision-making in retail.

Chapter 11, *Financial Services and the Next Wave of AI*, outlines the sector's next AI transformation, from customer insight and compliance automation to AI-enhanced risk management and service models.

Chapter 12, *RegData, MongoDB, and Voyage AI: Semantic Data Protection in FSI*, describes semantic protection and audit approaches that enable compliant use of **large language models** (**LLMs**) in finance.

Chapter 13, *Driving Client Success in Banking with GenAI Copilots*, shows how banking copilots can automate advisor tasks, surface research, and support compliant client communications.

Chapter 14, *Delivering Business Value with AI in Insurance*, outlines converged datastores and AI patterns to improve underwriting, claims, and customer outcomes.

Chapter 15, *Automating Insurance Underwriting with Fireworks AI and MongoDB*, details an end-to-end underwriting pipeline using retrieval-grounded AI for faster, more accurate policy intake and quoting.

Chapter 16, *AI-Powered Transformation of Healthcare and Life Sciences*, addresses clinician overload with FHIR facade patterns, clinical RAG, and multi-agent care coordination to achieve better patient outcomes.

Chapter 17, *Enterprise Document Management with MongoDB and AI*, demonstrates how to turn unstructured enterprise *dark* data into enriched, searchable knowledge for operational and AI use.

Chapter 18, *Democratizing Agentic AI for Enterprise with Dataworkz and MongoDB*, provides architectural guidance, governance practices, and real-world cases for deploying safe, observable, and effective agentic AI.

Chapter 19, *Outlook: Beyond Today's AI*, looks ahead to MCP, memory-driven agents, and causal AI as drivers of the next wave of intelligent systems.

To get the most out of this book

No specific tooling expertise is required, though a working understanding of enterprise systems and data architecture will help you engage more deeply with the material. Readers interested in implementation details, can explore:

- MongoDB Solutions Library: `https://www.mongodb.com/docs/atlas/architecture/current/solutions-library/`
- MongoDB for Artificial Intelligence: `https://www.mongodb.com/solutions/use-cases/artificial-intelligence`

Download the color images

We also provide a PDF file that has color images of the screenshots/diagrams used in this book. You can download it here: `https://packt.link/gbp/9781806117154`.

Conventions used

There are a number of text conventions used throughout this book.

`CodeInText`: Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input or prompts, and Twitter handles. For example: "In the relational model, fields use names such as `FIRST_NAME`."

A block of code is set as follows:

```
{
 "_id": "67c20cf886f35bcb8c71e53c",
 "agent_id": "default_agent",
 "profile": "Default Agent Profile",
 "instructions": "Follow diagnostic procedures meticulously.",
 "rules": "Ensure safety; validate sensor data; document all steps.",
 "goals": "Provide accurate diagnostics and actionable recommendations."
}
```

Bold: Indicates a new term, an important word, or words that you see on the screen. For example: "The terms **prompting** and **prompt engineering** are frequently used in the same breath as LLMs and GenAI."

Warnings or important notes appear like this.

Tips and tricks appear like this.

Customer stories and other real-world examples appear like this.

Get in touch

Feedback from our readers is always welcome.

General feedback: If you have questions about any aspect of this book or have any general feedback, please email us at `customercare@packt.com` and mention the book's title in the subject of your message.

Errata: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book, we would be grateful if you reported this to us. Please visit `http://www.packt.com/submit-errata`, click **Submit Errata**, and fill in the form.

Piracy: If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at `copyright@packt.com` with a link to the material.

If you are interested in becoming an author: If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit `http://authors.packt.com/`.

Part 1:
AI and Key Concepts

The following set of chapters lays the conceptual and architectural foundation for building intelligent systems with generative and agentic AI. It introduces the shift from modernization to innovation. It also explains the importance of real-time, document-based data models, and it describes the new architectural thinking required to implement AI at scale in a secure and responsible way.

This part of the book includes the following chapters:

- *Chapter 1, AI Modernization to Innovation*
- *Chapter 2, What Sets GenAI, RAG, and Agentic AI Apart*
- *Chapter 3, The System of Action*
- *Chapter 4, Trustworthy AI, Compliance, and Data Governance*
- *Chapter 5, Modernization Using AI*

1

AI Modernization to Innovation

Many readers are at different stages of their AI journey, from initial exploration to active implementation planning. Some are leading teams tasked with *doing something with AI*. Others are watching competitors announce AI initiatives while wondering what their next move should be. Still others have tried AI pilots that showed promise in demos but somehow never made it to production. If any of this sounds familiar, you've found the right resource.

While countless publications theorize about AI's promise, this book takes a different approach: it's a field guide built *by* practitioners *for* practitioners, designed to help you navigate past the endless testing phase and move toward real-world AI implementation at scale.

Across industries, we've gathered hard-won lessons from teams that have actually done the work. These aren't abstract frameworks or vendor pitches; they're playbooks forged in the fire of enterprise constraints, integration realities, and performance expectations. We also look beyond the buzzwords. Though we cover **generative AI** (**GenAI**) and AI agents, we also discuss the data architectures, governance models, and design patterns that make this technology actually work. Because in the end, AI transformation isn't just about adopting new tools; it's about rethinking how your systems, teams, and business strategy fit together.

This practical focus matters now more than ever. While the AI revolution promises unprecedented productivity growth potential, the reality is that most organizations remain trapped in a cycle of modernization without innovation. Nearly all companies are investing in AI technologies, but few have integrated them deeply enough to deliver game-changing results. The difference lies in their approach [1].

How do you close this gap? It starts with distinguishing between two concepts that often get conflated: **modernization** and **innovation**. In this chapter, we'll untangle those definitions and show why understanding the difference is strategic. You'll learn how modernization sets the stage for AI success, how innovation extends its reach, and how both become essential when navigating a moment of rapid disruption.

At the end of this chapter, you will walk away with a sharper grasp of the core dynamics shaping successful AI adoption, including:

- The fundamental differences between innovation and modernization, and why both matter in AI deployments
- How Andy Grove's theory of strategic inflection points applies to today's AI revolution
- Why modernizing legacy systems and data infrastructure is essential for successful AI implementation
- How to avoid common pitfalls organizations face when pursuing AI initiatives without proper foundations
- How to implement practical approaches to balancing innovation and modernization in your AI strategy

Understanding innovation: Creating new value

Before you can effectively harness AI to modernize or innovate, you need a practical understanding of what these terms actually mean and how they differ in both execution and impact.

Innovation, at its core, *is the process of creating and implementing new ideas, methods, products, or services that add value or improve upon existing ones*. It involves turning creative concepts into practical solutions that meet real-world needs or solve problems more effectively.

Innovation can take many forms, including the following:

- **Product innovation**: Developing new or significantly improved goods or services. Think of the move from paper maps to GPS apps.
- **Process innovation**: Introducing new ways of producing or delivering products. **Robotic process automation (RPA)** has revolutionized standardized processes and decision-making, leading to innovations such as autonomous underwriting or real-time claims management for insurance companies.
- **Business model innovation**: Redefining how a company creates and captures value. Probably the most famous of all innovations was Netflix's multi-level innovation, first shipping DVDs to people's homes, rather than *picking them up at Blockbuster*, and then delivering content via streaming to make it even easier than mail.

- **Social innovation**: Using new technologies to address environmental and societal needs. AI is accelerating **environmental, social, and governance** (**ESG**) progress. This includes everything from smart energy systems to automated carbon tracking. In places such as Brazil, platforms such as PicPay are expanding access to financial tools and social programs for underserved communities.

Each of these examples started with an idea, but became an innovation only once it was applied to solve a problem at scale. In this sense, innovation requires more than invention. It requires execution.

Strategic inflection points: Andy Grove's theory applied to AI

Few frameworks better explain what's at stake in today's AI race than Andy Grove's theory of strategic inflection points, introduced in his book *Only the Paranoid Survive*. Grove describes these moments as times when the fundamentals of a business (or an entire industry) undergo dramatic, irreversible change. These shifts can be triggered by new technology, regulatory upheaval, competitive pressure, or all three at once.

Grove's central insight? *You don't see an inflection point clearly until you're in it*. And by then, it's often too late to catch up. Companies that adapt early can leap ahead. Those that hesitate, resist, or cling to old models often don't survive.

Strategic inflection points require leadership to do more than just optimize; they demand reinvention. Grove famously said: "*Only the paranoid survive.*" In his view, success requires constant vigilance, relentless questioning of assumptions, and the courage to bet on transformation before the path is proven.

The best-known example? Intel.

In the 1980s, Intel dominated the memory chip business. But competitors from Japan began producing faster, cheaper, and higher-quality alternatives. Grove and then-CEO Gordon Moore made a bold call: they abandoned their legacy business and pivoted entirely to microprocessors. At the time, this market was small and uncertain. But the gamble paid off. Intel's chips became the backbone of the personal computing revolution, transforming the company into one of the most important tech players of the modern era.

Today, AI presents a similar inflection point. The introduction of **large language models** (**LLMs**), **retrieval-augmented generation** (**RAG**), and agentic systems may prove to be as foundational to the AI era as the first microprocessors were to the PC era. These technologies are not just new features; they're new substrates for how business is done.

Unlike the microprocessor revolution, which unfolded over decades, AI is evolving at unprecedented speed. Expectations are rising faster than infrastructure can keep up. And many organizations haven't even started modernizing their foundations.

If you're feeling the urgency, you should be. This is what an inflection point feels like.

Navigating the AI inflection point

Organizations that recognize this moment as a true inflection point can prepare their infrastructure and capabilities for the AI-driven future. Beyond new tools, success demands a fundamental rethink of how data, technology, and business processes work together.

At the core of this transformation are five critical capabilities:

- **Flexible, future-ready data infrastructure**: Legacy systems face significant challenges with AI requirements. Rigid database schemas and monolithic architectures often cannot support dynamic AI applications. Organizations need infrastructure that can adapt to rapidly evolving AI capabilities without requiring complete system overhauls. This means adopting platforms that support schema flexibility, can handle diverse data types from structured databases to unstructured documents and multimedia content, and can scale both vertically and horizontally as AI workloads grow. The infrastructure must also support real-time data processing and streaming, as many AI applications require immediate access to the most current information.
- **Fluency in vector embeddings and semantic search technologies**: These technologies form the backbone of many modern AI applications. Vector embeddings allow AI systems to understand and process human language, images, and other complex data types by converting them into mathematical representations that machines can work with. While many AI implementations can succeed without deep technical knowledge of embeddings, organizations pursuing more sophisticated or differentiated AI solutions will benefit from teams that understand how to generate, store, and query these embeddings effectively. This includes knowledge of different embedding models, understanding when to use pre-trained versus custom embeddings, and expertise in vector databases and similarity search algorithms. This expertise becomes particularly valuable when building AI applications that can truly understand and reason about complex, domain-specific data.

- **Architectures that bridge AI and operations**: Too often, AI initiatives are built separately from the data sources and business systems they're meant to enhance. That leads to data silos, synchronization problems, and AI applications that work with stale or incomplete information. Successful organizations design architectures where AI capabilities are deeply integrated with operational systems, allowing for real-time insights and automated decision-making based on current business data. This integration requires careful consideration of data flow, API design, and event-driven architectures that can propagate changes across both traditional and AI systems.
- **Strategies for maintaining data consistency between systems**: This becomes critical when AI applications need to work alongside legacy systems during transition periods, though this remains one of the most challenging aspects of AI implementation. Organizations cannot typically replace all their systems overnight, so they need approaches for managing data synchronization across multiple platforms, even when perfect consistency may not be achievable. This includes implementing change data capture mechanisms, designing data validation processes, and establishing clear data governance policies. The strategy must also account for the fact that AI systems may process and transform data differently than traditional applications, requiring new approaches to data lineage and quality management. Organizations should expect this to be an ongoing challenge rather than a problem with straightforward solutions.
- **Guardrails for responsible AI**: As pressure builds to move fast, the temptation to cut corners grows. But without robust governance frameworks, organizations risk deploying systems that are biased, brittle, or out of compliance. Practical AI governance means codifying policies for data privacy and security, algorithmic bias and fairness, model explainability and transparency, and regulatory compliance across different jurisdictions. The governance framework must be enforceable, providing clear guidelines for AI development teams while not slowing down innovation unnecessarily. Done well, governance becomes a catalyst, not a constraint.

These approaches address a critical challenge at the AI inflection point: the need to bridge operational data and AI capabilities without creating new data silos or overly complex architectures. Organizations that successfully navigate this inflection point will find themselves with significant competitive advantages, while those that fail to adapt risk being left behind as AI transforms their industries. The key is to start building these capabilities now, before the competitive pressure becomes overwhelming.

Understanding modernization: The often-overlooked prerequisite

In addition to the high impact potential of innovation, there's another critical lever in the transformation toolkit: modernization. While modernization itself is a broader term often used to describe physical infrastructure, **technical modernization** refers to the *upgrading or replacement of outdated technologies, systems, or processes with newer, more efficient, and more advanced ones to improve performance, productivity, and competitiveness*. Key elements of technical modernization include the following:

- **Digitalization** using technologies such as automation, AI, cloud computing, or **Internet of Things (IoT)**
- **Digitization** of analog processes, for instance, replacing paper records with digital systems
- **System upgrades**, such as modernizing legacy IT infrastructure or software (commonly referred to as refactoring)
- **Integration** of modern tools into existing workflows to increase efficiency and reduce costs
- **Cybersecurity improvements** that address evolving threats and meet industry standards

The goal of modernization is to **enhance capabilities**, **reduce operational risks**, and **remain competitive** in a fast-changing technological landscape.

In the context of this book, modernization comes in two distinct forms. The first form involves modernization through the adoption of advanced technologies to address what is arguably the most significant challenge facing many enterprises today: **legacy systems**. Here, the term *legacy system* is used in a broad sense. While these systems often undergo physical upgrades every three to five years (a cycle deeply familiar in the mainframe world and certainly impacting IBM results), the software running on these systems often remains outdated. This leads to situations where companies are running decades-old business logic and software, despite having upgraded to newer hardware. The underlying code and business processes can be as old as half a century, creating significant challenges for modernization efforts.

The second approach, which is often the best way to go in scenarios such as this, is a complete system replacement built from scratch. But, as is often the case in business, the direct approach may not be available as the reasons for not touching a system as such are plenty: implicit business know-how, regulatory compliance, connectivity to existing machines, and more.

Common modernization strategies

Modernizing legacy applications isn't one size fits all. According to industry research and best practices [2], there are several common modernization strategies that organizations can employ:

- **Refactoring**: When a company refactors, developers update the code base, improving the code structure, performance, and maintainability. This can involve enhancing existing code without changing functionality, or migrating from older languages and frameworks to more modern alternatives. For example, an application built in Ruby might be refactored into Rust, or a system using an old version of Java with Spring might be refactored to modern Java with Quarkus. This action allows for independent scaling, easier maintenance, and access to current development tools and practices.
- **Replatforming**: This involves updating parts of an application's platform, such as the underlying database, and typically necessitates some code base modifications as well.
- **Rehosting or redeploying**: Sometimes called *lift and shift*, this method involves moving applications to a public cloud, private cloud, hybrid cloud, or multi-cloud environment. It's a straightforward way to gain some of the cloud's benefits without extensive rework, but it does not attempt to address any problems with the application as is.
- **Rearchitecting**: This may involve updating the code base to leverage modern architectures, such as containers or microservices.
- **Rebuilding**: Starting from scratch, while preserving the application's scope and specifications, is also an option and is useful when the existing application is too outdated or inefficient for modernization.
- **Replacing**: This involves starting over with a completely new code base, beginning from basic requirements upward. Rather than replicating the legacy application, this approach goes back to the business to collect current, modern requirements and build from there. Many applications aren't worth modernizing because business processes have evolved significantly since they were originally built, making replacement often the most sensible path forward. This allows organizations to eliminate outdated features while incorporating new capabilities that align with current business needs.

The choice between these strategies depends on several factors, including the current state of the application, business requirements, available resources, and timeline constraints. Organizations often find that a hybrid approach works best, combining multiple strategies across different components of their systems. For instance, they might refactor critical applications while rehosting less critical ones, or rebuild core systems while replacing outdated peripheral applications.

Each strategy involves trade-offs between cost, risk, and potential benefits. Refactoring and replatforming offer lower risk but more limited improvements, while rebuilding and replacing provide greater transformation potential but require more significant investment and carry higher implementation risks. The key is aligning the modernization approach with business objectives and technical constraints.

While these strategies outline how modernization can be approached at the application level, executing them effectively often depends on the foundation provided by modern data platforms.

Where innovation meets modernization: The AI intersection

This is where AI-driven innovation and infrastructure modernization converge. Over the last 20 years, many companies have offered elaborate *modernization approaches* or full-scale practices, often involving reviews of existing software and processes to ultimately do a de facto rewrite from scratch, essentially rebuilding entire systems from the ground up while calling it *modernization*.

However, as we will discuss further in *Chapter 5*, *Modernization Using AI*, these complete rewrites have significant pitfalls. They lack the repeatability and transparency required to drive true modernization of existing systems, often becoming resource-intensive, one-off projects that risk losing embedded business logic and institutional knowledge.

AI shifts the equation. Instead of defaulting to wholesale replacement, it enables smarter, more sustainable modernization, preserving what works while evolving what doesn't. AI can analyze legacy code, convert it to more digestible, modern formats (such as different programming languages or paradigms), and facilitate comprehensive architecture changes from monolithic systems to modern distributed, n-tier architectures, including microservices-based designs. It accelerates test generation, improves maintainability, and bridges the gap between brittle systems and modern workflows. These strategies, explored further in *Chapter 5*, *Modernization Using AI*, equip organizations to evolve systems iteratively without starting over.

The AI implementation pitfall: When innovation lacks foundation

The second part, where modernization and AI innovation align, is more subtle. The strong push through the business for AI solutions and utilizing AI to improve business functionality may lead to quick solutions that are neither sustainable nor fit the definition of production readiness.

When AI projects don't make it to production

Consider this real-world scenario. A large multinational company with an elaborate innovation process received a mandate from business leadership to deliver AI across the organization. Excited by the possibilities, the company initiated a broader evaluation that ultimately spawned 500 independent projects, all focused on implementing some form of AI capability. The organization invested significant resources, assembled teams across multiple business units, and generated considerable enthusiasm for the AI transformation. Twelve months later, when these projects underwent formal evaluation for production deployment, none made the cut to move forward. Not a single project among the 500 initiatives demonstrated sufficient business value, technical reliability, or operational readiness to justify continued investment.

Analysis showed that there was a disconnect between resolving actual business problems and implementing solutions that, while appealing from an IT perspective, had no measurable impact. Second, each project was driven by independent research conducted by individuals or smaller teams, which ignored fundamental discussions about production deployment or continuous integration into workflows. Lastly, data integration was overlooked, as people worked with sample data that did not reflect the requirements of real-time use cases. Any changes or updates to these datasets could have led to unexpected results on the LLM side beyond simple hallucinations. For example, changing datasets with different calibrations (such as a software upgrade in a manufacturing line), could've resulted in, at best, unsellable products and, at worst, dangerous goods.

In other words, modernizing underlying data infrastructure is a prerequisite for such use cases to move beyond experimentation and begin actual implementation. AI applications require a substantially different approach to data infrastructure compared to legacy applications that utilize the same schema for decades; they must handle diverse data types, support real-time processing, and accommodate rapidly evolving business requirements and integration needs.

In this context, modernization is not an optional, nice-to-have side effect but a hard prerequisite for the production implementation of AI functionality. As a side effect, the modernization of data landscapes will simplify the discussion with the business about the actual value of AI.

So, what does a solid foundation look like in practice? It starts with modern data platforms designed to balance both the agility of innovation with the rigor of modernization.

Modern data platforms: The backbone of AI-ready transformation

Modern data platforms are not just an infrastructure upgrade; they are foundational to enabling AI-driven transformation, supporting both innovation and modernization simultaneously. In an era defined by rapid change, traditional systems are too rigid to meet the evolving needs of businesses pursuing AI. A modern platform bridges this gap by offering the flexibility, speed, and scalability that both innovation and modernization demand.

Why modern data platforms are necessary

At the heart of modernization and AI readiness is the ability to manage and operate on diverse, rapidly changing data. Many enterprises rely on relational databases with fixed schemas and monolithic architectures that require significant modification to support dynamic AI applications. Modern data platforms, especially those built around document-based or non-relational architectures, offer enhanced capabilities for contemporary workloads, including schema flexibility, real-time processing, and horizontal scaling, which AI applications typically require.

This flexibility is a strategic necessity. Modern business requirements often include real-time decisions, multi-format data, and agile development, none of which are feasible with legacy data systems. As organizations pursue AI transformation, their ability to innovate, modernize, and scale depends directly on the underlying data infrastructure.

Enabling innovation through agility and speed

Traditional databases force developers to spend weeks defining exact data structures before writing any code. If you want to add a new field or change how data connects, you need database administrators to manually update schemas, often taking days or weeks. Modern platforms let developers start coding immediately with whatever data structure makes sense. Need to add customer preferences to your insurance app? Just start storing that data. Want to test a new claims workflow? Build it without waiting for schema changes.

This means a team can go from idea to working prototype in days instead of months. Teams can build, test, and iterate on new solutions, such as autonomous underwriting or real-time claims processing, without relying on manual changes to rigid database structures.

Additionally, platforms such as MongoDB accommodate natural data formats such as images, documents, and time-series data, enabling a wider range of use cases. For example, by removing the need for complex data transformations, organizations can bring together previously siloed data sources to develop new applications faster, increasing time to value.

This flexibility and speed are essential because innovation is a continuous process that requires ongoing development and iteration. As ideas mature, modern data architectures evolve with them, supporting product growth, feature expansion, and user-driven feedback loops without requiring disruptive migrations.

Simplifying modernization without starting over

Modernization efforts are notoriously difficult, especially when legacy systems contain decades of embedded business logic and regulatory dependencies. Modern data platforms reduce the risk and cost of modernization by supporting incremental migration paths.

Instead of forcing big-bang rewrites, these platforms enable hybrid deployments where legacy and modern systems coexist. Seamless integration tools analyze existing schemas, map legacy structures into modern formats, and maintain data integrity during transitions. These platforms keep old and new systems synchronized automatically; when a customer updates their address in your legacy system, that change instantly appears in your new system too. This means you can migrate customers and functions gradually instead of risking a complete system shutdown. Furthermore, cloud-native deployment capabilities offer the flexibility to scale infrastructure on demand, reducing upfront costs and allowing experimentation with new architectures in sandbox environments. This makes it possible to modernize without losing institutional knowledge or disrupting core operations.

Powering AI at scale

AI applications depend on more than just models; they require infrastructure capable of storing, retrieving, and reasoning over complex data. Modern databases meet this need by supporting advanced AI workflows, including vector embeddings and semantic search.

By storing vector representations of data such as those generated by LLMs alongside operational records, organizations can implement capabilities such as similarity search, context-aware recommendations, and intelligent automation. Integrated support for vector search eliminates the need for disjointed pipelines and enables tighter coupling between operational data and AI decision-making.

Additionally, real-time data processing capabilities ensure that AI systems work with current information rather than static snapshots. As AI systems evolve rapidly with new models and architectures emerging every few months, modern platforms offer the agility needed to adapt without reengineering foundational systems.

Summary

This chapter explored the critical distinction between innovation and modernization in the context of AI transformation. It examined Andy Grove's strategic inflection points theory and demonstrated how today's AI revolution represents a comparable inflection point to the computing revolution of the 1980s.

The discussion illustrated why modernizing legacy systems and data infrastructure is essential for successful AI implementation through both cautionary tales and success stories. The chapter established that AI innovation requires a solid foundation. Modernization addresses the burden of legacy infrastructure, particularly in data systems.

By understanding the relationship between innovation and modernization, and leveraging modern data platforms that support both, organizations can successfully navigate the AI inflection point and create a sustainable competitive advantage.

The next chapter demystifies the AI landscape. You'll understand how the core technologies, that is, semantic search, LLMs, RAG, and agentic AI, fit together to create business value.

References

1. *Superagency in the workplace: Empowering people to unlock AI's full potential*: `https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/superagency-in-the-workplace-empowering-people-to-unlock-ais-full-potential-at-work`
2. *Why Application Modernization is Vital for Business Growth*: `https://www.mongodb.com/resources/solutions/use-cases/application-modernization#common-modernization-strategies`

3

The System of Action

Data is the foundation for turning GenAI into measurable business outcomes. AI-enabled capabilities such as vector-encoding of unstructured data, real-time synthesis across silos, and agents-in-the-loop are changing *how* we transact business. Enterprises that master these capabilities can move from reactive, siloed decisions to proactive, intelligence-driven operations, responding to market changes in real time, lowering operational expenses, and driving stronger customer outcomes.

Augmenting and automating business workflows with agents-in-the-loop places increased demands on the underlying data layer. Agents may access and analyze a myriad of siloed sources, build rich context, and inform business decisions. But where does that context persist? How can business users view, interact with, or even edit that information?

To support this evolution, enterprises must move beyond static *systems of record*, which passively store and serve data, toward dynamic *systems of action*: systems designed for real-time decisions, automation, and collaboration between humans, AI-assisted users, and fully autonomous agents to *act* on data. While traditional systems of record excel at maintaining data integrity and compliance, systems of action can autonomously trigger decisions, execute workflows, and learn from outcomes. For example, in retail, an AI agent could reorder inventory in real time as customer demand shifts.

This chapter explores how systems of action transform data into decisions: a shift that is becoming unavoidable. They must handle data in many forms (from original sources and text search indices to vector embeddings) and even LLM inputs requiring context-sensitive reranking, a process for prioritizing the most relevant results. Meeting those demands requires a unified, contextualized view of enterprise data that exceeds what their system of record and *system of insight* predecessors can provide. This shift also introduces new challenges in scalability, performance, security, and governance.

By the end of this chapter, you'll understand how:

- System of action databases support real-time AI and RAG applications by breaking traditional data modeling constraints and enabling signal processing with responsive, document-based data layers
- Unified data access enables GenAI to process diverse formats such as source data, embeddings, and real-time signals within a coherent framework
- Data quality and consistency reduce hallucinations and improve reliability through full lineage tracking and provenance awareness
- AI-ready data architectures break from traditional warehouses and systems of insight, supporting dynamic, multimodel workloads
- Governance and security strategies align with AI-specific needs such as privacy, access controls, and encryption
- Model training and fine-tuning pipelines prepare and optimize data for GenAI applications
- Implementation patterns follow the flow from signal ingestion through enrichment to intelligent response, providing deployment blueprints

Building an AI-ready data foundation

Delivering on the promise of systems of action requires a new kind of data foundation—one built for speed, context, and adaptability.

Agentic AI systems fundamentally differ from traditional systems of record in their operational demands. Where legacy systems focus on capturing and storing historical transactions, systems of action powered by agentic AI require real-time decision-making, dynamic data synthesis, and immediate response capabilities. This shift demands that our data architecture choices move beyond the rigid, siloed structures of traditional enterprise systems.

A unified view of core enterprise is essential. It must bring together the diverse data types that autonomous agents rely on (real-time operational signals, contextual documents, vector embeddings) into a single, coherent platform. That platform must be built on flexible data structures that can adapt as agent behaviors evolve.

The transition from supporting passive systems of record to enabling active systems of action introduces six critical architectural requirements that distinguish agentic AI infrastructure from legacy approaches:

- **Unified data access** to eliminate the complexity of managing multiple disparate datastores
- **Data quality and consistency** mechanisms that reduce hallucinations and errors from systems out of sync
- **Real-time context** capabilities that enable immediate signal processing for RAG applications
- **Scalability and performance** characteristics that support operational AI rather than only backward-looking analytics
- **Governance and security** frameworks that protect sensitive information while enabling innovation
- **Efficient model training workflows** that optimize data preparation for GenAI applications

Together, these elements form the data foundation for autonomous, intelligent systems. As we examine each in the sections ahead, we'll see how a system of action database departs from traditional data management and enables more intelligent, responsive, and scalable AI applications.

What is a system of action?

Systems of action are a new class of enterprise application, designed to execute decisions and drive workflows in real time. They enable collaboration between people, AI-assisted users, and AI agents, supporting everything from assisted decision-making to fully autonomous execution.

Unlike systems of record, which passively store historical transactions, or systems of insight, which analyze data retrospectively, systems of action operate in the moment. They process dynamic context, trigger decisions, and execute tasks through AI agents. For instance, they might reroute a delayed flight in real time or automatically adjust hospital staffing during a sudden surge.

Building systems of action requires more than analytical capabilities. They must ingest streaming signals, reason across unstructured and structured sources, and respond in real time. They require specialized database architectures capable of managing high-velocity, multimodal data streams and supporting complex state transitions over time. Most legacy systems, designed for static, batch-oriented workflows, simply cannot support this kind of continuous intelligence.

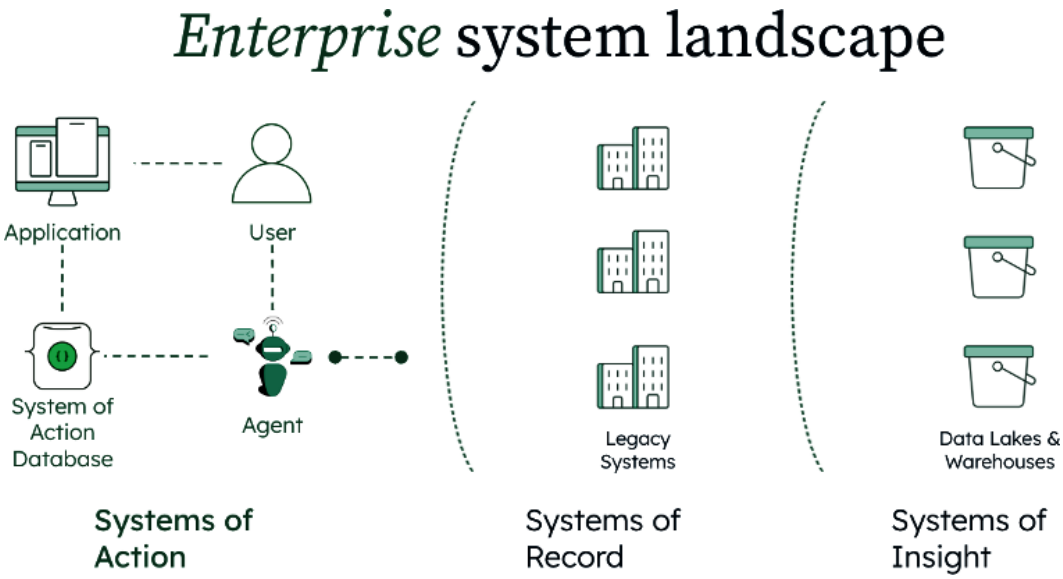

Figure 3.1: Enterprise system landscape: from system of record to system of action

Figure 3.1 illustrates this evolution across the enterprise landscape. Unlike traditional systems that passively store or retrospectively analyze data, systems of action enable real-time interaction between users, applications, and agents; all powered by a live, adaptable data layer.

Unified data access architecture

The foundation of any GenAI system begins with access to diverse, multimodal data, at speed, in formats AI can reason with. Unfortunately, this is also where most enterprises struggle. Traditional enterprise data architectures are fragmented across dozens of incompatible systems, each optimized for narrow use cases. The result is integration pain, access friction, and massive overhead.

Modern AI applications demand a fundamental departure: unified access must be treated not as a convenience but as a prerequisite.

Today's models must navigate a wide variety of inputs: text documents, application logs, product catalogs, support transcripts, and streaming sensor data. Relational and legacy systems often store semi-structured data (like JSON or XML) as **binary large objects (BLOBs)** or **character large objects (CLOBs)**, limiting their usability for AI systems. In these cases, the actual data is hidden inside a single entry and must be extracted and interpreted before it can be reasoned over or acted upon. This was tolerable when the goal was to store and retrieve files. But for GenAI systems, where models need immediate access to both structured and semi-structured data, often in the same query, this format becomes a bottleneck. Even a video can have its own addressable

metadata structure, rather than existing solely as an opaque BLOB, illustrates the shift needed to support AI-native reasoning.

Beyond the format problem lies a more urgent challenge: **fragmentation**.

An AI application might need to stitch together context from a CRM (customer profiles and account hierarchies), a product catalog (SKU-level details, pricing, availability), a data warehouse (historical transactions), a streaming platform (real-time behavioral signals), and a document store (contracts, support transcripts, policy documents). Each source has its own schema, access pattern, and often its own API. This complexity creates two persistent challenges:

- **Developer integration friction**: Each layer introduces its own headaches, from authentication and authorization to schema mismatches, brittle connectors, and inconsistent formats
- **System fragility/maintenance drag**: Over time, these integrations accumulate, introducing silent failures, versioning issues, and downstream reliability risks that make innovation slower and more expensive

MongoDB's document model takes a fundamentally different approach. Instead of forcing diverse data into rigid schemas or hiding it in unreadable blobs, it enables rich, hierarchical data structures that mirror how businesses actually operate [1]. Developers can model a full customer, order, or event in a single document, including nested context, version history, and behavioral attributes. This eliminates the need for complex joins while preserving the relationships critical for effective agentic reasoning.

Even more critically, *flexible schema design*, meaning the ability to store and query data without locking into a rigid blueprint, allows fields and document shapes to adapt as requirements change. This lets data evolve—new attributes can be added without downtime, and new types of signals can be integrated without costly migrations. For AI systems (especially those that learn, adapt, and extend themselves), this agility is essential.

This architectural convergence enables structured transactions, real-time signals, and unstructured content together in a single query or operation. Model updates, enrichment jobs, or downstream agent actions can all be triggered directly from the same data platform [2]. That unified model lays the groundwork for sophisticated, AI-native workflows.

Perhaps more importantly, unified data access transforms developer productivity. Instead of spending cycles reconciling formats or debugging brittle connectors, teams can focus on building intelligent systems. And, as we'll see in the sections ahead, everything from data quality and governance to real-time orchestration builds on this foundation.

Ensuring data quality and consistency

Data quality and consistency are non-negotiable for GenAI solutions. Unlike traditional analytics, where data quality issues might simply yield incorrect reports or delayed insights, poor data quality in AI systems can cause hallucinations, introduce biased outputs, and fundamentally unreliable behavior that undermines user trust and business value.

Legacy quality approaches tried to solve this through normalization, deduplication, and validation against external sources. Consider a familiar failure mode: a system validates `Joe Miller, 12 High` Street, through postal APIs and credit checks, yet fails to distinguish between three different Joe Millers (grandfather, father, and son) at the same address. For entity analytics, where precise relationship mapping matters, this is a critical flaw.

In this scenario, an online store might unknowingly treat all three individuals as the same customer, losing the ability to tailor interactions or offers. Relational star schemas exacerbate this problem by fragmenting contextual information across multiple tables. When customer data is split between fact tables, dimension tables, and lookup tables, the rich context that enables accurate entity resolution becomes scattered and difficult to reconstruct.

In our Joe Miller example, a document-based approach would maintain separate documents for each individual, complete with detailed demographic information, purchase history, behavioral patterns, and relationship data that enables clear differentiation.

Within a document, you can store original values alongside enrichments and enhancements within the same dataset. This approach improves output reliability and reduces hallucinations or contradictory results. When an AI system generates an output, the complete chain of data sources, transformations, and reasoning steps can be traced back through the document structure, enabling both debugging and compliance reporting.

This lineage capability proves essential for improving output reliability and reducing hallucinations or contradictory results. When AI models can access not just the current state of data but also its provenance and transformation history, they can make more informed decisions about data reliability and confidence levels. For example, customer service AI might weigh recent direct customer interactions more heavily than older inferred preferences, or flag potential inconsistencies when multiple data sources provide conflicting information.

For organizations implementing document-based data quality strategies, MongoDB offers comprehensive best practices, as well as compatibility with industry-leading tooling for data modeling and cataloging that make advanced quality management achievable at scale [3]. When high-quality, lineage-aware data becomes the default, AI systems can deliver results that are accurate, explainable, and trustworthy.

Real-time context and RAG

The definition of *real-time* varies significantly by use case and industry, but the real-time requirements of data in use with GenAI cannot be overstated. Hedge fund trading systems, for example, require millisecond responses, while life insurance underwriting processes measure time in days. While application response times continue to decrease, many architectures use caching layers that create an illusion of real-time performance at the expense of freshness of data.

A typical real-time environment follows a simple pattern where an interaction generates a signal that enables immediate interpretation. These signals may originate from diverse sources, such as a retail website recording shopping cart additions, a smart meter transmitting electricity usage, or a pathology lab completing cancer analysis data. All signals, when combined with existing datasets, enable text search, vector search, and LLM processing for reasoning and causal analysis. This applies equally to interactive systems, such as retail shopping carts, and autonomous agentic systems, such as automated insurance claim processing.

Real-time integration of signals with metadata, reference data, and historical information generates new knowledge instantaneously. Consider how this has evolved. Traditional rule-based systems might suggest `"You ordered a burger, would you like fries?"` In contrast, an AI-powered system recognizes patterns such as `"You order cat food bi-weekly, always the same brand"`, and reasons contextually with suggestions such as `"Based on your purchase history, you might be interested in our new, healthier formula. Would you like us to send you a free sample?"` The system identifies repeat customers and enhances their experience through reasoning that connects purchase patterns with product recommendations, requiring deeper knowledge about customer preferences and pet characteristics.

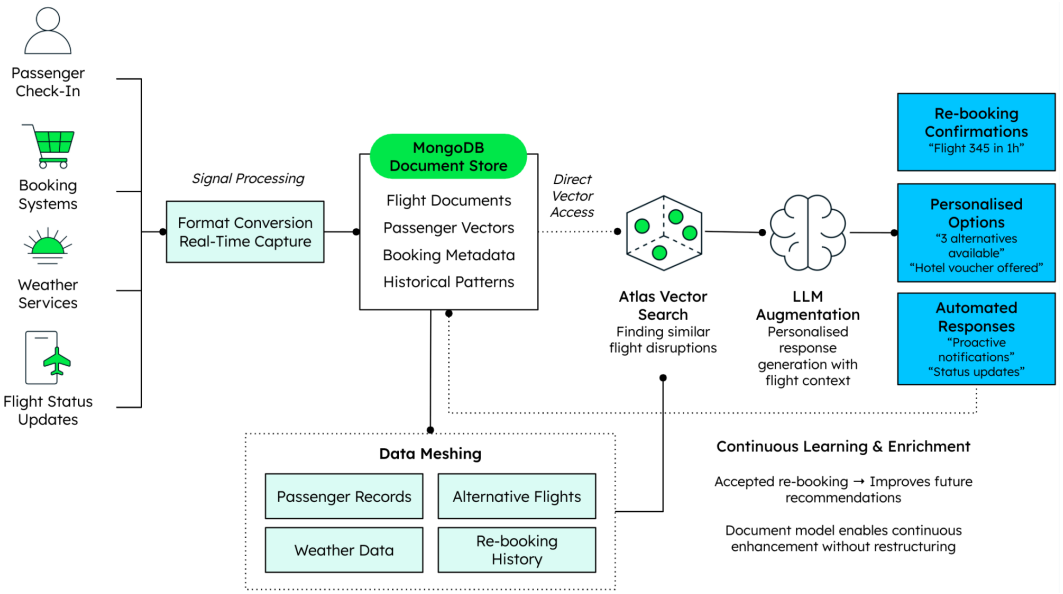

Figure 3.2: Real-time AI data flow

The architectural flow in *Figure 3.2* demonstrates how modern AI applications process real-time signals through a system of action database using an airline passenger assistance scenario. The flow begins with diverse signal sources on the left: **Passenger Check-In**, **Booking Systems**, **Weather Services**, and **Flight Status Updates**, which feed into **Signal Processing** and **Format Conversion/Real-Time Capture** components. These signals are then ingested into the central **MongoDB Document Store**, which contains **Flight Documents**, **Passenger Vectors**, **Booking Metadata**, and **Historical Patterns** with **Direct Vector Access** capabilities.

The system processes this data through **Atlas Vector Search** (finding similar flight disruptions) and **LLM Augmentation** (generating personalized responses with flight context) to produce three types of intelligent outputs: **Re-Booking Confirmations**, **Personalized Options**, and **Automated Responses**. At the foundation sits the **Operational Data Layer (ODL)**, an architectural pattern that centrally integrates and organizes siloed enterprise data, serving as an intermediary between existing data sources and consuming applications. In this case, the ODL enriches signals with contextual information from passenger records, alternative flights, weather data, and rebooking history.

A continuous learning and enrichment feedback loop ensures that every interaction outcome, whether accepted re-bookings or user preferences, flows back to improve future recommendations. The document model enables continuous enhancement without requiring system restructuring, creating a system that grows smarter with each passenger interaction while delivering real-time, context-aware responses vital for modern AI applications.

Critically, the feedback loop ensures continuous improvement, ensuring every interaction outcome enriches the system of action database, making future responses more accurate and contextual. This circular flow embodies the key advantage of document-based architectures: *the ability to evolve and improve without the schema rigidity that constrains traditional relational systems*. The result is a system that grows smarter with each interaction, delivering real-time, context-aware responses that modern AI applications require.

Scalability, availability, and performance

Historically, enterprise data warehouses represented the largest database implementations, with denormalized, column-oriented star schemas designed for analytical queries. These systems perform well with queries such as `"Display yogurt sales by region"`, where large datasets are filtered by specific criteria (region, store, price) to generate insights. The integration of multiple sources led to the development of **extract**, **transform**, **load** (**ETL**) processes and master data management systems. While these platforms have added machine learning features and now claim to support GenAI capabilities, they remain primarily designed for backward-looking analytical tools, unsuited to real-time, agentic, and causal AI applications.

Consider the contrast. A chatbot assisting an airline passenger who missed a connection requires fundamentally different capabilities than answering `"How many passengers experienced day-long delays in Frankfurt last year?"` The chatbot and its underlying agentic system must address immediate needs, finding available seats, offering mitigation services, and responding empathetically to frustrated passengers. The required data is real-time, context-sensitive, and simply not available from a historic warehouse.

To be successful in the request for the passenger, the system needs both real-time seat information access (easy to achieve with an API to the usual booking systems), as well as more important detailed context and information about the passenger and their situation. Is it a family stranded, or a single adult? What other ticket dependencies exist? Can the passenger be rerouted via a different track, or is the best option to stay overnight?

This scenario demands that all passenger data reside in an up-to-date system of action database, as real-time interactions fail without current information. As these systems achieve global coverage, non-functional requirements mandate not only 24/7/365 availability but also the ability to handle transaction volume fluctuations from quiet periods to peak travel seasons such as Thanksgiving. Even minimal outages become unacceptable, and caching solutions that simply solve a data availability challenge compromise on data *accuracy* by introducing data staleness issues.

Document-based architectures, such as those provided by MongoDB, offer advantages in specific scenarios for this type of data availability and scalability. Rather than requiring complex joins across multiple tables to reconstruct user context, document models can store complete contextual information in a single, efficiently retrievable record. This approach reduces the computational overhead of context reconstruction while enabling more sophisticated caching and optimization strategies.

The performance characteristics of AI workloads also differ significantly from traditional analytical patterns. While analytical queries typically process large volumes of data to generate aggregate results, AI applications often require rapid access to specific, contextually relevant information. This pattern favors architectures optimized for high-concurrency, low-latency access to individual records, rather than bulk processing of large datasets.

Governance, security, and compliance

Governance and compliance requirements stem from a fundamental need to protect individuals from flawed decision-making in systems that lack adequate self-regulation. These safeguards exist to prevent real harms, from biased loan approvals to unsafe product recommendations.

GenAI faces intense scrutiny regarding accuracy, with media coverage of hallucinations bringing this concern to the forefront. Therefore, transparency in data lineage, reasoning processes, and result interpretation becomes critical for any GenAI solution. The document model in a system of action database enables tracking of all changes, transformations, and actions related to specific datasets. Unlike legacy relational databases, documents offer the flexibility for enhancement and enrichment throughout the process without requiring upfront planning.

From a governance perspective, this enables precise and comprehensive tracking of communication and decision-making processes. It facilitates decision auditing and corrective actions when compliance challenges arise, often due to gradual shifts in decision criteria requiring adjustment.

Security represents an additional critical dimension. MongoDB's Queryable Encryption keeps data absolutely protected from unauthorized access. While passenger data may have moderate sensitivity, healthcare provider consultations about potential illnesses require the highest security levels. The system of action database enables transparent security implementation, significantly more challenging when coordinating multiple data sources with potentially incompatible security and policy systems [4].

Model training and fine-tuning

Training or fine-tuning models requires large volumes of clean, labeled, and diverse data. The system of action database ensures efficient data curation, sampling, and preprocessing for training pipelines. Data enrichment becomes key, as features such as MongoDB's aggregation pipeline enable data annotation and continuous analysis of criteria such as minimum or maximum values and moving averages to validate reasoning processes.

The subject of data preparation for GenAI is often misunderstood, stemming from the evolution of early AI solutions supporting ML systems (systems that were derived from **business intelligence** (**BI**) architectures). This sometimes leads to the mistaken assumption that all data for AI usage and interaction must first be prepared, or readied, in lakes, warehouses, or marts, requiring extensive transformation and data pipeline processing. The resulting data objects are often stored as star schemas with fact tables, each containing hundreds of columns and accompanying dimension tables. Star schemas, a data modeling format originally designed to solve the problem of performant analytics queries executed against relational database objects, introduce the need for complex queries and join operations to extract insight, an architecture still employed by platforms such as Snowflake.

Apache Spark object-storage implementations, such as Databricks, offer more complex query capabilities through distributed computing frameworks and in-memory processing, representing a significant advancement over traditional batch processing systems. Both approaches, star schemas and Spark-manipulated object storage files, share a foundation in backward-looking data warehousing, regardless of contemporary terminology such as *data lake* or *lakehouse*.

These systems are optimized for processing large volumes of homogeneous data aligned along dimensional axes. Real-time access to individual datasets for operational processing falls outside their design parameters. Historically, this was the realm of **online transaction processing** (**OLTP**) systems. While transactional logging isn't central to GenAI data structures, the access patterns remain similar.

Often, the example of building models for embeddings is referenced as justification for why the data warehouse must be the source of data for GenAI, but this is misleading. Firstly, many business solutions successfully deploy standard embedding models for PDFs, images, and audio, without the need for custom development. Secondly, and more importantly, the comparison doesn't hold, as warehouses analyzing quarterly sales have no relevance to point-of-sale operations and transaction booking.

Practical considerations for AI data design

While the theoretical foundations of system of action databases provide the conceptual framework for AI-ready system of action architectures, successful implementation requires attention to practical design principles and operational realities. This section examines three critical aspects that determine the success of real-world GenAI implementations: the fundamental importance of well-structured and organized data, patterns of data movement through AI systems, and the operational considerations necessary for deploying and maintaining these architectures at scale.

A good data structure is critical

Well-organized data, such as documents, indexes, and embeddings, enable fast and relevant information access, essential for RAG.

In contrast, traditional relational database storage patterns create fundamental impedance mismatches with GenAI applications, which require unified, contextually rich data objects that preserve semantic relationships and business meaning. When business entities span reference data, metadata, and operational information across multiple normalized tables, extensive join operations and additional application-layer code are often required to reconstruct fragmented data into coherent business objects, ready for AI processing. This relational fragmentation not only degrades query performance and increases system complexity but also obscures the natural data relationships essential for effective AI consumption, creating artificial boundaries that require ongoing maintenance of interdependent schema relationships. Consequently, GenAI applications demand data architectures that can natively represent complex, hierarchical business entities without extensive reconstruction logic, favoring document-based architectures that align more naturally with how AI models consume data and that is, rich, contextual objects, rather than decomposed, normalized fragments.

A well-designed system of action database based on the document model is a solution that addresses all the above challenges while also delivering a wealth of non-tangible benefits, including reducing developer cognitive load and infrastructure tech sprawl.

The combination of various sources enhances metadata understanding and improves the accuracy and relevance of the models. From a developer perspective, document structures enable superior prompt construction, as denormalized documents are much easier for both developers and LLMs to work with, thereby improving the quality of the prompts. Additionally, the simpler structure with reduced normalization reduces the number of data objects required to generate appropriate context. The fewer the number of data objects, the easier it is to ensure data quality for the purposes of reducing hallucinations and improving factuality.

In the previous airline example, multiple models might contribute different reasoning aspects: a general LLM to handle chatbot communication, and a domain-specific fine-tuned model for determining routing options. Secondary LLMs might specialize in summarization versus data validation, demonstrating why relying on a single LLM likely may not achieve reasonable quality in agentic systems.

The document model enhances output quality through iterative enrichment, evolving with use cases rather than requiring re-architecting or refactoring to accommodate new or changed requirements (an unavoidable pain point in relational systems). Ultimately, enriched documents improve the fluency, coherence, and creativity of reasoning-based agentic systems.

Poor data retrieval design, stemming from loosely organized sources, results in slow performance, irrelevant results, or inaccurate information. Conversely, well-aligned architectures ensure fast, contextually relevant retrieval that supports meaningful model output.

In short, when data structure and model design are thoughtfully aligned, the result is a more accurate, responsive, and scalable GenAI solution.

Data flow

GenAI solution data flow typically begins with transactional data captured during machine, process, or human interactions. It may then be enriched with related unstructured artifact vector-encoding, or even existing reference data, so that it can be made actionable within agentic system workflows. This real-time business object enrichment in MongoDB results in a single document consisting of all relevant information enriched to the maximum extent. In a legacy architecture design, by contrast, the data would be referenced only from its sources, requiring, at best, calls to APIs, and at worst, possible direct access to databases.

Data flow with a document-based approach allows the passing of all contextual information in one object and format, facilitating collaborative work between multiple agentic systems for a single process or workflow. In our example, the flight interruption is compiled into a single document, managing all different aspects of the case, including passenger interactions, flight information, contractual data, and even situation factors such as weather conditions.

This enables multiple agentic systems to collaborate on different aspects of the same business object or transactional interaction. For example, while a chatbot communicates with the passenger and provides real-time status updates, another system component proactively works on the underlying issue, having been triggered by the transactional system detecting a flight delay and calculating the probability that the passenger will miss their connecting flight. This means the *case* begins processing before the passenger even initiates contact.

During this case creation process, the system generates multiple vectors through specific embeddings and performs semantic searches against similar historical cases. This allows the LLM to prepare natural language responses, such as `"We identified multiple flight options…"`, at the earliest possible moment. In an optimal scenario, the agentic system can proactively generate multiple solutions and communicate them to the passenger with a message such as `"We are sorry you missed your flight. We have three available options to continue your journey…"`.

Once the case is successfully resolved, the document is enriched with comprehensive outcome data, such as `"Passenger accepted rebooking on Flight 345 departing in one hour, expressed satisfaction with proactive communication, and declined meal voucher offer"`. The interactions with the passenger can then be interpreted, and the overall outcome classified. This allows critical quality assurance, helps identify emerging trends, and validates and tests newer models and their suggested outcomes.

Operationalizing a system of action database

Moving from architectural principles to production deployment requires addressing the operational complexities that distinguish AI data systems from traditional databases. The unique characteristics of GenAI workloads, including real-time vector search, continuous model evolution, and dynamic schema requirements, demand specialized approaches to deployment, monitoring, and maintenance that extend far beyond conventional database administration practices.

Deployment patterns

Implementing a system of action database requires careful planning around deployment architecture. Organizations typically follow one of three primary patterns: greenfield implementations for new AI initiatives, gradual migration strategies that slowly transition away from legacy systems, or hybrid approaches that maintain existing systems while building new AI capabilities alongside them.

Cloud-native deployments offer the greatest flexibility and scalability, with managed services such as MongoDB Atlas, providing automatic scaling, backup, and security features. On-premises deployments may be necessary for organizations with strict data sovereignty requirements, while hybrid cloud approaches can balance security needs with operational efficiency.

Performance monitoring and optimization

Real-time AI applications demand continuous performance monitoring across multiple dimensions. Query performance metrics must track not just response times but also relevance scores for vector searches and accuracy metrics for AI-generated outputs. Document-based systems require monitoring of collection sizes, index effectiveness, and aggregation pipeline performance.

Key performance indicators should include throughput metrics for data ingestion, latency measurements for retrieval operations, and resource utilization patterns during peak AI processing loads. Automated alerting systems should trigger when performance degrades below acceptable thresholds, particularly for real-time applications where delays directly impact user experience.

Cost management and resource allocation

The economics of AI data infrastructure differ significantly from traditional database systems. Vector storage and similarity searches consume different resource patterns than relational queries, requiring new approaches to capacity planning and cost optimization.

Storage costs scale with both document size and embedding dimensions, while compute costs vary based on model complexity and query frequency. Organizations should implement tiered storage strategies, moving older or less frequently accessed data to less frequently accessed, lower-cost storage tiers while maintaining frequently accessed, or hot data in high-performance systems for real-time access.

Maintenance workflows and data lifecycle management

Document-based AI systems require specialized maintenance procedures that account for schema evolution, embedding model updates, and data quality drift over time. Unlike traditional databases, where schema changes require careful migration planning, document stores allow for more flexible evolution, but this flexibility doesn't negate the need for governance frameworks that ensure data consistency.

Regular re-processing of embeddings becomes desirable as newer, more effective models become available. Automated pipelines should be utilized to manage embedding updates while maintaining high system availability, potentially utilizing blue-green deployment strategies to minimize system disruptions during major model transitions.

Migration strategies from legacy systems

Organizations rarely start with a clean slate when implementing systems of action data stores. The necessary migration of data from existing relational systems, data warehouses, and disparate operational systems requires phased approaches that minimize business disruption while maximizing the benefits of unified data access.

Experience from a wide range of customers across many industries has shown that the most successful migrations begin with pilot projects that demonstrate value quickly, then expand scope incrementally. Data synchronization strategies should maintain business data parity between old and new systems during transition periods, with automated validation ensuring data integrity throughout the migration process.

Team training and adoption considerations

Successfully operationalizing a system of action databases requires investment in team capabilities. Traditional database administrators may need training in document modeling principles, while application developers may need to learn new query patterns and optimization techniques specific to AI workloads.

Data scientists and ML engineers require an understanding of how document structures impact model training and inference performance, while DevOps teams need familiarity with AI-specific monitoring and scaling requirements. Cross-functional collaboration becomes essential as the boundaries between data engineering, AI development, and operations blur in system of action architectures.

Summary

This chapter explored the crucial role of system of action databases in order to build effective GenAI solutions. We examined the limitations of traditional data management approaches and presented an alternative paradigm centered on the document model. Key aspects discussed include unified access to diverse data sources, improved data quality and consistency, real-time context for RAG, scalability, security and governance, and the importance of aligning data structures with model design for optimal performance and accuracy.

The primary benefit of a well-designed system of action database, based on the document model, is its ability to provide real-time, context-sensitive, and holistic access to diverse data, essential for reasoning, causal analysis, and generating accurate, relevant outputs. This approach differs from traditional data warehousing by providing a unified view of enriched data optimized for RAG, model training, and fine-tuning. This results in GenAI solutions that are safer, more accurate, responsive, scalable, and secure.

In the next chapter, we will explore the critical foundations of trustworthy AI, examining how organizations can navigate the complex landscape of ethical frameworks, regulatory compliance, and data governance requirements. As AI systems become increasingly embedded in critical decision-making processes, ensuring they operate within the boundaries of ethics, law, and society becomes paramount. We will discuss how proper data governance, the foundation we've established with system of action databases, enables organizations to build AI systems that are transparent, fair, accountable, and compliant with evolving regulations across different industries and jurisdictions.

References

1. *MongoDB's document model approach*: `https://www.mongodb.com/docs/manual/data-modeling/`
2. *A comprehensive guide to data modeling*: `https://www.mongodb.com/resources/basics/databases/data-modeling`
3. *Implementing effective data quality patterns*: `https://www.mongodb.com/developer/products/mongodb/modernizing-rdbms-schemas-mongodb-document/`
4. *MongoDB's advanced security features*: `https://www.mongodb.com/docs/manual/core/queryable-encryption/`

Enjoyed this sample?

Continue reading by purchasing the full book.

https://packt.link/YroGv